

Simulation von Herzkatheterinterventionen bei Kleinkindern

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Dipl.-Inf. Nico Becherer
aus Heidelberg

Mannheim, 2008

Dekan: Professor Dr. Krause, Universität Mannheim
Referent: Professor Dr. Hesser, Universität Mannheim
Korreferent: Professor Dr. Fischer, Universität Mannheim

Tag der mündlichen Prüfung: 25. April 2008

Zusammenfassung

Interventionelle Verfahren stellen seit vielen Jahren ein stetig wachsendes Gebiet in der Therapie angeborener Herzfehler dar. Dennoch existiert bis heute ein Mangel an adäquaten Ausbildungsmethoden um junge Mediziner erfolgreich und risikoarm an die Verwendung dieser Therapien heranzuführen.

Diese Arbeit beschäftigt sich mit der Entwicklung einer Simulationsumgebung zum Training von Herzkatheterinterventionen bei Kleinkindern. Neue Lösungsmöglichkeiten für die physikalische Simulation medizinischer Instrumente (Katheter) im Zusammenspiel mit dem schlagendem Herzen werden vorgestellt.

Der gezeigte Ansatz kombiniert hohe Geschwindigkeit in der Berechnung mit einer realistischen Bewegungssimulation und kann unabhängig von Topologie und Form umliegender Gefäße angewandt werden ohne dabei zugunsten der Performanz auf Freiheitsgrade verzichten zu müssen. Der präsentierte Ansatz basiert auf einem quasi-statischen Modell und einer effizienten Energieminimierung.

Ein Verfahren zur Deformation des schlagenden Herzens mittels einer in Echtzeit berechenbaren pseudoinversen Free Form Deformation Technik wird gezeigt und besprochen.

Weiterhin wird ein schnelles, deskriptives Simulationsverfahren für Kontrastmittelfluss vorgestellt, sowie eine Vielzahl grafischer Verfahren zur Nachbildung realistisch anmutender Bildgebungsverfahren implementiert und ausgewertet.

Es wird im Rahmen dieser Arbeit mittels eines voll funktionsfähigen Prototypen gezeigt, dass mit heutigen Rechnern eine vollständig realistische Simulation von Herzkathetereingriffen bei Kleinkindern möglich ist.

Ehrenwörtliche Erklärung

Ich versichere, dass die vorliegende Arbeit "Simulation von Herzkatheterinterventionen bei Kleinkindern" ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel von mir angefertigt wurde. Die aus den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen wurden als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

(Nico Becherer)

Mannheim, den 1. Februar 2008

Thema:

Simulation von Herzkathetereingriffen bei Kleinkindern

Die Dissertationsarbeit wurde von Nico Becherer ausgeführt am
ICM (Institut für computergestützte Medizin) der Universität Mannheim
unter der Betreuung von

Prof. Dr. J. Hesser¹

Prof. Dr. D. Schranz²

¹Klinik für Strahlentherapie und Radioonkologie am Universitätsklinikum Mannheim, Fakultät für Medizin Mannheim der Universität Heidelberg und Institut für computergestützte Medizin der Universität Heidelberg.

²Hessisches Kinder-Herzzentrum der Justus-Liebig Universität Giessen

Vorwort

”Ein Gelehrter in seinem Laboratorium ist nicht nur ein Techniker; er steht auch vor den Naturgesetzen wie ein Kind vor der Märchenwelt.” Besser als mit dem berühmten Zitat von Marie Curie könnte man diese Arbeit meiner Meinung nach nicht einleiten. Kinder, Naturgesetze, Technik - drei Punkte, die meine Doktorarbeit wohl mehr als alles andere dominiert haben.

In den dreieinhalb Jahren, die ich mit dem Aufbau, der Forschung und der Entwicklung einer Simulationsumgebung für die Kinderkardiologie verbracht habe, gab es immer wieder spannende Entdeckungen zum Staunen, zum Lernen, zum Wundern und zum darüber Nachdenken. Einige meiner Erkenntnisse & Erfahrungen sind in dieser Niederschrift zu finden. Andere werden wahrscheinlich für immer Gedanken bleiben, ohne konkret zwischen den folgenden Seiten aufzutauchen.

Wer mir aber für immer im Gedächtnis und in guter Erinnerung verbleiben wird, das sind die Menschen, die mich während meiner Arbeit unterstützt und mir den Rücken gestärkt haben. Menschen, von denen ich froh bin, einige auch privat als Freunde bezeichnen zu dürfen. Mein besonderer Dank gilt daher:

Prof. Dr. J. Hesser Schon während meiner Diplomarbeit war ich begeistert von seinem Arbeitseifer, seiner Kreativität und seinem Ideenreichtum. Während der Doktorarbeit hat sich das nur noch gesteigert.

Danke für viele spannende Diskussionen, Zeit, Ideen und hilfreiche Literaturvorschläge. Danke für zu jeder Zeit konstruktive Kritik und ehrliche Meinungen. Danke aber vor allem für eine exzellente Betreuung meiner Arbeit und ein offenes Ohr zu jeder Zeit.

Prof. Dr. D. Schranz Ein Mann, der mich mit seiner nicht enden wollenden Energie & seinem unglaublichen Drang, kranken Kindern auf jede denkbare Weise zu helfen immer wieder motiviert hat. Es war mir eine Freude, mit Ihnen zusammenarbeiten zu dürfen, und so danke ich auch dem gesamten Team des Kinderherzzentrums Gießen für ausführliche Erklärungen der medizinischen Wunderwelt, immer ein freundliches Lächeln und viel frischen Kaffee.

Dr. Ulrike Kornmesser & CATHI GmbH Diese Arbeit wäre nie geboren worden ohne die Initiative und die Hilfe von Frau Dr. Kornmesser und der CATHI

GmbH. Menschen die auch während größtdenkbarem Stress und Arbeitsaufkommen immer wieder Zeit für kreative Einfälle, schöne Momente und stundenlange (aber niemals langweilige), angeregte fachliche Diskussionen hatten. It's Rock'n Roll (but I like it).

ICM Mein weiterer Dank gilt den lieben Kollegen vom ICM und damit insbesondere den beiden Sekretärinnen, Andrea Seeger und Christiane Glasbrenner. Auch wenn ich inzwischen den heimischen Hafen verlassen habe und im fernen Norden wohne - in Mannheim & Heidelberg steht ein tolles Team, das sicher noch viele großartige Forschungsergebnisse präsentieren wird.

Danke für Ideenaustausch, für den stärksten Kaffee der Welt und Multikulti in seiner besten Form.

Florian Becker & Ciamak Abkai Ganz besonders möchte ich auch zwei lieben Freunden danken, die nicht nur während der Bürostunden und Mittagspausen immer wieder für schöne, lustige und lehrreiche fachliche wie private Momente gesorgt haben, sondern die auch jetzt, wo diese Arbeit ihrem Ende zugeht, noch unermüdlich nach Fehlern suchen.

Danke für ein paar tolle Jahre, für potenzierte Hilfsbereitschaft, Ehrlichkeit, spannenden Meinungs austausch und großen Spaß. Wir freuen uns auf euren Besuch in Hamburg.

Eveline Becherer Meine Frau, die irgendwie immer noch und jeden Tag an mich glaubt. Die trotz Schwangerschaft (Juna Marie - Wir freuen uns jetzt schon auf Dich...) und Hamburgtrip kurz vor Mitternacht mit Argusaugen diese Arbeit auf Fehler korrigiert und dabei die Genauigkeit jedes Computers um Längen schlägt. Du bist großartig, genau so wie Du bist, und ich freue mich unglaublich auf unsere kleine Hamburger Familie...

Meinen Eltern, Egbert und Regina Becherer. Auch hier kann ich nur sagen, dass ich endlos dankbar bin für alles, was sie in den letzten 29 Jahren für mich getan haben. Und wenn ich mich inzwischen so umsehe, haben sie das gar nicht so verkehrt gemacht.

Danke konkret wieder einmal für eifriges Korrekturlesen und hilfreiche Kritik. Danke für Rückenstärkung und Stressentlastung als es am nötigsten war.

Inhaltsverzeichnis

1	Einleitung	1
2	Ziele und Motivation	3
2.1	Anforderungen an eine Simulation	4
2.2	Herzkathetereingriffe bei Kindern	4
2.2.1	PDA	5
2.2.2	ASD/VSD	6
2.3	Zielsetzungen	7
2.4	Projektspezifikation	7
2.4.1	Im Rahmen dieser Arbeit relevante Spezifikationen	8
2.5	Relevanz	9
3	Grundlagen	11
3.1	Anatomie des Herzens	11
3.1.1	Der Blutkreislauf eines Erwachsenen	11
3.1.2	Der fetale Blutkreislauf	13
3.2	Interventionelle Kardiologie	14
3.2.1	Geschichte der Katheterinterventionen	15
3.2.2	Angeborene Herzfehler	17
3.2.3	PDA	17
3.2.4	ASD	21
3.2.5	VSD	24
3.2.6	Zukünftige Entwicklungen	25
3.3	Bildgebung	25
3.3.1	Röntgen	26
3.3.2	Ultraschall	29
3.3.3	Computertomographie (CT)	31
3.3.4	Kernspintomographie	33
3.4	Mathematische Grundlagen	35
3.4.1	Integrationsverfahren	35
3.4.2	Splines	38
3.5	Physikalische Grundlagen	41
3.5.1	Physik der Instrumente	41
3.5.2	Bewegungsgleichung	42
3.5.3	Lösung der Bewegungsgleichung	43
3.5.4	Fluidsimulation	45

3.6	Computergrafik & Geometrie	48
3.6.1	Quaternionen	48
3.7	Grafische Simulation	49
3.7.1	OpenGL	50
4	Stand der Forschung	52
4.1	Simulationssysteme	52
4.1.1	Nutzen in der Kinderkardiologie	52
4.1.2	Schulungssysteme	53
4.1.3	Mechanische Modelle	53
4.1.4	Tierversuche	55
4.1.5	Ausbildung am Patienten	55
4.1.6	Simulationssysteme	56
4.2	Instrumentensimulation	62
4.2.1	Quasi-Statische Mechanik	63
4.2.2	Direkte, semi-analytische Optimierung	64
4.2.3	Feder Masse Modelle	67
4.2.4	Finite Elemente Methode	69
4.2.5	Andere Modelle	72
4.2.6	Fazit	73
4.3	Kollisionsdetektion	73
4.3.1	Allgemeine Verfahren	73
4.3.2	Verfahren in der kardiologischen Simulation	77
4.4	Distanzberechnungen	77
4.4.1	Distanzfelder	78
4.4.2	Adaptive Distanzfelder	81
4.5	Kontrastmittelsimulation	81
4.5.1	Partikelmodelle	82
4.5.2	Gridbasierte Methoden	83
4.5.3	Textursynthese	85
4.6	Modellanimation	88
4.6.1	Physikalische Deformation	88
4.6.2	Manuelle Deformation	89
4.6.3	Free Form Deformation	89
4.7	Simulations- und Systemarchitekturen	91
4.7.1	Spring	92
4.7.2	GiPSi	92
4.7.3	Sofa	93
4.8	Beschreibungssprachen	95
4.8.1	AnatML	95
4.8.2	SSTML	95
5	Physikalische Realisierung	97
5.1	Instrumentensimulation	97
5.1.1	Einführung	97
5.1.2	Geometrische Repräsentation	98
5.1.3	Interne Energie	100

5.1.4	Externe Energie	101
5.1.5	Numerische Approximation des Gradienten	102
5.1.6	Verschachtelte Instrumente	103
5.1.7	Erste Anpassung des Funktionsraums	105
5.1.8	Lösung des Optimierungsproblems	107
5.1.9	Geschwindigkeitsoptimierung	117
5.1.10	Algorithmus	119
5.2	Resultate der Instrumentensimulation	120
5.2.1	Realitätsnähe	120
5.2.2	Performanz	122
5.3	Datengewinnung	124
5.3.1	Rekonstruktion	124
5.3.2	Baumstrukturen	125
5.3.3	Wandlung in Distanzfelder	126
5.4	Animation	127
5.5	Kontrastmittel	131
5.5.1	Zelluläre Baummethode	133
5.6	Resultate der Kontrastmittelsimulation	134
6	Graphische Realisierung	137
6.1	Kerammatrizen	138
6.1.1	Modelview Matrix	138
6.1.2	Projektionsmatrix	139
6.2	Volumenrendering	140
6.3	Herzschatten	143
6.4	Kontrastmittel	144
6.5	Instrumente	145
6.6	Postprocessing	147
6.6.1	Rauschen	147
7	Technische Umsetzung	149
7.1	Überblick	149
7.2	Klassenstruktur	150
7.2.1	System & GUI Libraries	151
7.2.2	Physik & Simulation	153
7.2.3	Grafik	156
7.2.4	Werkzeuge & Generische Klassen	157
7.3	Performanceoptimierung	158
7.3.1	Threadoptimierung	158
7.3.2	Funktionsoptimierung	160
7.4	Werkzeuge	161
7.4.1	Cathi ^{Kid} Designer	161
7.4.2	dtr2raw, stl2dtr, cathi2dtr	161
7.4.3	Cinema 4D Skripte	162

8 Zusammenfassung und Ausblick	163
8.1 Beispieleingriff	164
8.1.1 Evaluation	165
8.1.2 Ausblick	165

Abbildungsverzeichnis

2.1	Persistierender Ductus Arteriosus	5
2.2	Atrium Septum Defekt	6
2.3	Ventrikel Septum Defekt	7
3.1	Anatomische Darstellung des Herzens	12
3.2	Überblick über den fetalen Kreislauf	13
3.3	Klinische Relevanz angeborener Herzfehler	14
3.4	Schweregrad angeborener Herzfehler	14
3.5	Häufigkeit spezifischer angeborener Herzfehler	15
3.6	Dr. Werner Forßmann und sein berühmtes Röntgenbild	16
3.7	Ductus Arteriosus in Röntgensicht	18
3.8	Gekrümmt geformte Spitze eines Cobra Katheters	19
3.9	Platzierung eines Ductus Occluders	21
3.10	Darstellung eines ASD im Ultraschall	22
3.11	Platzierung eines ASD Occluders	23
3.12	VSD im Farbdopplerultraschall	24
3.13	Schematische Darstellung der Brennflecken	26
3.14	Auf einen C-Arm montierte Röntgenanlage	28
3.15	Kinderkardiologische Aufnahme im Farbdopplerverfahren	30
3.16	Schichtbild aus einer CT Untersuchung (hier VSD).	31
3.17	Simulierte Röntgenaufnahme aus CT Volumendaten.	33
3.18	Schichtbild aus einer Kernspinuntersuchung (hier VSD).	34
3.19	Darstellung des expliziten Euler Verfahrens	36
3.20	Die 5 Bernstein Basispolynome vom Grad 4	40
3.21	Vergleich verschiedener Strömungsbilder	45
4.1	Pulsatiles Corosim Modell der Firma Mecora	54
4.2	Pulsatiles Modell der Firma Sawbones	55
4.3	Cathi Simulator der Firma CATHI GmbH	57
4.4	CathLabVR Simulator der Firma Immersion	57
4.5	VIST Simulator der Firma Mentice	59
4.6	ANGIO Mentor Simulator der Firma Symbionix	59
4.7	SimSuite Simulator der Firma Medical Simulation Corporation	60
4.8	Screenshot aus dem EVE System der Fa. The Simulation Group	61
4.9	Führungsdrahtsimulation von Alderliesten et al. [30].	66
4.10	Führungsdrahtsimulation von Wang et al. [204].	68
4.11	Kathetersimulation von Lawton et al. [128].	70

4.12	Kathetersimulation von Duriez et al. [68].	72
4.13	AABBs im exemplarischen Beispiel.	74
4.14	Spheretrees im exemplarischen Beispiel (Bild aus [155])	75
4.15	Swept Sphere Volumes aus Punkt, Linie und Rechteck im Vergleich	76
4.16	Beispielhafte Darstellung eines Distanzfelds	80
4.17	Fluidsimulation mit SPH von Müller et al.	83
4.18	Fluidsimulation mit SPH auf GPU von Zhang et al.	83
4.19	Gridbasierte Fluidsimulation von Jos Stam	84
4.20	Gridbasierte Fluidsimulation von Wei	85
4.21	Mittels Perlin Rauschen simulierter Rauch	86
4.22	Darstellung einer Flow Noise Textur	87
4.23	Grobe Darstellung des Spring Frameworks	92
4.24	Grobe Darstellung des GiPSi Frameworks	93
4.25	Multi Model Repräsentierung im SOFA Framework	94
4.26	Szenengraph aus dem SOFA Framework	94
4.27	Beispielhafte SSTML Daten	96
5.1	Katheterform mit Gelenken	99
5.2	Transformation des Koordinatensystems am Kathetergelenk	100
5.3	Darstellung der Deformation	101
5.4	Darstellung verschachtelter Instrumente	104
5.5	Einteilung eines Instruments in Formpakete	104
5.6	Aufspaltung der Instrumentenpakete	105
5.7	Energieplot der Instrumentenfunktion.	106
5.8	Graph aus Gefäßstrukturen	107
5.9	Energieplot der optimierten Instrumentenfunktion.	108
5.10	Repräsentative Darstellung der Multimodalität der Zielfunktion	110
5.11	Redundanter Einfluss aufeinanderfolgender Gelenke	110
5.12	Vergleich der Lösungsverfahren	112
5.13	Vergleich der Lösungsverfahren 2	113
5.14	Optimierungsverlauf nahe des Optimums	115
5.15	Neustart des KG Verfahrens	115
5.16	Optimierung mit gleitender Gradientenberechnung	116
5.17	Optimierung mit gleitender Gradientenberechnung	117
5.18	Großes Distanzfeld der Beinarterien	118
5.19	Darstellung der Instrumentenberechnung	120
5.20	Veränderung der Steifigkeit	121
5.21	Aufbau des Phantoms	121
5.22	Vergleich realer Katheter (blau) und virtuelles Instrument (rot).	122
5.23	Rechenzeit ohne Linearisierung	123
5.24	Rechenzeit mit Linearisierung	124
5.25	CT Daten als Grundlage der Herzrekonstruktion	125
5.26	Rekonstruierte Polygonale Daten	125
5.27	Geglättete Polygonale Daten	126
5.28	Komplett bearbeiteter und unterteilter Datensatz	127
5.29	Auswirkung einer FFD auf das Herzmodell	128
5.30	Durch FFD Volumen direkt deformierter Katheter	129

5.31	Sich selbst schneidendes FFD Volumen	130
5.32	Zerlegung eines Würfels in Tetraeder	130
5.33	Deformation und inverse Deformation	131
5.34	Verlauf des applizierten Kontrastmittels	134
5.35	Schematische Darstellung der Partikelbewegung	135
5.36	Rechenzeit im Zusammenhang mit Anzahl der Partikel	135
6.1	Darstellung der graphischen Elemente	137
6.2	Auf C-Arm montierte Röntgenanlage	138
6.3	Röntgenhintergrund als Volumengrafik	140
6.4	Schematische Darstellung eines 3D Texture Mappings	141
6.5	Auswirkungen der Kantendetektion auf das Hintergrundbild	142
6.6	Generierung des Herzschattens	143
6.7	Mögliche Texturen für das Partikelsystem	144
6.8	Auswirkung des Stencil Buffers auf die Partikelgrafiken	145
6.9	Darstellung der Gitterstruktur bei einem Occluder	145
6.10	Skizzierung zur Parallel Transport Frame Technik	146
6.11	Berechnung einer Röhre mit Fixed Up- (links) und PTF Methode	147
6.12	Darstellung der Diskontinuitäten im Frenet Verfahren	147
7.1	Übersicht der entwickelten Bibliotheken	150
7.2	Klassendiagramm (vereinfacht) der Kernel Library	152
7.3	Verschiedener Bedienoberflächen für Cathi ^{Kid}	153
7.4	Klassendiagramm der Instrumenten Library	154
7.5	Klassendiagramm (vereinfacht) der 3D Modell Library	156
7.6	Systemtrace der Simulationsanwendung	159
7.7	Screenshot von Cathi ^{Kid} Designer	161
8.1	Die vollständige Simulationsumgebung in Aktion	163
8.2	Exemplarisch simulierter Ductuseingriff	164

Kapitel 1

Einleitung

Als Dr. Werner Forßmann¹ 1929 im Alter von 25 Jahren im Selbstversuch seine Idee der Herzkatheterisierung durch eine Armvene testete, ahnte wohl noch niemand, welche Auswirkungen auf die Medizin, insbesondere die Kardiologie, dieser Versuch haben würde.

Heute, knapp 78 Jahre später, sind katheterbasierte Verfahren zu einem nicht mehr wegzudenkenden Eckpfeiler der Medizin geworden. Das Spektrum reicht von mehr oder weniger einfachen Angiographien², in denen Katheter zur Applikation des Kontrastmittels genutzt wird, bis hin zu hochkomplizierten Eingriffen bei denen Herzklappen ersetzt oder Defekte in der Herzstruktur durch das Einsetzen spezieller Plastiken repariert werden können.

Die Auswirkungen, die die interventionelle Kardiologie auf die Behandlung kongenitaler³ Herzfehler hat, ist beachtlich. Hatte man zu Beginn der 70er Jahre noch mit einer Sterberate von knapp unter 40% [177] im ersten Lebensjahr zu kämpfen, ist der Überlebenssatz heute auf 80-90% [40] angestiegen. Zu verdanken ist diese Verbesserung zu großen Teilen der Tatsache, dass viele der angeborenen Fehlbildungen heute bereits im ersten Jahr nach der Geburt weitestgehend sicher behandelt werden können [17].

Die stetige Verbesserung der Methoden ist allerdings ein langsamer Vorgang. Weltweit gibt es nur wenige Ärzte, die ausreichend in der interventionellen Kinderkardiologie geschult sind, um die vielversprechenden aber komplizierten Methoden mit einem minimalen Risiko anwenden zu können.

Während in der konventionellen (Erwachsenen) Kardiologie immer mehr VR-Trainingssimulationen ihren Weg in die ärztliche Ausbildung finden, sind derartige Geräte in der Kinderkardiologie noch nicht vorhanden.

Mag es auf den ersten Blick erstaunlich klingen, dass der Markt der Simulationssysteme so stark in die Richtung der herkömmlichen Kardiologie verschoben ist, so fallen bei näherer Betrachtung schnell einige offensichtliche Gründe ins Auge.

An vorderster Front stehen die technischen Probleme. Während der Großteil der Eingriffe im Bereich der Erwachsenenherzkardiologie in den Koronarien - räumlich stark begrenzten Gefäßen - oder im Bereich der Aorta mit wenig Variabilität stattfindet,

¹Deutscher Mediziner 1904-1979

²Gefäßuntersuchungen

³angeborener

agieren Kinderkardiologen im Bereich des gesamten Herzens - vornehmlich in den Kavernen⁴.

Viele vereinfachende Annahmen und Restriktionen, die bei der Simulation von Eingriffen in den Koronarien getroffen werden können, verlieren im Bereich der Kinderkardiologie ihre Gültigkeit. Bisherige Entwicklungen sind gar nicht oder nur mit sehr hohem Aufwand zwischen den Anwendungsgebieten übertragbar.

Hinzu kommen die rein wirtschaftlichen Ursachen. Die Entwicklung eines möglichst realistischen Simulationssystems ist ein kostspieliges Unterfangen. Forschung, Simulationsdesign, Datenerhebung - all das verursacht Kosten, die durch die zu erwartenden Einnahmen (der Anteil an Kinderkardiologen mit interventioneller Qualifikation ist im Vergleich zum Anteil an konventionellen Kardiologen verschwindend gering) nicht sicher gedeckt werden können. Viele kommerzielle Unternehmen scheuen dieses Risiko. Ein Mangel, der mit den Möglichkeiten der universitären Forschung zumindest teilweise ausgeglichen werden sollte.

Diese Arbeit beschäftigt sich daher mit der Entwicklung eines Simulationssystems zum Training von typischen Eingriffen in der Kinderkardiologie. Technische Aufgabenstellungen, die sich beim Entwurf einer solchen Simulationsumgebung stellen - sowohl im Bereich der physikalischen als auch der grafischen Simulation - werden diskutiert und, wo möglich, Lösungen präsentiert.

Ziel dieser Dissertation sollte es von Beginn an sein, ein Trainingssystem zur Ausbildung von Kinderkardiologen zu erschaffen.

Mit Hilfe der CATHI GmbH (insbesondere Frau Dr. Kornmesser), der großartigen Unterstützung durch meinen Doktorvater (Prof. Dr. Jürgen Hesser) und Herrn Prof. Dr. Schranz - Kinderkardiologe mit Schwerpunkt Herzkatheter und Intensivmedizin sowie Leiter des Kinderherzzentrums Gießen - ist dies auch gelungen. Das System Cathi^{Kid} kann heute bereits in experimentellen Umgebungen eingesetzt werden und wird voraussichtlich plangemäß zu Beginn des zweiten Quartals 2008 erstmals im Rahmen einer Schulung von Siemens Medical öffentlich präsentiert werden.

Zukünftig ist zu hoffen, dass die Simulation ihren Platz als wertvolles Instrument in der praktischen Ausbildung von (Kinder)Kardiologen einnehmen wird.

Die weiteren Kapitel gliedern sich wie folgt: in Kapitel 2 werden Motivation und Hintergründe zu dieser Arbeit erläutert. Kapitel 3 schließt mit einer Vorstellung der thematisch relevanten Grundlagen an. In Kapitel 4 wird auf den aktuellen Stand der Technik in den verschiedenen Bereichen eingegangen. Kapitel 5 und 6 beschäftigen sich mit der eigenen Arbeit im Bereich der physikalischen und graphischen Simulation. Von der technischen Umsetzung berichtet Kapitel 7. In Kapitel 8 werden zusammenfassend Resultate vorgestellt und abschließend ein Ausblick auf mögliche zukünftige Entwicklungen geworfen.

⁴Hohlräumen

Kapitel 2

Ziele und Motivation

Die Medizin ist eine, im Gegensatz zu den Naturwissenschaften, stark von Erfahrungswerten geprägte Wissenschaft. Techniken und Behandlungsmethoden werden zwar genauestens evaluiert, sind aber häufig - insbesondere im interventionellen Umfeld - nur durch stetiges Training und jahrelange Erfahrung sicher zu erlernen. Besonders deutlich zeigt sich das in der Ausbildung.

Situationen, Menschen und Patienten lassen sich oft nur schwer in objektiv zu erfassenden Zahlen und Werten ausdrücken. Aus diesem Grund folgt auf die theoretische Ausbildung eines Arztes, bei der allgemein gültige Fakten und Grundlagen vermittelt werden, eine lange praktische Phase am Patienten, in der der Mediziner von den Kenntnissen und dem Wissen seiner älteren und erfahreneren Kollegen profitieren kann.

Auch in der Kardiologie stellen diese Erfahrungen einen unschätzbaren Wert dar. Situationen müssen oft blitzschnell beurteilt, Interventionen routiniert und mit Sachverstand durchgeführt werden. Bei all dem handelt es sich um Fähigkeiten, die nicht allein aus einer theoretischen Beschäftigung heraus gewonnen werden können.

Bis vor wenigen Jahren gab es nur wenige Möglichkeiten, effektiv Eingriffe zu trainieren. Neben dem Training an Phantommodellen stellte der Tierversuch oft die einzige Möglichkeit zur Ausbildung der jungen Mediziner dar. Wo Flugsimulationen schon seit langer Zeit zum Standard in der Pilotenausbildung gehören, scheiterte es in der Medizin noch an der technischen Leistungsfähigkeit und am Vertrauen in die unbekannte Technologie der computergestützten medizinischen Simulation.

Das hat sich inzwischen geändert. Seit den 90er Jahren halten immer mehr Simulatoren Einzug in die medizinische Ausbildung. Das Spektrum reicht dabei von einfachen EKG Simulatoren, die spezifische Störungsbilder aufgrund vorher aufgenommener Kurven reproduzieren, bis hin zu hoch komplexen Full Scale Simulationen, die es sich zur Aufgabe gemacht haben, den Patienten in seiner ganzen physiologischen und anatomischen Komplexität nachzubilden. Andere Systeme konzentrieren sich auf die möglichst exakte Simulation konkreter Eingriffe. Augenoperationen, endoskopische Anwendungen und - nicht zuletzt - natürlich Herzkatheterinterventionen sind inzwischen in verschiedenen Realitätsgraden simulierbar.

Die Vorteile liegen auf der Hand. Situationen können beliebig oft und ohne Gefahr für den Patienten simuliert werden. Techniken können bis hin zur Perfektion trainiert, selten auftretende Komplikationen reproduziert werden. Mediziner sind dadurch in der Lage bereits Routine in spezifischen Eingriffen zu erlangen, bevor sie mit ihrem

ersten Patienten in Kontakt kommen. Als Resultat kann die steile Lernkurve in der praktischen Medizin deutlich abgeflacht werden [111].

Ziel dieser Arbeit ist es, ein Simulationssystem für die Ausbildung von Kinderkardiologen zu schaffen. Konkret soll ein Framework für die Simulation verschiedener interventionelle Eingriffe geschaffen werden. Das resultierende System - Cathi^{Kid} - baut hardwaretechnisch auf das existierende Cathi System der Firma CATHI GmbH auf, verwendet aber aufgrund der durch die Kinderkardiologie gegebenen anderen Ansprüche eine von Grund auf neu geschriebene Softwarebasis mit neuen Denk- und Entwicklungsansätzen.

2.1 Anforderungen an eine Simulation

Während der Durchführung verschiedener Eingriffe verwenden Kardiologen lange, drahtähnliche Instrumente (Katheter, Führungsdrähte...). Diese werden über große Körpervenen oder -arterien in den Patienten eingeführt und, mittels manueller Translation und Rotation, unter Röntgensicht durch das Gefäßsystem navigiert. Zur optischen Orientierung wird Kontrastmittel über die Katheter appliziert, das sich dann mit dem Blutfluss im Körper verteilt.

Für ein computergestütztes Training primär relevant ist die Handhabung der Instrumente während der Navigation. Um so realer das Verhalten der virtuellen Instrumente in der Simulation abgebildet wird, um so eher ist der Kardiologe später in der Lage, die Erfahrungen aus der Simulation in den medizinischen Alltag in der Klinik zu übertragen.

Instrumente müssen physikalisch korrekt und in sehr kurzen Zeitabschnitten (typischerweise bis zu 50 Bilder pro Sekunde) berechnet werden, um auch auf kleinste Veränderungen noch adäquat zu reagieren. Erschwerend kommt hinzu, dass Instrumente miteinander (bsp. durch Katheter hindurch geführte Drähte) interagieren, und durch das schlagende Herz beeinflusst werden.

Daneben ist auch eine gute graphische Umsetzung für den Trainingseffekt von großer Bedeutung. Diagnostische Methoden sollen erlernbar werden und das Auge des Kardiologen für bestimmte Details geschult werden. Möglichst realistische Approximationen von Röntgenbildern müssen berechnet werden, der Fluss des Kontrastmittels simuliert werden.

All diese Aufgaben müssen, gemeinsam mit der Verwaltung der Simulationsumgebung und der Kommunikation mit der Hardware parallel in Echtzeit gerechnet werden, um dem Nutzer den Eindruck eines realen Eingriffs verschaffen zu können.

2.2 Herzkathetereingriffe bei Kindern

Allein in Deutschland leben ca. 200.000-300.000 Menschen mit einem angeborenen Herzfehler. Jahr für Jahr kommen etwa 8.000 Neugeborene mit einem kongenitalen Herzfehler zur Welt [18]. Statistisch gesehen wird also etwa jedes 100. Kind mit einem Herzfehler geboren. Ein Großteil dieser Defekte ist zum Glück klinisch ohne große Bedeutung oder mit relativ einfachen Mitteln zu beheben [23]. Ein signifikanter Teil bedarf aber auch einer interventionellen Behandlung. Dabei handelt es sich

hauptsächlich um Interventionen zum Verschluss eines sogenannten PDA¹ oder eines Loches in der das linke und das rechte Ventrikel² oder Atrium³ voneinander trennenden Herzscheidewand.

2.2.1 PDA

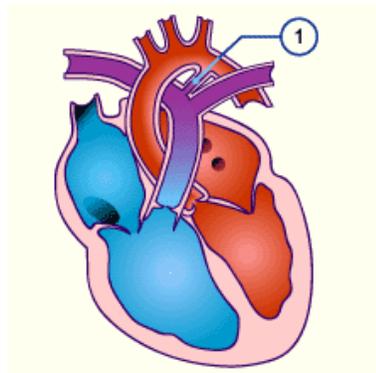


Abbildung 2.1: Schematische Darstellung eines persistierenden Ductus Arteriosus (1). Blut kann über den offenen Shunt von der aortalen auf die pulmonale Seite fließen.

Der Ductus arteriosus hält im fetalen⁴ Blutkreislauf eine Verbindung zwischen Aorta und Lungenarterie aufrecht. Das ist nötig, da die Lunge vor der Geburt noch nicht belüftet und voll entfaltet ist. Da sie somit auch nicht relevant durchblutet ist, fließt das Blut über den Ductus arteriosus⁵ direkt in die Aorta (siehe Abbildung 2.1).

In den ersten Lebensstunden bis -tagen [167] verschließt sich üblicherweise der Ductus und wird zum Ligamentum arteriosum - einem Bindegewebsstrang, der den Truncus pulmonalis mit dem Aortenbogen verbindet.

Bleibt dieser selbstständige Verschluss aus, spricht man vom PDA - einem persistierenden Ductus arteriosus. Ein größerer PDA bedeutet einen Links-Rechts-Shunt, bei dem Blut aus dem arteriellen Kreislauf (Körperkreislauf) direkt wieder in den Lungenkreislauf gelangt. Dadurch kann es zu Herzinsuffizienzzeichen und einer vermehrten Infektanfälligkeit der Kinder kommen. Ein größerer PDA bedeutet auch eine Endokarditisgefahr. Deshalb wird ein größerer PDA beinahe immer verschlossen (Ausnahmen sind PDAs, die in Zusammenhang mit einigen komplexen Herzfehlern auftreten - hier kann ggf. sogar ein interventionelles Offenhalten des Ductus via Stentimplantation nötig werden).

Der Verschluss des PDA geschieht üblicherweise als Herzkathetereingriff. Während der Intervention (siehe Kapitel 3.2.3), stellt der behandelnde Arzt den Ductus in Röntgendurchleuchtung dar, während über einen in der Aorta platzierten Katheter

¹Persistierender Ductus Arteriosus

²Herzkammer

³Vorhof

⁴vor der Geburt

⁵auch Ductus Botalli genannt, nach Leonard Botallo

Kontrastmittel in Ductusnähe injiziert wird. Zum Verschluss wird über mehrere Zwischenschritte ein sogenannter PDA Occluder⁶ in den Ductus navigiert und mechanisch aufgespannt.

Im Gegensatz zum chirurgischen Eingriff ist der Kathetereingriff komplikationsärmer. Die Überlebensrate steigt deutlich, Krankenhausaufenthalte können verkürzt werden und die Heilungsdauer ist, aufgrund des Ausbleibens einer größeren offenen Wunde, verkürzt.

2.2.2 ASD/VSD

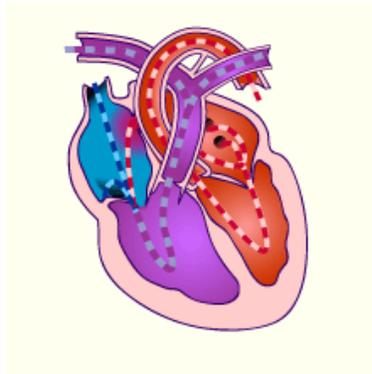


Abbildung 2.2: Schematische Darstellung eines Atrium Septum Defekts. Sauerstoffreiches Blut kann vom linken in den rechten Vorhof überfließen (rot markiert)

Beim ASD⁷ handelt es sich um ein Loch in der Herzscheidewand, die den linken und den rechten Vorhof voneinander trennt (siehe Abbildung 2.2). Er ist mit ca. 7,5% aller angeborenen Herzfehler die dritthäufigste angeborene Herzfehlbildung [23]. Der VSD⁸ stellt ebenfalls ein Loch in der Herzscheidewand dar - allerdings zwischen den Herzkammern (siehe Abbildung 2.3).

Durch das Loch, das meist in der Mitte der Scheidewand liegt, kann je nach Durchmesser aufgrund des Druckunterschiedes in den Kavernen eine mehr oder weniger große Blutmenge von links nach rechts, also wieder in den Lungenkreislauf rückfließen.

Sofern die Voraussetzungen dafür gegeben sind (glatte Ränder, mittige Platzierung und ausreichend kleiner Durchmesser), geschieht ein Verschluss des ASD heute in der Regel über Kathetertechniken (alternativ - chirurgischer Verschluss). Ein Doppelschirmchen wird durch einen Katheter in das Loch eingeführt und dort entfaltet (siehe auch Kapitel 3.2.4 & 3.2.5).

Auch hier ist die Katheterintervention der chirurgischen Behandlung deutlich überlegen. Die großen Vorteile sind identisch zu den Vorteilen während der Ductusintervention.

⁶Konisch geformter, elastischer Verschlussstopfen aus Draht

⁷Atrium Septum Defekt

⁸Ventrikel Septum Defekt

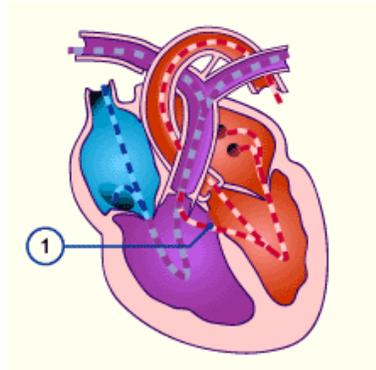


Abbildung 2.3: Schematische Darstellung eines Ventrikel Septum Defekts (1). Sauerstoffreiches Blut kann von der linken in die rechte Kammer überfließen.

2.3 Zielsetzungen

Der Schwerpunkt der vorliegenden Arbeit liegt in der technischen Umsetzung und Aufarbeitung der Simulationsumgebung. Um im kardiologischen Umfeld eingesetzt werden zu können, ist daher vor allem eine echtzeitfähige und physikalisch weitgehend korrekte Simulation der Instrumente in einer deformierbaren und sich selbsttätig bewegenden Umgebung gefordert.

Die Simulation der Instrumente stellt aber immer auch eine Wechselwirkung mit dem umliegenden Gewebe dar. Daher ist es für eine entsprechend genaue Untersuchung erforderlich, Datensätze zu erstellen - einen künstlichen Patienten zu erschaffen. Die Verfahren, die hierfür zum Einsatz kamen werden im im Kapitel über physikalische Simulation näher vorgestellt und erläutert.

Ein System zur Simulation des Kontrastmittelflusses mit der Anforderung, realistische Ergebnisse zu liefern ohne einen großen Einfluss auf die Rechenzeit auszuüben soll entwickelt werden. Herkömmliche Verfahren sind in der Regel zu rechenaufwendig oder zu stark reduziert, als dass sie direkt in der Simulation hätten verwendet werden können. Verfahren aus verschiedenen Bereichen der Partikelsimulation und der zellulären Automaten wurden daher kombiniert um ein schnelles und optisch ansprechendes Verfahren zu kreieren.

Abgesehen von der physikalischen Simulation wurde eine Vielzahl an Techniken eingesetzt um die graphische Darstellung möglichst nahe an die medizinische Realität anzunähern, ohne dabei die wichtigen Echtzeitfähigkeiten einbüßen zu müssen. Die Umsetzung dieser Effekte wird im zweiten großen Teil der Arbeit beschrieben.

2.4 Projektspezifikation

Zu Beginn der Arbeit (Januar 2005) wurde das Projekt in Zusammenarbeit mit dem Kinderherzzentrum des Universitätsklinikums Gießen folgendermaßen spezifiziert und die forschungsrelevanten Pakete zusammengefasst.

2.4.1 Im Rahmen dieser Arbeit relevante Spezifikationen

Herzmodell

Ziel ist es, ein exaktes dreidimensionales Modell des schlagenden Herzens aus MRT, CT- oder Angiographiedaten zu rekonstruieren. Hierfür ist es nötig, zuerst die MRT Daten mittels bekannter Methoden zu segmentieren um anschließend eine Umwandlung der Herzvolumendaten in Polygondaten vorzunehmen.

Die Rekonstruktion kann, falls nötig, durch die Verknüpfung mit Daten aus Angiographien weiter unterstützt und verbessert werden, um im weiteren Verlauf des Projektes ein möglichst realistisches Herzmodell verwenden zu können. Weiterhin ist die Simulation der Deformation des schlagenden Herzens in Echtzeit geplant.

Draht/Kathetermodell

Um Eingriffe realisieren zu können, ist es nötig, die physikalischen Eigenschaften von Kathetern, Führungsdrähten und anderen, spezialisierten Instrumenten in Echtzeit zu simulieren. Das betrifft sowohl die Interaktion zwischen Instrumenten und Gewebe als auch die Beeinflussung von Instrumenten untereinander. Da das bestehende Cathi System nur ein stark vereinfachtes Drahtmodell verwendet, ist eine Erweiterung auf physikalisch korrekte Berechnungen mittels näher zu spezifizierender Methoden zwingend notwendig.

Instrumente

Die Hardwaresimulation der verwendeten Instrumente (Katheter, Draht, Ballon/Schirmchen) ist bereits zu großen Teilen in Cathi implementiert und kann für das geplante Projekt übernommen werden. Eine Erweiterung in Bezug auf die Anzahl der zeitgleich nutzbaren Instrumente ist allerdings nötig, da Cathi auf die Verwendung einer einzelnen Katheter/Draht/Ballon Kombination beschränkt ist.

Flussmodell

Ein neues Flussmodell zur Simulation von Strömungen sowie der Verteilung von Kontrastmittel in großen Hohlräumen muss entwickelt werden. Es ist geplant, soweit das rechentechnisch umsetzbar ist, eine simplifizierte Form des bekannten Navier/Stokes Modells zu verwenden, um ein möglichst realistisches Fluidverhalten zu simulieren.

Grafik

Die in Cathi bestehende Grafikumgebung kann nicht für die Darstellung von Hohlräumen verwendet werden. Daher wird es nötig sein, eine völlig neue Grafikumgebung zu entwickeln. Geplant ist der Einsatz von OpenGL zur Berechnung von Translationen und Rotationen der C-Arme, sowie zur Darstellung der anschließenden Projektion. Hintergründe werden aus Volumendaten unter Verwendung von 3D-Texture Mapping berechnet. Für die Visualisierung von Kontrastmittel ist der Einsatz eines Partikelsystems geplant.

Nicht im Rahmen dieser Arbeit relevante Spezifikationen

Force Feedback

Ein Force Feedback System ist derzeit für Cathi in Entwicklung. Eine Verwendung der Resultate in Cathi^{Kid} ist geplant⁹.

Komplikationen

Die nichtdeterministische Simulation einer möglichst großen Auswahl von Komplikationen soll ermöglicht werden. Ein auf System zur schnellen Auswertung von Komplikationsrisiken, basierend auf Bayesschen Netzen wurde bereits für Cathi entwickelt und kann zu großen Teilen übernommen werden¹⁰.

Ultraschallmodell

Die Entwicklung eines simulierten, den Eingriff unterstützenden Ultraschalls ist zu einem späteren Zeitpunkt im Projekt vorgesehen. Eine Simulation von (mehrgradigen) Refraktionen ist denkbar, die Nachbildung von Dopplertechniken geplant¹¹.

EKG-Modell

Die Simulation von EKG-Kurven ist derzeit für Cathi in Entwicklung. Eine Verwendung der Resultate in Cathi^{Kid} ist geplant¹².

E-Learning

Die Entwicklung eines Systems zur optimalen, rechnergestützten Lernbetreuung ist derzeit für Cathi in Entwicklung. Die Anpassung und der Einsatz des Systems für Cathi^{Kid} sind geplant¹³.

2.5 Relevanz

Bisher existiert, nach Wissen des Autors und Prüfung der bekannten Veröffentlichungen (Stand Januar 2008) kein Simulationssystem für die kinder-kardiologische Ausbildung.

Die Firma SimSuite bietet zwar eine Simulation zum ASD Verschluss an [12] - allerdings wird dabei viel mit aufgenommenen Filmen und vorberechneten Pfaden gearbeitet, was das System in Bezug auf Simulation und Nutzbarkeit sehr unflexibel macht

⁹Zum Zeitpunkt des Abschlusses dieser Arbeit existiert ein Force Feedback System bei der Firma CATHI GmbH, dessen Verwendung für Cathi^{Kid} geplant ist.

¹⁰Die Berechnung der Komplikationen ist nicht Teil dieser Arbeit, sondern wurde vom Autor bereits in einer vorhergehenden Arbeit eingehend untersucht und behandelt [42].

¹¹Eine Echtzeit Ultraschallsimulation wurde am ICM entwickelt, allerdings nur sehr am Rande unter der Mitarbeit des Autors. Daher liegt sie nicht im Fokus dieser Arbeit. Quellen finden sich unter [27].

¹²Zum Zeitpunkt des Abschlusses dieser Arbeit existiert ein auf Single Moving Dipole Simulation [94] basierendes EKG System bei der Firma CATHI GmbH, dessen Verwendung für Cathi^{Kid} geplant ist.

¹³Zum Zeitpunkt des Abschlusses dieser Arbeit ist ein verbessertes E-Learning System bei der CATHI GmbH in Entwicklung

und eher die Einordnung in die Kategorie der Lernsystem nahelegt. Zudem stellt die Simulation zum gegenwärtigen Zeitpunkt nur eine Plattform für eine Intervention im Erwachsenenalter, die sich stellenweise in Prozedere und Diagnostik von dem pädiatrischen Fällen unterscheidet.

Cathi^{Kid} - das Projekt dieser Dissertation - stellt somit ein Novum im Bereich der medizinischen Simulation dar. Erstmals ist es gelungen einen Simulator zu entwickeln, der das Verhalten von Instrumenten auch in Hohlräumen realistisch abbildet und exemplarisch eine vollständige Simulation des PDA Eingriffes ermöglicht. Instrumente werden in Echtzeit schnell und mit hoher Genauigkeit berechnet. Andere physikalische und graphische Aspekte können realistisch und mit einem Minimum an Rechenzeitverbrauch simuliert werden.

Dank größtmöglicher Flexibilität ist die Simulation weiterer kardiologischer (und ggf. auch kardiologieferner) Eingriffe mit den geschaffenen Libraries und Frameworks nur eine Frage von Datensätzen und kleinerer Detailanpassungen.

Kapitel 3

Grundlagen

Im folgenden Kapitel wird zunächst, soweit das für das Verständnis dieser Arbeit relevant ist, ein Überblick über die menschliche Anatomie gegeben. Der grundlegende Aufbau des Herzens wird gezeigt und der Blutkreislauf erläutert. Zudem werden Unterschiede zwischen dem fetalen und dem erwachsenen Blutkreislauf aufgezeigt.

Es folgt eine Einführung in die interventionelle (Kinder)Kardiologie. Einige angeborene Herzfehler werden ausführlicher vorgestellt und Möglichkeiten der Therapie erläutert. Ein besonderer Augenmerk wird dabei auf die in dieser Arbeit simulierte PDA Intervention gelegt. Kurz werden auch derzeitige Entwicklungen präsentiert, die möglicherweise einen maßgeblichen Einfluss auf zukünftige Interventionen haben werden.

Der darauf folgende technische Teil beginnt mit einem Abschnitt über verschiedene Bildgebungsverfahren, die direkt oder indirekt im Rahmen dieser Arbeit verwendet werden.

Ein Abschnitt über mathematische Grundlagen leitet dann über zu den Kernabschnitten über physikalische und grafische Simulation. Abschließend werden hier Mittel und Methoden vorgestellt, die die Basis für die Simulation von Instrumenten, Kontrastmittelfluss und Röntgendarstellung bilden.

3.1 Anatomie des Herzens

Das Herz (lateinisch-anatomisch Cor) ist ein ca. faustgroßes muskuläres Hohlorgan, das den Körper durch rhythmische Kontraktionen mit Blut versorgt und dadurch die lebenswichtige Durchblutung aller Organe sichert. Aufgebaut ist es vergleichbar mit einer Pumpe, die Flüssigkeit ventilgesteuert (Klappen) aus dem venösen Teil des Gefäßsystems ansaugt und in den arteriellen Teil ausstößt.

3.1.1 Der Blutkreislauf eines Erwachsenen

Die grundlegende Funktionsweise ist in Bild 3.1 dargestellt. Über die Vena Cava Inferior/Superior¹ gelangt während der Diastole² verbrauchtes, sauerstoffarmes (im Bild

¹Untere/Obere Hohlvene

²Füllphase

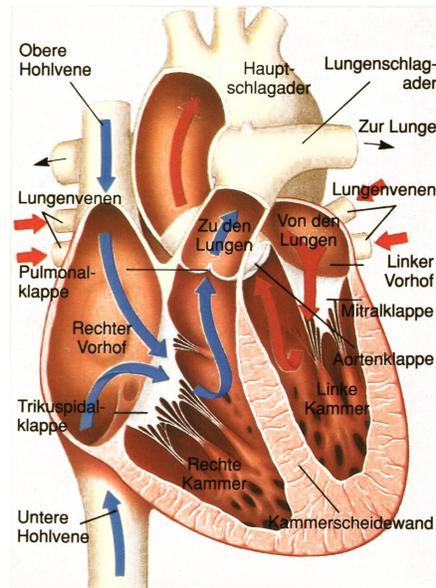


Abbildung 3.1: Anatomische Darstellung des Herzens. Blaue Pfeile symbolisieren den Fluss des sauerstoffarmen, rote die des sauerstoffreichen Blutes.

durch blaue Pfeile verdeutlicht) Blut in den rechten Vorhof. Das Blut wird durch rhythmische Kontraktionen des Vorhofs in die rechte Kammer gepumpt. Dabei passiert es die sich während der Kontraktion der Vorhöfe öffnende Trikuspidalklappe³. Während der Systole⁴ wird das Blut durch die Pulmonalklappe in die Lungenarterie ausgeworfen.

In der Lunge wird Blut mit Sauerstoff angereichert und gelangt schließlich über die Lungenvenen in den linken Vorhof (kleiner Kreislauf). Die Funktionsweise der linken Herzhälfte ist weitgehend mit der der rechten Herzhälfte identisch. Blut wird während der Kontraktion des Vorhofs durch die Mitralklappe⁵ in die linke Kammer gepumpt und während der Systole durch die Aortenklappe in die aufsteigende Aorta ausgeworfen. Von dort wird das Blut aufgrund der stetigen Pumpfunktion des Herzens weiter durch den gesamten Körper befördert, um die Organe mit Sauerstoff zu versorgen.

Die beiden Herzhälften sind durch ein muskulöses Septum⁶ voneinander getrennt und unterscheiden sich neben ihrer Größe (im Normalfall ist die linke Herzhälfte deutlich größer und muskulöser als die rechte, weil sie den großen Kreislauf des gesamten Körpers mit Blut versorgen muss) vor allem in den darin herrschenden Druckverhältnissen.

Das Herz selbst wird über die Koronargefäße⁷ mit Sauerstoff versorgt. Eine Durchlässigkeitsstörung und die damit verbundene Sauerstoffunterversorgung (Ischämie) am

³Dreizipflige Segelklappe

⁴Auswurfphase

⁵Zweizipflige Segelklappe

⁶Scheidewand

⁷Herzkranzgefäße

Herzmuskel führt in der Regel zum Absterben des Muskelgewebes - dem Herzinfarkt. Die Herzkranzgefäße spielen in der Kinderkardiologie allerdings nur eine stark untergeordnete Rolle.

3.1.2 Der fetale Blutkreislauf

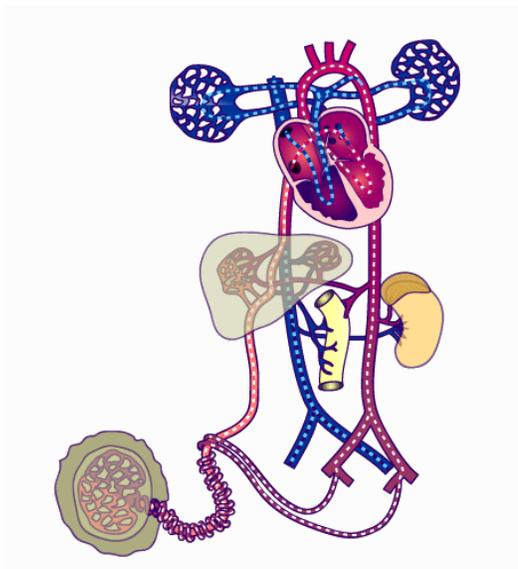


Abbildung 3.2: Überblick über den fetalen Kreislauf. Zu erkennen ist der Ductus Arteriosus, der Aorta und Lungenkreislauf miteinander verbindet sowie das Foramen Ovale - die natürliche Öffnung in der Herzscheidewand zwischen den Vorhöfen.

Im Unterschied zum erwachsenen Blutkreislauf ist der fetale Kreislauf noch nicht vollständig ausbalanciert. Die Lungen sind noch nicht voll entwickelt - die Lungengefäße haben noch eine sehr geringe Kapazität und setzen dem rechten Herz daher einen hohen Widerstand entgegen. Der Lungenkreislauf ist nicht in der Lage das gesamte Blutvolumen aufzunehmen, das ihm normalerweise angeboten würde.

Um die Balance zu wahren, gibt es daher im fetalen Kreislauf zwei Shunts⁸, die den Lungenkreislauf überbrücken. Der erste Shunt ist eine direkte Verbindung zwischen rechtem und linkem Atrium. Durch das sogenannte Foramen Ovale fließt Blut direkt von der rechten in die linke Herzhälfte und umgeht so die Lunge vollständig. Zudem werden bereits hier Voraussetzungen für später geschaffen - durch die verstärkte Belastung der linken Herzhälfte wird deren Training verbessert, und eine kräftigere Muskulatur bildet sich aus.

Allerdings ist die linke Herzhälfte nicht kräftig genug, um die Last des gesamten Blutvolumens des Körpers eigenständig zu tragen. Daher fließt auch bei intaktem Shunt eine kleine Menge des Blutes weiterhin über die rechte Hälfte des Herzens. Von hier geht der Strom über die Lungenarterie und von dort über den zweiten Shunt - den Ductus Arteriosus direkt in die Aorta (siehe Abbildung 3.2).

⁸Kurzschlüsse

3.2 Interventionelle Kardiologie



Abbildung 3.3: Schätzung der klinischen Relevanz angeborener Herzfehler im Verlauf der kommenden Jahrzehnte. [23]

Etwa jedes 100. Kind kommt in Deutschland mit einem angeborenen Herzfehler zur Welt. Das NRAH⁹ geht derzeit davon aus, dass die klinische Relevanz der angeborenen Herzfehler in den nächsten Jahrzehnten deutlich ansteigen wird (siehe Abbildung 3.3) [23].

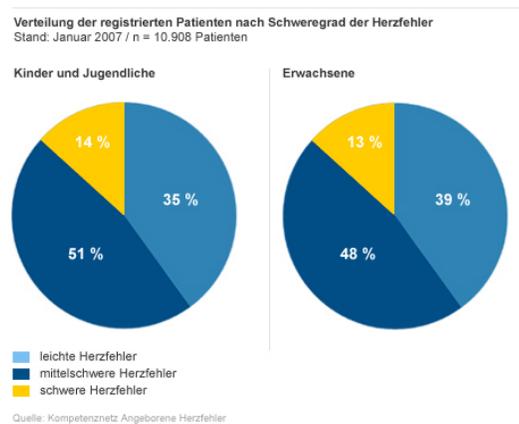


Abbildung 3.4: Schweregrad angeborener Herzfehler

Begründet wird das vor allem mit der verbesserten Behandlungssituation und neuen Erkenntnissen in der medizinischen Forschung. Auch wenn die Anzahl der Neugeborenen mit Herzfehler sich höchstwahrscheinlich nicht signifikant verändern wird, ist zu erwarten, dass sich aufgrund verbesserter Therapien der prozentuale Anteil an

⁹Nationale Register für angeborene Herzfehler

Erwachsenen mit angeborenem Herzfehler stark erhöhen wird, was bisher nicht abschätzbare Konsequenzen in der Langzeitbehandlung mit sich bringen könnte.

Ein Großteil der angeborenen Herzfehler hat nur sehr geringe Auswirkungen auf die Lebensqualität der Betroffenen. Einige Defekte bleiben gar auf Lebenszeit unentdeckt oder werden erst Jahre nach der Geburt bei Routineuntersuchungen festgestellt. Laut den Statistiken des NRAH machen die schweren (interventionell behandlungsbedürftigen) Herzfehler nur etwa 14% aller kongenitalen Herzfehlbildungen aus (siehe Abbildung 3.4).

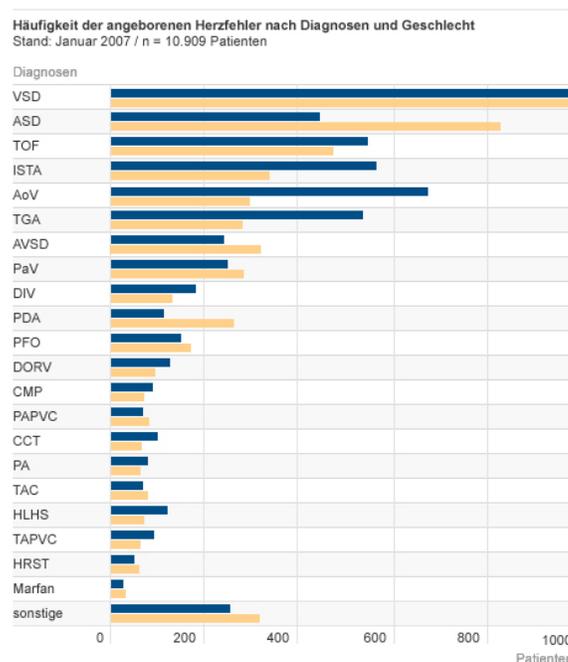


Abbildung 3.5: Häufigkeit spezifischer angeborener Herzfehler. Die blauen Linien zeigen die Anzahl der Vorkommen bei männlichen, die gelben bei weiblichen Patienten.

Betrachtet man sich die Häufigkeit der spezifischen Herzerkrankungen bei Neugeborenen (siehe Abb. 3.5), fällt auf, dass die Liste deutlich von zwei Defekten angeführt wird. ASD und VSD machen gemeinsam fast ein Drittel aller auftretenden Fälle aus. Eine Tatsache, die verstehen lässt, welche Bedeutung einer effizienten Behandlung dieser Fehlbildungen zukommt.

3.2.1 Geschichte der Katheterinterventionen

Herkömmliche Katheterinterventionen

Nachdem 1733 der englische Pfarrer Stephen Hales an einem lebenden Pferd die erste Herzkatheteruntersuchung mit dem Ziel, den Druck im Ventrikel zu messen, durchgeführt hatte, wurde erst 1844 von dem französischen Physiologen Claude Bernard der Begriff des Herzkatheters geprägt. Auf ähnliche Weise wie Hales nutzte Bernard den

Katheter, um den Druck und die Temperatur innerhalb tierischer Herzen zu messen [138].

Der erste Herzkatheter am Menschen geht auf den deutschen Urologen Werner Forßmann zurück. In einem (aus heutiger Sicht fragwürdigen) Selbstversuch führte er 1929 einen Uretherkatheter in eine Vene seines linken Armes ein und platzierte ihn durch Verschieben vor dem rechten Atrium seines eigenen Herzens. Er dokumentierte die Lage in einer Röntgenaufnahme. 1956 wurde er für sein Selbstexperiment mit dem Nobelpreis ausgezeichnet [16].

Die diagnostische Koronarangiographie wird auf Mason Sones zurückgeführt, der 1958 erstmals unter Röntgensicht eine Linksherzkatheterisierung mit Kontrastmittelapplikation zu diagnostischen Zwecken durchführte und dokumentierte [82].

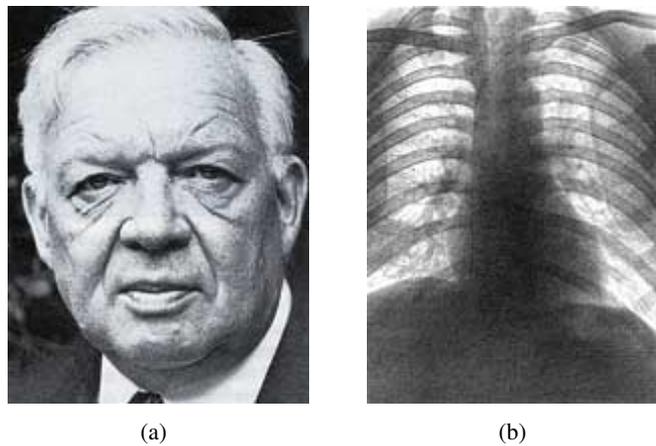


Abbildung 3.6: Dr. Werner Forßmann im Alter von 56 Jahren (a) und sein berühmtes Röntgenbild (b)

1964 entdeckte der Radiologe Charles Dotter durch einen Zufall die therapeutische Bedeutung der Katheterintervention. Beim Passieren einer peripheren Arterie einer 82-jährigen Patientin mit einem Katheter wurde das Verschlussmaterial gegen die Gefäßwand gepresst. Durch Wiederholen der Prozedur mit der Reihe nach größer werdenden Kathetern (Bougierkathetern) konnte eine vollständige Rekanalisierung erreicht werden. In dem auf diese Weise geöffneten Gefäß bildete sich eine neue, glättende Gefäßinnenhaut (Neointima) - eine Restenose fand nicht statt.

In den kommenden Jahren wurden die Methoden der Katheterintervention unter Werner Porstmann, Andreas Grüntzig und Andreas Hopf immer weiter verfeinert bis 1977 in Zürich die erste PTCA an einem wachen Patienten durchgeführt wurde. In den drei darauf folgenden Jahren wurden über 1000 weitere Eingriffe vorgenommen.

Katheterinterventionen in der Kinderkardiologie

1967 wagte Dr. Porstmann an der Ostberliner Charite erstmals den Versuch, einen persistierenden Ductus mit einem sog. Ivalon Pfropf zu verschließen. Zum damaligen Zeitpunkt waren die benötigten Katheter noch zu dick, als dass sie bei Kindern hätten eingesetzt werden können. 1979 gelang es Rashkind und Cuaso den Ductus bei

einem 3.5 kg schweren Säugling mit einer regenschirmartig aufgespannten Schaumstoffscheibe zu verschließen. 1977 bis 1986 wurden in den USA insgesamt 146 Kinder und Säuglinge mit dieser Technik behandelt. 66% davon erfolgreich.

Heute wird nur noch bei Frühgeborenen oder extrem weitem Ductus ein chirurgischer Verschluss angestrebt. Ansonsten ist die Kathetermethode das Mittel der Wahl, nicht zuletzt wegen des nur sehr kurzen Krankenhausaufenthaltes von zwei bis drei Tagen im Vergleich zu den meist mehrwöchigen Erholungsphasen, die eine herkömmliche Operation mit sich bringt.

Durch die konsequente Weiterentwicklung der kardiologischen Eingriffe mit dem Herzkatheter wurden in den letzten Jahrzehnten viele neuartige und schonende Interventionen ermöglicht. Kindern mit Herzfehlern bleiben inzwischen oft die Operation, die Herz-Lungen-Maschine, die Narben und längere Krankenhausaufenthalte erspart. In anderen Fällen kann Kindern mit Hilfe der Kathetertechnik vorübergehend geholfen werden, so dass die Operation in ein günstigeres Alter verschoben werden konnte.

Aber die Entwicklung dieser erfolgreichen Methode ist noch nicht abgeschlossen. Techniken werden weiter verfeinert, neue Möglichkeiten, sie einzusetzen, werden erprobt. Beispielsweise gelang es Ende 2001 P. Bonhoeffer, in Paris bei einem Kind biologische Pulmonalklappen einzig unter Verwendung eines Herzkatheters einzusetzen. Ein Eingriff, der noch vor wenigen Jahren ohne Operation undenkbar gewesen wäre [145].

3.2.2 Angeborene Herzfehler

Die zu entwickelnde Simulation beschäftigt sich hauptsächlich mit dem persistierenden Ductus Arteriosus. Als alternative Anwendungsfälle werden der Atrium- und der Ventrikelseptumdefekt betrachtet. Diese drei häufigen kongenitalen Herzfehlbildungen werden im Folgenden genauer vorgestellt.

Die Erläuterungen dienen dem Verständnis der späteren Ausführungen über Simulation der Katheterinterventionen. Aus diesem Grund erheben die folgenden Abschnitte keinen Anspruch auf medizinische Vollständigkeit. Besonders in den Erläuterungen zu Therapien liegt der Schwerpunkt auf der Beschreibung der simulationsrelevanten Prozeduren.

3.2.3 PDA

Terminologie

Während der Entwicklung des Fetus bis zur Geburt wird der Kreislauf des ungeborenen Kindes über die Nabelschnur mit sauerstoffreichem Blut versorgt. Die Lungen sind noch nicht entfaltet und haben ihre Funktion noch nicht aufgenommen. Aus diesem Grund hat die Evolution zwei Überbrückungsmechanismen entwickelt, die das rechte Herz vor den Überlastungen schützen sollen, die entstehen würden, wenn die rechte Kammer gegen den hohen Gefäßwiderstand der nicht entfalteten Lunge anpumpen würde.

Der erste dieser beiden Mechanismen ist der Ductus Arteriosus. Eine konisch geformte, röhrenartige Verbindung zwischen dem transversalen Teil der Aorta und der ersten Bifurkation der Pulmonalarterie. Über diese Brücke fließt vor der Geburt aus

der rechten Kammer ausgeworfenes Blut direkt über in die Aorta und von dort in den großen Körperkreislauf.

Nach der Geburt sinkt der Druck im Lungenkreislauf und der Blutfluss kehrt sich um - der DA fungiert als links-rechts Shunt. Zur gleichen Zeit beginnt die Lunge mit der Ausschüttung von Bradykinin - einem Protein, das der Kontraktion der glatten Muskulatur im DA dient und zugleich den Blutfluss senkt [167].

Normalerweise beginnt der Ductus selbstständig innerhalb von 15 Stunden nach der Geburt damit, sich zu schließen - bereits drei Wochen später ist der Übergang vollständig verwachsen. Einzig ein kleiner Verwachsungsstrang - das Ligamentum Arteriosum bleibt im erwachsenen Herz anstelle des DA erhalten.

Pathophysiologie

Verbleibt der Ductus nach der Geburt geöffnet, spricht man von einem PDA - einem persistierenden Ductus Arteriosus. Ein PDA tritt gehäuft bei Frühgeburten und Kindern auf, die während oder kurz nach der Geburt mit Atemproblemen zu kämpfen hatten. Aufgrund des geringen Sauerstoffanteils, der die Lunge erreicht, wird die Bildung des Bradykinin verringert, und der Ductus ist nicht in der Lage, sich zu kontrahieren und zu schließen. Die Druckumkehr (Aortendruck steigt, Lungendruck sinkt) bewirkt einen Blutfluss von sauerstoffreichem Blut aus der Aorta zurück in den Lungentrakt. Dementsprechend gestalten sich die Symptome, die ein offener PDA hervorruft.

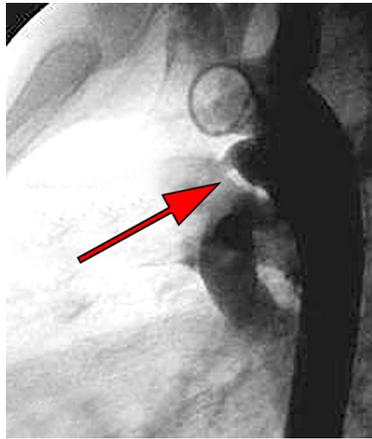


Abbildung 3.7: Ductus Arteriosus in Röntgensicht. Der Pfeil markiert die Schwelle zur Aorta, über die sauerstoffreiches Blut in den Lungenkreislauf gelangt.

Die Herzfrequenz steigt an, da das über den Ductus abfließende Blutvolumen ausgeglichen werden muss. Kurzatmigkeit, Arrhythmien und Sauerstoffunterversorgung der Organe sind häufige Folgen. Auf längere Sicht entwickelt sich oft ein vergrößerter Herzmuskel.

Die Diagnose des PDA erfolgt in der Regel über Röntgendiagnostik. EKG und Ultraschall können zur Sicherstellung von Arrhythmien und der ggf. veränderten Struktur und Bewegung des Herzmuskels hinzugezogen werden. Veränderte Herztöne können in der Regel über Auskultieren wahrgenommen werden. Die sicherste Diagnose kann

über eine invasive Angiographie des Aortentraktes mit Kontrastmitteln erfolgen.

Therapie

Bei asymptomatischen Ductus erfolgt in der Regel keine sofortige Therapie. Patienten werden zwar über lange Zeit beobachtet - ein Verschluss des PDA wird jedoch nicht angestrebt.

Symptomatische Vorkommen können sowohl invasiv als auch nicht invasiv behandelt werden. Als nicht invasive Methoden kommen vor allem medikamentöse Behandlungen mit flussverringenden Medikamenten in Frage.

Die invasive Behandlung kann chirurgisch oder katheterbasiert durchgeführt werden. Beim chirurgischen Verschluss wird der DA manuell abgebunden und so der weitere Durchfluss verhindert. Seit der Entwicklung der kathetergestützten Therapien wird aber in der Regel versucht, den chirurgischen Verschluss zu umgehen. Die Nachteile (deutlich höhere Komplikationsrate, längere Ausheilungszeit, Narbenbildung) sind zu hoch, Vorteile praktisch nicht mehr vorhanden. Nur in sehr komplizierten Fällen, die mit den herkömmlichen Katheterverfahren aufgrund abnormer Anatomien oder Komplikationen nicht zu beherrschen sind, wird die Chirurgie angestrebt.

Der perkutane Verschluss geschieht in mehreren Schritten. Zu Beginn muss der Kardiologe den Ductus visualisieren und ausmessen. Zu diesem Zweck wird eine große Körperarterie (in der Regel femoral) punktiert und ein sogenannter Pigtailkatheter durch die Aorta zur Einmündung des Ductus navigiert. Per Kontrastmittelgabe kann unter Röntgendurchleuchtung deutlich der Blutfluss von aortaler zu pulmonaler Seite dargestellt werden.

Kleine Ductus werden unter Verwendung eines Coils, eines spiralförmigen Instrumentes mit ca. 5-6 Wicklungen, verschlossen. Für diesen Eingriff, der komplett von der aortalen Seite aus geschieht, wird ein sogenannter Cobra Katheter (siehe Abbildung 3.8) in der absteigenden Aorta platziert, um sich im aortalen Abschnitt des Ductus einzuhaken. Ein weicher Führungsdraht wird durch den Katheter hindurch geschoben und mit diesem der Ductus sondiert. Weiter in den Lungentrakt eingeschoben fungiert dieser Führungsdraht als Wegvorgabe für den nun nachzuschiebenden Cobra Katheter.



Abbildung 3.8: Gekrümmt geformte Spitze eines Cobra Katheters

Ist der Katheter im Lungenast platziert, wird der Führungsdraht entfernt und der

auf einem flexiblen Mandrin¹⁰ aufgewickelte und an der Spitze eines Zubringerstabs aufgeschraubte Coil durch den Katheter vorgeschoben. Beim Zurückziehen des Mandrin und gleichzeitigem Festhalten des Coils, wird die erste Spule des Coils entfaltet. Der Katheter wird wieder auf die aortale Seite zurückgezogen, während der mit einer Wicklung leicht entfaltete Coil und der Mandrin vorerst auf pulmonaler Seite verbleiben. Anschließend werden Coil und Mandrin soweit zurückgezogen, dass die offene Wicklung des Coil sich auf der pulmonalen Einmündung des Ductus festsetzt. Ein weiteres Zurückziehen des Mandrin setzt die übrigen Wicklungen des Coils frei, die im Ductus zum Liegen kommen. Zuletzt wird der Coil durch Drehbewegungen am Zubringerdraht abgeschraubt. Der so präparierte Ductus setzt in der Regel schnell Gewebe an der rauhen Oberfläche der Coilspiralen an und wächst im Laufe der kommenden Tage und Wochen vollständig zu.

Bei größeren Ductus erfolgt der Verschluss über die venöse Seite, da der Coil im großen Ductus nicht ausreichend Halt finden würde. Ziel ist es, in diesem Fall ein konisch geformtes Drahtgeflecht - einen Ductus Occluder - im Ductus zu platzieren, der den Übergang abdichtet und mit der Zeit verwächst.

Diese Form der Intervention ist deutlich komplizierter als die auf den Coil basierende Variante. Nach der Visualisierung des Ductus von der aortalen Seite aus wird eine große Körpervene (in der Regel femoral) punktiert und ein zweiter Pigtailkatheter über die untere Hohlvene zum rechten Vorhof navigiert. Aufgrund der Lage des Vorhofs zur Kammer ist es nicht möglich, die rechte Herzkammer direkt mit dem Katheter zu erreichen.

Es gibt verschiedene Verfahren, um dennoch in die rechte Herzkammer vordringen zu können. Beispielsweise kann ein steiferer und an der Spitze stark gebogener Hilfsdraht in den Katheter geschoben werden, um das distale Ende des Katheters zu biegen. Der auf diese Weise besser navigierbare Katheter wird in die rechte Kammer eingebracht. Der Hilfsdraht wird entfernt und der Katheter mit einigen geübten Dreh- und Schiebewegungen in der Lungenarterie platziert. Hierbei ist darauf zu achten, dass das Kathetermaterial nicht die empfindliche Herzwand verletzt. Da der Eingriff unter Röntgenkontrolle erfolgt, ist es zugleich wichtig, dass die Lungenarterie schnell sondiert werden kann, um die Strahlenexposition so gering wie möglich zu halten.

Ist der Pigtailkatheter sicher in der Lungenarterie zum Liegen gekommen, wird ein weicher Draht durch den Katheter geschoben und in einem Nebenast der Lungenarterie platziert. Der Katheter wird entfernt und - den Draht als führende Wegvorgabe ausnutzend - durch einen Multi Purpose Katheter¹¹ ausgetauscht.

Mit dem Allzweckkatheter wird nun, durch eine Folge von Rotations- und Translationsbewegungen, der Ductus sondiert und zur aortalen Seite hin durchquert. Der weiche Führungsdraht wird weiter vorgeschoben und verbleibt vorerst im absteigenden Teil der Aorta.

Nach Entfernung des Allzweckkatheters wird ein breiterer, gerader Schleusen-katheter mit integriertem Führungsmandrin über den Draht geschoben und in der absteigenden Aorta platziert. Führungsmandrin und Draht werden entfernt und der auf einen Zubringerdraht aufgebrachte Ductus Occluder durch die Schleuse vorgeschoben. Der selbstentfaltende Ductus Occluder wird soweit aus der Schleuse herausgeschoben, bis

¹⁰Führungsstab

¹¹Allzweckkatheter

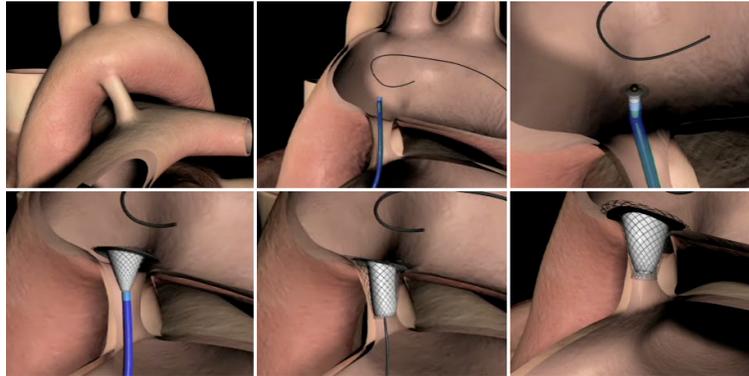


Abbildung 3.9: Platzierung eines Ductus Occluders. Die Bilder zeigen die Platzierung des Occluders durch die Schleuse, das Öffnen der distalen Retentionsscheibe, die finale Öffnung des Occluders und das Ablösen des Instruments.

sich die distale Retentionsscheibe¹² völlig entfaltet hat. Der eigentliche Körper des Ductus Occluders bleibt vorerst geschlossen. Schleusenkatheter und Occluder werden zugleich zurückgezogen, bis die Retentionsscheibe an der aortalen Seite des Ductus zum Liegen kommt. An diesem Punkt wird die Schleuse isoliert zurückgezogen, was dazu führt, dass sich der Körper des Occluder im Ductus vollständig entfaltet.

Form und Materialbeschaffenheit des Occluder sorgen für einen leichten Anpressdruck, der dafür sorgt, dass das Instrument fest im PDA verbleibt. Abschließend muss der Occluder durch Drehbewegung am Zubringerdraht abgeschraubt werden. Die Platzierung des Occluder ist in Abbildung 3.9 dargestellt.

3.2.4 ASD

Terminologie

Neben dem Ductus Arteriosus fungiert ein zweiter Shunt als Überbrückungskanal zwischen rechter und linker Herzhälfte. Das Foramen Ovale - ein kleines Loch in der Vorhofscheidewand - sorgt dafür, dass große Mengen des venösen Blutes unter Umgehung des Lungenkreislaufs direkt wieder in den linken Vorhof überfließen. Möglich ist das, weil der Fötus noch vollständig über das mütterliche Blut mit Sauerstoff versorgt wird.

Für gewöhnlich bildet sich in den letzten Tagen vor der Geburt eine dünne Hautschicht über dem FO¹³, die sich langsam verengt. Steigt nach der Geburt der Druck im rechten Vorhof an, kann das FO vollständig zuwachsen.

Pathophysiologie

Häufig ist das jedoch nicht der Fall, und ein kleines Loch bleibt in der Scheidewand bestehen. Ca. 25% aller Erwachsenen sind davon betroffen. Bleibt die Öffnung ungeschlossen, spricht man von einem Patent Foramen Ovale. Symptomatisch sind diese Defekte in der Regel unauffällig - häufig bleiben sie sogar unentdeckt. Probleme

¹²Rückhaltescheibe

¹³Foramen Ovale

können allerdings in gesonderten Situationen, etwa beim Tauchen, entstehen, wo Septumdefekte das Risiko für das Auftreten einer Dekompressionserkrankung deutlich erhöhen können [131][173].

Daneben sind in den letzten Jahren Studien veröffentlicht worden, die einen Zusammenhang zwischen PFO¹⁴ und Migränebeschwerden nahelegen. Die gegenwärtige Forschungssituation lässt es allerdings noch nicht zu, sichere Aussagen über diesen Zusammenhang zu treffen. Sollten sich die Untersuchungen allerdings bestätigen, entsteht dadurch in Zukunft möglicherweise ein weit höherer Bedarf an interventionell durchgeführten PFO Verschlüssen, als das bisher der Fall war [28][37].

Hat das Loch einen sehr großen Durchmesser (>9 mm), können hämodynamische und klinische Effekte auftreten. Durch den im Vergleich zum rechten Vorhof höheren Druck im linken Vorhof, entsteht ein links-rechts Shunt der sauerstoffreiches Blut aus dem linken Vorhof zurück in den rechten Vorhof pumpt. Bleibt ein großer ASD unkorrigiert können auf lange Zeit ein die Alveolen¹⁵ schädigender Lungenhochdruck, Rechtsherzversagen, Herzflimmern oder -flattern bis hin zum Schlaganfall die Folgen sein. Typische Symptome sind Blässe und eine geringe Ausdauer.



Abbildung 3.10: Darstellung eines ASD im Ultraschall. Der Pfeil deutet auf die Öffnung zwischen den Vorhöfen.

Die Diagnose des ASD geschieht üblicherweise über die Auskultation der Herzgeräusche und wird über Ultraschall oder Kernspinaufnahmen gesichert. Eine Ultraschallaufnahme eines ASD zeigt Abbildung 3.10.

Therapie

Bei gesichertem ASD muss zuvor festgestellt werden, ob die (zu erwartenden) Symptome einen interventionellen oder chirurgischen Verschluss des ASD ausreichend rechtfertigen, oder ob eine medikamentöse Behandlung im speziellen Fall ausreicht.

Beim chirurgischen Verschluss muss mindestens einer der Vorhöfe geöffnet und das Loch unter direkter Einsicht mit einem kleinen Gewebstück vernäht werden.

Sind die Wände des ASD ausreichend glatt und der Durchmesser nicht zu groß, kann ein perkutaner¹⁶ Verschluss angestrebt werden. Dieser geschieht vollständig über die venöse Seite und wird unter Röntgensicht durchgeführt.

¹⁴Patent Foramen Ovale

¹⁵Lungenbläschen

¹⁶durch die Haut. Gemeint: Katheterbasiert

Der Kardiologe sondiert eine große Körpervene (in der Regel femoral - allerdings sind auch brachiale oder zentrale Zugänge möglich) mit einem Katheter und platziert diesen im rechten Vorhof. Über Kontrastmittelgabe wird der Defekt aus mehreren Sichten visualisiert.

Anschließend wird ein spezieller sogenannter Messballon durch den Katheter geschoben und in der Mitte des ASD platziert. Dieser - sehr weiche - Ballon wird mit Kontrastmittel gefüllt, bis er den gleichen Durchmesser wie der ASD einnimmt.

Das Kontrastmittel wird wieder aus dem Ballon abgepumpt und der Ballon entfernt. Ausgehend von den Ergebnissen der Messung wird ein sogenannter Septaloccluder¹⁷ (siehe Abbildung 3.11) ausgewählt und - aufgebracht an das Ende eines speziellen Platzierungsdrahtes - platziert.

Die eigentliche Platzierung ist ein komplizierter Prozess, der viel Übung und Sensibilität bei der Katheternavigation erfordert. Die Prozedur sieht vor, dass der (selbst-entfaltende) Occluder erst vollständig (geschlossen) im Katheter durch das Loch hindurch geschoben wird. Anschließend entfaltet man die erste (distale¹⁸) Retentionsscheibe durch weiteres Verschieben des Occluders aus dem Katheter heraus. Das halb entfaltete Instrument wird zurück durch das Loch gezogen, wobei die aufgespannte Retentionsscheibe an der Herzscheidewand hängen bleibt und somit im linken Vorhof zum Liegen kommt.

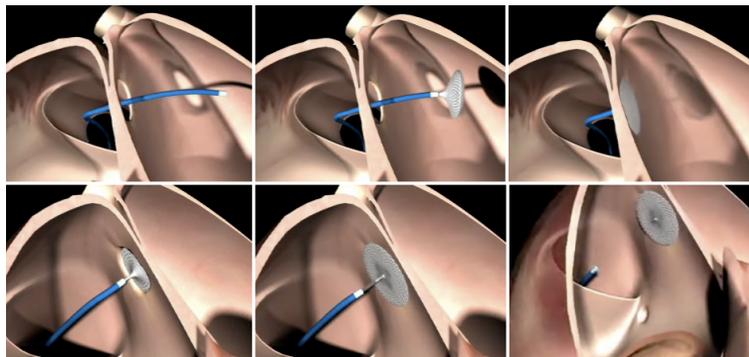


Abbildung 3.11: Platzierung eines ASD Occluders. Die Bilder zeigen die Öffnung der distalen Retentionsscheibe, das Anlegen des Occluders, die Öffnung der proximalen Retentionsscheibe und das Ablösen des Instruments.

Durch weiteres Zurückziehen des Katheters (wobei der Occluder in seiner Position am Septum liegen bleibt) entfaltet sich die zweite (proximale¹⁹) Retentionsscheibe auf der rechten Seite der Herzscheidewand. Der Durchmesser der Brücke zwischen proximalem und distalem Schirmchen sollte dabei ca. 1 mm größer sein als der Durchmesser des Septumdefektes, um einen sicheren Halt zu gewährleisten.

Der Erfolg der Platzierung wird unter Röntgensicht und mit weiterer Kontrastmittelgabe überprüft und über Ultraschallbildgebung gesichert.

Ist der Occluder vollständig entfaltet und korrekt platziert, kann er durch Drehen

¹⁷Doppelschirmchen

¹⁸entfernte

¹⁹nahe

des Zubringerdrahtes abgelöst werden. In der Regel verbleibt der Occluder im Herzen und wird mit der Zeit von neu gebildetem Gewebe überwachsen. Eine schematische Darstellung des Eingriffs zeigt Abbildung 3.11.

3.2.5 VSD

Terminologie

Während der Entwicklung der Herzmuskulatur wächst die Scheidewand, die das linke und das rechte Ventrikel voneinander trennt, von oben und unten zusammen. Ist dieser Prozess während der Entwicklung verlangsamt oder gestört, kann das zu einem verbleibenden Loch im Septum führen. Der VSD macht - in verschiedenen Formen - ca. 30% aller angeborenen Herzfehler aus und steht damit gegenwärtig an der Spitze der kongenitalen Herzdefekte.

Pathophysiologie

Die Stärke der Symptome hängt vom Durchmesser des Defekts ab. Kleine Defekte können asymptomatisch sein und bedürfen nicht zwingend einer Behandlung, während größere Defekte ähnliche Symptome wie beim Atriumseptumdefekt hervorrufen können. Durch den im Vergleich zum rechten Ventrikel erhöhten Druck in der linken Kammer wird während der Kontraktionsphase Blut durch den VSD in die rechte Kammer gepumpt. Zugleich entsteht ein Überdruck im pulmonalen System mit den damit verbundenen Symptomen. Wie bereits beim ASD beschrieben, können eine Überlastung des linken Herzens, Kurzatmigkeit, und geringe Belastbarkeit die Folgen eines VSD sein.

Diagnostiziert wird der VSD üblicherweise über Auskultation, da ein VSD-Defekt über die Veränderung der Blutströmungen eine deutliche Auswirkung auf die Herztöne hat. Daneben werden in der Regel Ultraschall (Doppler) und Angiographie (Druckmessung) zur Sicherung der Diagnose herangezogen.

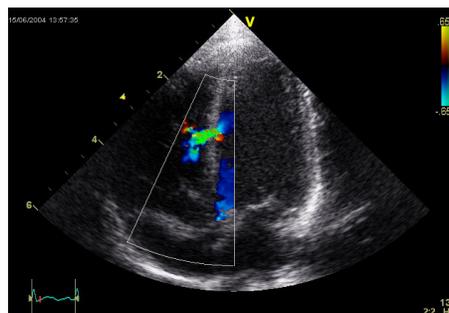


Abbildung 3.12: VSD im Farbdopplerultraschall. Die waagrechte Farbmarkierung im grünen Bereich deutet auf einen Durchfluss zwischen den Herzkammern hin.

Therapie

Wie auch schon bei der Behandlung von Ductus und ASD, gibt es auch hier die Möglichkeit, medikamentös symptomorientiert, chirurgisch oder perkutan zu behandeln. Die medikamentöse Therapie ist bei kleinen Defekten das Mittel der Wahl, da sich diese mit dem Wachstum des Herzens häufig selbst schließen. Bei mittleren und größeren Defekten hängt es vom Zustand des Patienten und der Beschaffenheit des Defektes (bsp. Wandglattheit) ab, ob eine chirurgische oder perkutane Therapie gewählt werden kann.

Im Fall eines chirurgischen Eingriffs wird unter Verwendung einer Herz-Lungen Maschine und am offenen Herzen operiert und das Loch, ggf. unter Zuhilfenahme eines zusätzlichen Gewebstücks, vernäht.

Die perkutane Methode ähnelt dem beim ASD beschriebenen Verfahren. Eine zusätzliche Schwierigkeit entsteht jedoch bei der ohnehin schon komplizierten Katheternavigation. Aufgrund von Form und Beschaffenheit des Herzmuskels ist es nicht möglich, das Loch in der Scheidewand direkt von venöser Seite aus zu sondieren. Stattdessen muss ein weicher Draht über die aortale Seite durch das Loch eingeschwenkt werden, mit einer Schlinge von venöser Seite aus gebunden und schließlich durch das venöse System zur Punktionsstelle gezogen werden.

Über diesen Draht kann dann der Septaloccluder eingebracht werden, was mit der unter ASD beschriebenen Prozedur vergleichbar ist.

3.2.6 Zukünftige Entwicklungen

In den letzten Jahren hat eine starke Entwicklung auf dem Gebiet der Katheterinterventionen immer neue Verfahren hervorgebracht. Herzfehler, die noch vor kurzer Zeit einer Operation am offenen Herzen bedurft hätten, sind heute vollständig perkutan zu behandeln. Auch für diese neuen Interventionen steigt der Trainingsbedarf täglich. Immer kompliziertere Handgriffe sind nötig - der Umgang mit neuen Instrumenten muss erlernt werden, was wiederum nach flexibel erweiterbaren Systemen verlangt.

Hinzu kommen völlig neue Technologien, die die Katheternavigation maßgeblich verändern werden. Magnettrackingsysteme sind in der Lage, Katheter und Drähte auch in komplexen, dreidimensionalen Umgebungen sicher zu steuern - allerdings werden diese Systeme zum jetzigen Zeitpunkt noch nicht in der Kinderkardiologie eingesetzt. Gründe dafür sind vor allem die unter Verwendung eines solchen Systems stark eingeschränkte Beweglichkeit der Röntgenquellen.

3.3 Bildgebung

Als bildgebende Verfahren bezeichnet man die Gesamtheit der apparativen Verfahren, mit denen beispielsweise medizinische Befunde oder physikalische und chemische Phänomene visualisiert werden.

In der Behandlungskette von der ersten Diagnostik bis zur Erfolgskontrolle der Intervention kommen primär vier bildgebende Verfahren zum Einsatz. Röntgendiagnostik wird hauptsächlich während des Eingriffs genutzt. Im Zusammenhang mit über den Katheter zu applizierendem Kontrastmittel dient sie zur Diagnostik, als optisches Kontrollinstrument und als unersetzliche Navigationshilfe.

Vor, während und nach dem Eingriff kann die Ultraschallbildgebung zusätzlich zur Diagnostik genutzt werden. Die speziellen Möglichkeiten der Blutflussdarstellung durch ein Dopplerultraschall können durch andere bildgebende Verfahren nicht oder nur mit schlechter Qualität und deutlich höherem Aufwand (MRT) ersetzt werden.

Kernspintomographie kann aufgrund der aufwendigeren Technik und der nicht vorhandenen Möglichkeit der Echtzeitdarstellung (noch) nicht während eines Eingriffs genutzt werden - wohl aber im Vorfeld und zur Nachbereitung. Aufgenommen werden dreidimensionale Volumendaten, die eine relativ sichere und vor allem projektionsunabhängige Beurteilung des Defekts erlauben.

Computertomographie spielt in der Kinderkardiologie eher eine untergeordnete Rolle. Auch wenn sich angeborene Herzfehler sehr gut unter dem CT darstellen würden, scheut man in der Regel davor zurück, die jungen Patienten der im Vergleich sehr hohen Strahlungsbelastung auszusetzen.

3.3.1 Röntgen

Technik der Bildgebung

Als Röntgenstrahlung wird Strahlung mit einer Wellenlänge zwischen 10^{-4} nm und 10 nm bezeichnet. Entdeckt wurde sie 1895 von Wilhelm Conrad Röntgen, der sie damals selbst noch als X-Strahlen bezeichnete. Dieser Terminus ist im Englischen bis heute als X-Rays erhalten geblieben. 1901 wurde er für seine Entdeckung mit dem Nobelpreis geehrt.

In einer Hochvakuumröhre (ca 7-10 *mm.Hg*) wird zwischen einer Glühkathode und einer Anode aus einer Wolfram-Rhenium-Legierung eine Hochspannung (in der Röntgendiagnostik zwischen 25 und 120 *kV*) angelegt. Aus der Kathode lösen sich Elektronen, die im elektrischen Feld zwischen Kathode und Anode Richtung Anode beschleunigt werden.

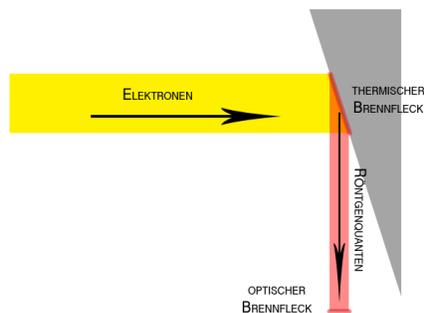


Abbildung 3.13: Schematische Darstellung der Brennflecken

Trifft das Elektronenstrahlbündel die Anode, so wird diese Fläche als *thermischer Brennfleck* bezeichnet. Die Strahlung wird - abhängig vom Anodenwinkel - auf den *optischen Brennfleck* (kurz Brennfleck) reflektiert (siehe Abbildung 3.13), wobei der überwiegende Teil der Bewegungsenergie in Wärme verwandelt wird. Nur 1% wird in Röntgenstrahlung umgewandelt, wobei man zwischen der Bremsstrahlung und der

charakteristischen Strahlung als den beiden Formen der Wechselwirkung unterscheidet. Während die charakteristische Strahlung für die Röntgendiagnostik keine weitere Rolle spielt, ist die Bremsstrahlung entscheidend für die Bildgebung.

Aufgrund der Coulombwechselwirkung zwischen Elektronen und Atomkernen wird ein Teil der Elektronen vom elektrischen Feld der Anodenkerne abgelenkt und abgebremst. Hierbei werden Lichtquanten mit einer Energie von $h * v$ abgestrahlt.

Durchqueren diese Lichtquanten Materie, wird die elektromagnetische Strahlung aufgrund von Strahlungsanregung bzw. Stoßanregung abgeschwächt. Die Materie wird durch Aufnahme der Strahlungsenergie in einen Zustand höherer Energie überführt. Die Stärke der Absorption ist durch vier Faktoren gegeben (Beispiele für einige spezifische Absorptionskoeffizienten finden sich in Tabelle 3.3.1).

1. Der Dicke des durchstrahlten Materials (Durchleuchtungslänge)
2. Der Dichte des durchstrahlten Materials (spezifisches Gewicht)
3. Der dritten Potenz der Ordnungszahl der Elemente des Materials
4. Etwa der dritten Potenz der Energie der Röntgenstrahlung.

Nachdem die Strahlung die Materie verlassen hat und entsprechend der Materienbeschaffenheit absorbiert wurde, trifft sie in der Röntgendiagnostik auf einen Bildverstärker, der die einfallenden Röntgenquanten verstärkt (nur 1 bis 2% der einfallenden Röntgenquanten treten in Wechselwirkung mit dem Detektor) und das resultierende Bild auf einem Monitor abbildet. Abhängig von der Absorption erreichen mehr oder weniger Quanten den Bildverstärker, was je nach Dichte des durchstrahlten Gewebes zu unterschiedlichen Bildern führt.

Element	Absorptionskoeffizient in cm^{-1}
Luft	$0.2 * 10^{-3}$
Wasser	0.2
Kupfer	1.4
Jod	9.6
Zinn	13.9
Blei	72.3

Tabelle 3.1: Röntgenabsorptionskoeffizienten

Großen Einfluss auf die Bildgebung hat dabei die in der Röntgenröhre angelegte Spannung. Wird eine vergleichsweise geringe Spannung (ca. 40 kV) angelegt, spricht man von weicher Strahlung. Ein Großteil der Energie wird vom Gewebe absorbiert. Feine Gewebeunterschiede, wie das etwa im Fall der Mammographie²⁰ nötig ist, können sichtbar gemacht werden. Die Strahlenbelastung ist jedoch relativ hoch.

Starke Kontrastreduzierungen, wie sie etwa bei der Untersuchung von Lungenaufnahmen nötig sind, erreicht man durch die Verwendung von harter ($> 100 kV$) Strahlung.

²⁰Röntgenuntersuchung der weiblichen Brust

Verwendung in der Kinderkardiologie

In der Kinderkardiologie wird vor allem während Katheterinterventionen auf Röntgen-diagnostik zurückgegriffen. Die Lage des Katheters im Körper muss visualisiert, Defekte dargestellt werden. Da Blut etwa den gleichen Röntgenabsorptionskoeffizienten wie das umliegende Gewebe aufweist, ist hierfür die Gabe eines geeigneten Kontrastmittels (beispielsweise Jod), das den Absorptionskoeffizienten des Bluts innerhalb der Gefäße entsprechend anhebt, nötig.



Abbildung 3.14: Auf einen C-Arm montierte Röntgenanlage

Röntgendetektor und Quelle sind während der Intervention in der Regel auf einem oder zwei beweglichen Bögen montiert (siehe Abbildung 3.14). Entsprechend der Anzahl wird die Bildgebung als mono- oder biplan bezeichnet. Unter Verwendung dieser sogenannten C-Arme können die Röntengeräte relativ frei um den Patienten rotieren und somit verschiedene Projektionen dargestellt werden.

Um die Strahlenbelastung zu minimieren, werden meist nur kurze Kontrollsequenzen aufgenommen, während Instrumente bewegt werden. Kontrastmittelvisualisierungen werden in der Regel so kurz wie möglich gehalten.

Bedeutung in der Simulation

Die vorliegende Simulation generiert aus Volumen- und Oberflächendaten in Echtzeit Bilder, die den Eindruck von Bildern einer Röntgenaufnahme vermitteln. Dabei wird ein Verfahren verwendet, das sehr schonend mit den zur Verfügung stehenden Rechenressourcen umgeht, um mehr Leistung für die physikalischen Simulationskomponenten zur Verfügung zu stellen.

Die Geometrie eines C-Arm-Systems und Projektionen werden mathematisch exakt abgebildet, andere (rechenlastigere) Merkmale wie korrekte Absorption durch aus Beobachtungen der Realität motivierte Heuristiken simuliert. Details zur Bildentstehung und Simulation finden sich in Kapitel 6.

3.3.2 Ultraschall

Technik der Bildgebung

Schall oberhalb der menschlichen Hörschwelle im Bereich zwischen 20 kHz und 1 GHz wird als Ultraschall bezeichnet. Ultraschallwellen breiten sich als Longitudinalwellen in Gasen, Flüssigkeiten und Festkörpern aus, wobei in Festkörpern aufgrund der Formelastizität auch Transversalwellen auftreten können.

Zur Erzeugung von Ultraschallwellen werden in der Regel Piezolausprecher genutzt, welche durch Umkehr des Piezoeffektes zu Schwingungen angeregt werden. Dabei handelt es sich um einen Effekt, der auftritt, wenn bestimmte Materialien unter mechanischem Druck elektrische Ladungen auf der Oberfläche ausbilden. Wird an ein solches Material (bsp. PVDF²¹) eine Wechselspannung angelegt, kommt es zu Schwingungen die zur Generierung von Ultraschallwellen genutzt werden können.

In der medizinischen Bildgebung nutzt man die Tatsache, dass Ultraschallwellen abhängig von der Zusammensetzung (bzw. dessen akustischen Impedanz) eines Gewebes an senkrechten Grenzflächen reflektiert (Übergang zwischen zwei verschiedenen Impedanzen) oder absorbiert (Durchqueren eines Gewebeabschnitts) werden.

Das Ultraschallsignal wird in kurzen Impulsen direkt an der Körperoberfläche (häufig noch durch ein gelartiges Fluid gekoppelt) von einem Sender abgestrahlt, im Körper teilabsorbiert und das reflektierte Signal anschließend von einem Empfänger registriert. Da der Reflektionskoeffizient R in der Form $R = \left(\frac{Z_2 - Z_1}{Z_2 + Z_1}\right)^2$ von der Materialimpedanz Z abhängt, ist die Reflektion beim Übergang zwischen zwei Materialien mit stark unterschiedlicher Impedanz (bsp. Luft, Fett) besonders hoch. (Tabelle 3.2 zeigt eine Auflistung verschiedener Impedanzen).

Medium	Schallgeschwindigkeit c in m/s	Schallimpedanz Z in kg/m^2s	Dichte ρ in kg/m^3
Luft	340	4104	1.2
Fett/Wasser/Hirn/Muskeln	1500	$1.5 * 10^6$	1000
Knochen (gesund)	3600	$6 * 10^6$	1700

Tabelle 3.2: Ultraschallspezifische Materialgrößen (circa Werte)

Aber nicht nur bei senkrecht zum Einfallswinkel stehenden Grenzflächen kann zurückgeworfenes Echo registriert werden. Jede Inhomogenität führt dazu, dass ein je nach Gewebestruktur charakteristischer, diffuser Strahlungskegel zurückgestreut wird, der zur Unterscheidung verschiedener Gewebetypen genutzt werden kann.

Bei der Absorption wird Energie in Wärme umgesetzt. Mit zunehmender Entfernung x vom Schallkopf nimmt die Schwächung J dabei nach der Formel $J(x) = J(0) * e^{-\mu * x}$ exponentiell zu. Dabei ist der Absorptionskoeffizient μ stark vom Gewebe und der einstrahlenden Frequenz abhängig. Das ist auch der Grund, warum die Frequenz abhängig von der benötigten Eindringtiefe gewählt werden muss. Da mit sinkender Frequenz (=höhere Eindringtiefe) aber auch die Wellenlänge größer wird, verschlechtert sich automatisch auch die örtliche Auflösungsgenauigkeit. Je nach Un-

²¹Polyvinylidenfluorid - piezoaktiver Kunststoff

tersuchungsart muss die Wahl der Frequenz also einen geeigneten Kompromiss aus Eindringtiefe und Auflösung abbilden. Übliche Frequenzbereiche in der Kinderkardiologie liegen zwischen 7.5 und 3.5 MHz.

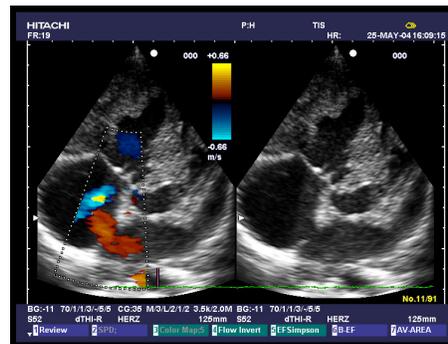


Abbildung 3.15: Kinderkardiologische Aufnahme im Farbdopplerverfahren. Die farblichen Markierungen zeigen Richtung und Geschwindigkeit des Blutflusses an.

Zusätzlich zur herkömmlichen Ultraschalluntersuchung kommt in der Kardiologie noch vermehrt das Doppler-Verfahren²² zum Einsatz. Dabei wird der Effekt ausgenutzt, dass sich Wellenlängen gegeneinander verschieben, wenn Quelle und Empfänger sich relativ zueinander bewegen. Vom unbewegten Schallkopf geht eine Welle der Frequenz f aus, die von den sich bewegenden Blutkörperchen reflektiert wird. Ein sich bewegendes Teilchen mit Flussgeschwindigkeit v reflektiert Schall mit der Frequenzverschiebung δf_1 . Die gesamte Frequenzverschiebung beträgt $\delta f = 2 * f * f(v/c) * \cos \theta$ (θ bezeichnet dabei den Winkel zwischen Teilchenbahn und Schallstrahl).

Aus dem Vorzeichen der Frequenzverschiebung lässt sich dann die Flussrichtung rekonstruieren, die im Farbdoppler durch eine entsprechend farbige Kodierung dargestellt werden kann. Bild 3.15 zeigt ein Bild aus einem solchen Farbdopplerverfahren.

Verwendung in der Kinderkardiologie

In der Kinderkardiologie findet Ultraschall vor allem in der Diagnostik von Septumdefekten seinen Einsatz. Durch das Dopplerverfahren kann die veränderte Flussrichtung des Blutes dargestellt und untersucht werden.

Nach und vor allem auch während Interventionen kann Ultraschall genutzt werden, um die Lage und die Effizienz eingesetzter Occluder zu überprüfen. Bei korrekt eingesetztem Occluder stellt sich schnell eine Änderung der Flussrichtung ein.

Bedeutung in der Simulation

Da die Simulation primär Handling und Navigation der Instrumente zum Ziel hatte, wurde der Effekt der Ultraschallsimulation bisher nicht innerhalb des Frameworks implementiert. Vorarbeiten zur Echtzeitsimulation von Ultraschall in der Kardiologie

²²Benannt nach dem Physiker Christian Andreas Doppler

(IVUS²³) wurden aber geleistet und sind ausführlich in [26], [27] und [46] nachzulesen.

3.3.3 Computertomographie (CT)

Technik der Bildgebung

Bei der Computertomographie handelt es sich um ein sogenanntes schnittbildgebendes Verfahren. Anders als beim herkömmlichen Röntgen oder Ultraschall sind nicht zweidimensionale Bilder das Ergebnis der Aufnahme, sondern ein dreidimensionales Volumen in dem Organe und Defekte genauestens lokalisiert und begutachtet werden können.

Da die Computertomographie - genau wie die Kernspintomographie nicht während einer kardiologischen Intervention, sondern nur vor und nach dem Eingriff angewandt wird, findet eine Simulation der CT Bildgebung nicht statt. Wohl aber werden die Ergebnisse dieser Untersuchung verwendet, um Diagnosen zu stellen und - im Fall der Simulation - Datensätze abzuleiten. Daher folgt an dieser Stelle nur eine sehr knappe Erklärung des Verfahrens und eine Einordnung in die Arbeit.

Die Computertomographie baut auf Röntgenbildgebung auf. Anstatt allerdings nur eine einzige Projektion aufzunehmen, werden Röntgenquelle und Detektor um den Patienten rotiert (es gibt auch Verfahren, die mit mehreren Quellen und Detektoren arbeiten, die kreisförmig um den Patienten angebracht sind), um in regelmäßigen Abständen Bilder aus verschiedenen Projektionsrichtungen aufzunehmen. Die aufgenommenen Bilder werden anschließend mit einem geeigneten Verfahren zur inversen Radon-Transformation (bsp. gefilterte Rückprojektion [197]) zu einem dreidimensionalen Volumen, bzw. den entsprechenden Schnittbildern rekonstruiert.



Abbildung 3.16: Schichtbild aus einer CT Untersuchung (hier VSD).

Der Koeffizient mit dem die generierten Röntgenstrahlen abgeschwächt werden, wird in der Computertomographie auf der Hounsfield-Skala angegeben und als Grauwert dargestellt. Tabelle 3.3 zeigt einige Beispiele für verschiedene Materialien und ihre zugehörigen Hounsfield-Werte. In der Praxis wird jeder Hounsfield-Unit ein Grauwert in der bildlichen Darstellung des CT Scans zugeordnet. Die Hounsfield Skala ist

²³Intravaskulärer Ultraschall

theoretisch zwar offen, in der Praxis jedoch auf 12 Bit (-1024 bis +3071) begrenzt. Da das menschliche Auge aber nicht in der Lage ist, die knapp über 4000 resultierenden Grauwerte zu unterscheiden, wird der Bereich der Grauwert-Darstellung je nach untersuchtem Organsystem begrenzt (Fenster-Weite). Bild 3.16 zeigt ein typisches Schichtbild aus einer CT Untersuchung.

Medium	Hounsfield-Einheiten (HU)
Luft	-1000
Wasser	0
Metall	> 1000
Knochenbälkchen	400 – 800
Knochengewebe (kompakt)	≫ 1000

Tabelle 3.3: Absorptionswerte auf der Hounsfield Skala

Vorteil des Verfahrens ist seine - verglichen mit anderen Volumenverfahren wie Kernspintomographie - hohe räumliche Auflösung. Nachteil die hohe Strahlenbelastung, der sich der zu untersuchende Patient aussetzen muss.

Schlagende Herzen werden üblicherweise über mehrere Phasen verteilt hinweg aufgenommen. Zur Synchronisierung werden die einzelnen Herzphasen per EKG ausgemessen und die CT Scans entsprechend getriggert. Dieses Verfahren versagt jedoch bei sehr schnell schlagenden Herzen und bei nicht vorhersagbaren (unregelmäßigen) Rhythmen.

Seit 2006 ist es dank eines neuartigen Computertomographen möglich, präzisere Bilder von schnell oder unregelmäßig schlagenden Herzen zu bekommen. Dazu arbeiten in sogenannten Dual-Source-Computertomographen erstmals gleichzeitig zwei rotierende Röntgenstrahler mit jeweils 64 Zeilen, um möglichst schnell Bilder zu erfassen [203].

Verwendung in der Kinderkardiologie

In der Kinderkardiologie ist der Einsatz der Computertomographie auf wenige Einzelfälle beschränkt. Aufgrund der hohen Strahlendosis beschränkt man sich in der Regel auf die niedrig aufgelöste Kernspintomographie. Erst wenn die Kernspintomographie aufgrund ihrer geringen zeitlichen Auflösung Defekte im schnell schlagenden Kinderherz (>120 Schläge pro Minute) nicht mehr ausreichend erfassen kann, und es diagnostisch erforderlich ist, wird eine computertomographische Aufnahme (unter Verwendung von Kontrastmittel) angestrebt.

Bedeutung in der Simulation

Um Herzen in der Simulationsumgebung korrekt simulieren zu können, ist eine umfangreiche Datenerfassung notwendig. Herzmodelle müssen aus realen Daten rekonstruiert werden, um entsprechend hohen Ansprüchen gerecht zu werden. Zu diesem Zweck wurden zur Modellrekonstruktion Daten aus CT Scans und Kernspintomographien verwendet und aufbereitet. Genauere Details über das Verfahren werden in Kapitel 5 geschildert.



Abbildung 3.17: Simulierte Röntgenaufnahme aus CT Volumendaten.

Abgesehen davon werden in der Simulation Kernspinaufnahmen genutzt, um umliegendes Gewebe graphisch zu simulieren (siehe Abbildung 3.17). Details hierzu sind in Kapitel 6 zu finden.

3.3.4 Kernspintomographie

Technik der Bildgebung

Die Kernspintomographie (auch Magnetresonanztomographie) ist genau wie die Computertomographie ein Schnittbildverfahren, das jedoch auf völlig anderen physikalischen Gegebenheiten aufbaut.

Anstatt Projektionen über Röntgenbildgebung aufzunehmen, nutzt man die Eigenschaft von Protonen und Neutronen, über einen Eigendrehimpuls (Spin) zu verfügen und so ein magnetisches Moment aufbauen zu können. Befindet sich ein so rotierender Kern in einem Magnetfeld B_0 , richtet sich der Kern nach B_0 aus. Der Kern verfällt in eine Präzessionsbewegung²⁴, die solange aufrecht erhalten wird, bis das Magnetfeld abgeschaltet wird, und der Kern wieder in seine ursprüngliche Lage (thermisches Gleichgewicht) verfällt.

Die Kerne rotieren im Magnetfeld mit einer festgelegten Frequenz (Lamorfrequenz). Unter Einstrahlung eines Hochfrequenzimpulses, senkrecht zum Magnetfeld, kippen die längs des äußeren Magnetfeldes ausgerichteten Spins in einen hochenergetischen Zustand wobei sich die Längsmagnetisierung temporär aufhebt. Zeitgleich synchronisiert sich die Präzessionsbewegung, und es entsteht senkrecht zu den Feldlinien des äußeren Magnetfeldes eine Transversalmagnetisierung.

Nach Abschalten des Lamorimpulses können zwei Werte gemessen werden - die Kerne richten sich wieder entlang des äußeren Magnetfeldes aus (kippen in einen niederenergetischen Zustand), wobei sie Wärme abgeben. Dieser Prozess, bei dem die Längsmagnetisierung wieder aufgebaut wird, bezeichnet man als T1-Relaxation. Da dieser Prozess im Wesentlichen von der Wärmeleitfähigkeit des Gewebes abhängt,

²⁴Kreiselartige Bewegung um die Achse des Magnetfeldes

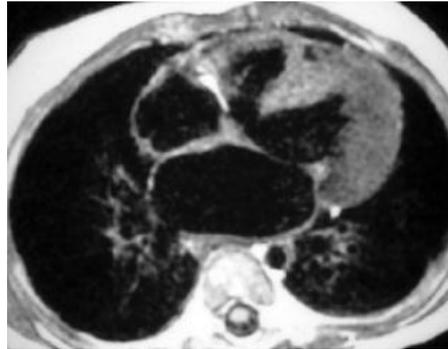


Abbildung 3.18: Schichtbild aus einer Kernspinuntersuchung (hier VSD).

stellt sich Gewebe mit schnellem Wärmetransfer (bsp. Fett) in T1-gewichteten Bildern hell dar, während Gewebe mit langsamem Wärmetransfer (bsp. Liquor) dunkel erscheint.

Die Rückbildung der durch die Synchronisierung der Präzession entstandenen Transversalmagnetisierung bezeichnet man als T2-Relaxation. Wasserstoffhaltiges Gewebe kann die Transversalmagnetisierung länger aufrechterhalten als Gewebe mit geringem Wasserstoffanteil, weshalb solches Gewebe sich in T2-gewichteten Bildern als hell darstellt.

Aufnahmen von schlagenden Herzen werden wie im CT über mehrere Phasen und per EKG-Triggerung gesteuert.

Verwendung in der Kinderkardiologie

Die Kernspintomographie ist in der Kinderkardiologie zu einem Standardwerkzeug geworden. Ein Großteil der Diagnosen kann deshalb ohne Strahlenbelastung gesichert und genauer beurteilt werden. Da die Untersuchung im Tomographen vergleichsweise zeitaufwendig ist, und Kinder/Neugeborene in der Regel eine niedrige Toleranzschwelle gegenüber der notwendigen Enge und der akustischen Belastung während der Untersuchung haben, wird die Bilderfassung meist am sedierten Patienten durchgeführt. Die Belastung durch Medikamente kann allerdings im Hinblick auf die Qualität der zu gewinnenden Ergebnisse vernachlässigt werden. Die Untersuchung wird in der Regel unter Gabe von Kontrastmittel durchgeführt.

Im direkten Vergleich zur Computertomographie fällt vor allem die wegfallende Strahlenbelastung ins Gewicht. Dem entgegen steht nur die geringere Auflösung (typischerweise auf ca. 1 mm^3 begrenzt), die allerdings in einem Großteil der Untersuchungsfälle die Aussagekraft über den Defekt nicht schmälert.

Einige der neueren Entwicklungen im Bereich der Magnetresonanz lassen hoffen, dass sie in Zukunft in der interventionellen Kinderkardiologie eingesetzt werden können. So wäre es beispielsweise möglich, dass zukünftig MRI Bilder während der Intervention mit Röntgenbildern überlagert werden, um dreidimensionale Einblicke zu ermöglichen [100].

Bedeutung in der Simulation

Analog zur Computertomographie wird die Kernspintomographie nur zur Vor- bzw. Nachuntersuchung eingesetzt. Im Rahmen dieser Arbeit wurden Herzmodelle teilweise aus Kernspindaten rekonstruiert und die Aufnahmen genutzt, um manuell erstellte anatomische Fehlbildungen mit Realdaten abzugleichen.

3.4 Mathematische Grundlagen

Um physikalische Vorgänge wie fließendes Kontrastmittel oder sich deformierende Instrumente effizient zu simulieren, bedarf es einer Vielzahl mathematischer Methoden. Dabei nehmen stabile und schnelle Lösungsverfahren für Differentialgleichungssysteme, ebenso wie Optimierungsverfahren, eine große Rolle ein. Verschiedene Formen herkömmlicher Splines werden vor allem für graphische Darstellung - Splinevolumina für glatte Deformationen genutzt.

Viele der verwendeten Methoden zielen auf die Erreichung der Echtzeitfähigkeit ab. Daher ist ein grober Überblick über die Unterschiede in Bezug auf eine Relation zwischen Genauigkeit und Performanz von Bedeutung. Da die Instrumente in der Simulation als Folge steifer, rotierbarer Elemente repräsentiert werden, ist außerdem eine Grundkenntnis über die Funktionsweise von Quaternionen hilfreich.

Ein Grundwissen in der linearen Algebra und der höheren Mathematik wird im Folgenden vorausgesetzt, die in dieser Arbeit verwendeten Verfahren aber erklärt und vorgestellt.

3.4.1 Integrationsverfahren

In der physikalischen Simulation steht man oft vor dem Problem, Anfangswertprobleme schnell, stabil und genau lösen zu müssen. Dabei handelt es sich um Problemstellungen, in denen aus einem vorgegebenen Wert y_0 zu einem Zeitpunkt t_0 und einer Differentialgleichung für $y(t)$ Funktionswerte für beliebige t berechnet werden sollen.

Ein typisches Beispiel, das auch im Zusammenhang mit der Simulation von Kontrastmittelausbreitung eine große Rolle spielt, ist die Simulation von Partikeln. Die Partikel sind in ihrer einfachsten Form durch Ort x zum Zeitpunkt t_0 , Masse m , Geschwindigkeit v und Beschleunigung a (als Resultat der auf sie wirkende Kräfte) repräsentiert. Die Berechnung des Ortes eines Partikels zum Zeitpunkt t ist dabei ein klassisches Anfangswertproblem.

In der allgemeinen Form seien $k \in \mathbb{N}$, $D \subseteq \mathbb{R} \times \mathbb{R}^{n \times k}$, $(x_0, y_0, y_1, \dots, y_{k-1}) \in D$, $f : D \rightarrow \mathbb{R}^n$. Dann heißt

$$\begin{cases} y^{(k)} = f(x, y(x), y'(x), \dots, y^{(k-1)}(x)) \\ y(x_0) = y_0, \dots, y^{(k-1)}(x_0) = y_{k-1} \end{cases} \quad (3.1)$$

Anfangswertproblem k -ter Ordnung, wobei $y^{(k)}$ die k -te Ableitung von y bezeichnet. Da Anfangswertprobleme 1. Ordnung von primärem Interesse im Zusammenhang dieser Arbeit sind und Probleme der Ordnung $k > 1$ stets auf die Ordnung 1 reduziert

werden können, kann im Folgenden ohne Beschränkung der Allgemeinheit angenommen werden, dass es sich bei den betrachteten Anfangswertproblemen um Probleme 1. Ordnung handelt.

Explizites Euler Verfahren

Das einfachste Verfahren zur numerischen Lösung von Anfangswertproblemen ist das explizite Euler-Verfahren. Zur Lösung des Problems

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0, \quad (3.2)$$

werden die ersten beiden Terme der Taylor-Reihe von y genutzt, die die lineare Approximation um den Punkt $(t_0, y(t_0))$ repräsentiert. Mittels dem Eulerverfahren nähert man sich dann durch sukzessive Anwendung der Formel

$$y_{n+1} = y_n + hf(t_n, y_n) \quad (3.3)$$

schrittweise dem gesuchten Punkt. Die Wahl der Schrittweite h ist dabei entscheidend für den Erfolg des Verfahrens. Bei großer Schrittweite wird ein Ergebnis zwar schnell erreicht, der Fehler ist aber entsprechend groß. Das Verfahren tendiert bei nicht rein linearen Funktionen schnell dazu instabil zu werden und grob falsche Ergebnisse zu liefern.

Ein Ausgleich über eine ausreichend kleine Schrittweite resultiert meist in einer Vielzahl an Berechnungsschritten und entsprechend langer Laufzeit. Abbildung 3.19 verdeutlicht das Problem skizzenhaft.

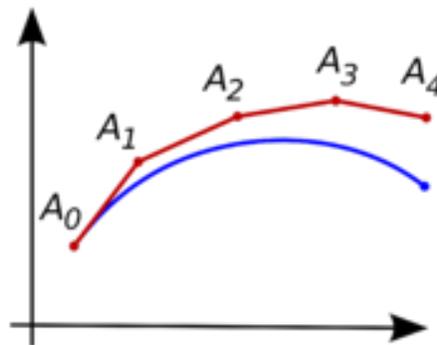


Abbildung 3.19: Entwicklung einer Funktion bei Anwendung des expliziten Euler Verfahrens. Die blaue Kurve stellt die Originalfunktion, die rote die resultierende Kurve bei Anwendung des expliziten Euler Verfahrens und Auswertung an den Punkten $A_0 - A_4$ dar.

Runge Kutta Methode

Während beim expliziten Euler Verfahren nur die Ableitung (Steigung) am aktuellen Funktionspunkt zur Berechnung des nächsten Wertes genutzt wird, errechnet sich die

Steigung bei der Familie der Runge Kutta Methoden (benannt nach den deutschen Mathematikern C. Runge und M.W. Kutta) aus der gewichteten Summe von mehreren Ableitungen im aktuellen Intervall. Die allgemeine Formel lautet:

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i, \quad (3.4)$$

wobei

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f(t_n + c_2 h, y_n + a_{21} h k_1), \\ k_3 &= f(t_n + c_3 h, y_n + a_{31} h k_1 + a_{32} h k_2), \\ &\vdots \\ k_s &= f(t_n + c_s h, y_n + a_{s1} h k_1 + a_{s2} h k_2 + \dots + a_{s,s-1} h k_{s-1}). \end{aligned}$$

Die Koeffizienten a_{ij} , b_i und c_i sind dabei allgemein so zu wählen, dass sie der Regel

$$\sum_{j=1}^{i-1} a_{ij} = c_i \forall i = 2, \dots, s \quad (3.5)$$

genügen. Da Runge Kutta eine Familie von Verfahren bezeichnet (der bei geeigneter Wahl der Koeffizienten auch das explizite Euler Verfahren angehört), kann im eigentlichen Sinne nicht von *dem* Runge-Kutta Verfahren gesprochen werden. Ist dennoch davon die Rede, ist meist eine spezielle Form des Verfahrens - auch bekannt als RK4 oder klassischem Runge Kutta - gemeint. RK4 ist definiert als

$$y_{n+1} = y_n + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (3.6)$$

wobei

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} k_1\right) \\ k_3 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} k_2\right) \\ k_4 &= f(t_n + h, y_n + h k_3) \end{aligned}$$

RK4 benötigt zwar vier Auswertungen der Funktion f , weicht aber auch bei vergleichsweise größeren Schrittweiten nicht so schnell vom gesuchten Ergebnis ab wie das explizite Euler Verfahren. Daher ist bei komplexeren Funktionen in der Regel RK4 vorzuziehen.

Eine gute und vor allem schnelle Berechenbarkeit von f ist allerdings Voraussetzung.

Verlet Integration

Die Verlet Integration weicht ein wenig vom Schema der Runge-Kutta Verfahren ab, da hier nicht die erste sondern die zweite Ableitung zur Berechnung eines neuen Funktionswertes verwendet wird. Sie stellt auch insofern einen Sonderfall dar, als dass sie hauptsächlich für die Berechnung von Partikeltrajektorien verwendet wird.

Die Funktionsweise lässt sich am besten an einem Beispiel verdeutlichen. Ein Partikel p sei definiert durch einen Ort x zum Zeitpunkt t_0 , eine Geschwindigkeit v (entspricht x') und einer Beschleunigung a (entspricht x'').

Sieht die normale Euler Integration vor, die Geschwindigkeit des Partikels durch

$$v(t_0 + \Delta t) = v_0 + a_0 \Delta t \quad (3.7)$$

und die resultierende Position durch

$$x(t_0 + \Delta t) = x_0 + v_0 \Delta t \quad (3.8)$$

zu berechnen, wird in der Verlet Integration die Berechnung der Geschwindigkeit durch eine Approximation aus aktueller und vergangener Position berechnet. Die resultierende Formel lautet

$$x(t_0 + \Delta t) = x_0 + (x_0 - x(t_0 - \Delta t)) + a_0 \Delta t^2 = 2x_0 - x(t_0 - \Delta t) + a_0 \Delta t^2 \quad (3.9)$$

Wird die Geschwindigkeit zusätzlich benötigt, so lässt sie sich anhand einer zweiseitigen numerischen Ableitung im nachfolgenden Zeitschritt durch

$$v_0 = \frac{x(t_0 + \Delta t) - x(t_0 - \Delta t)}{2\Delta t} \quad (3.10)$$

approximieren. Die Verlet Integration bietet gegenüber der expliziten Euler Integration vor allem den Vorteil der erhöhten Stabilität. Da die erste Ableitung explizit aus den Positionen berechnet wird, ergibt sich nicht die bei der Euler Integration übliche starke Diskrepanz zwischen berechneten und eigentlichen Werten. Als Folge wachsen die resultierenden Werte nicht mehr in einen nicht mehr kontrollierbaren Bereich an.

Die Verlet Integration bietet bei höheren Schrittweiten zwar nicht die gleiche Genauigkeit wie RK4, ist aber in der Regel schneller zu berechnen und wird auch bei großen Schrittweiten nicht instabil. Solange eine leicht verminderte Genauigkeit der Berechnung also kein Ausschlusskriterium darstellt, und die Berechnungsgeschwindigkeit eine große Rolle spielt, ist die Verlet Integration in der Regel RK4 und dem expliziten Euler Verfahren vorzuziehen.

3.4.2 Splines

Der Begriff Spline bezeichnet in der Mathematik eine bestimmte Klasse von Funktionen, die jeweils stückweise aus Polynomen zusammengesetzt sind. Dabei handelt es

sich bei einem Spline vom Grad n um eine Funktion, die aus Polynomen mit maximalem Grad n zusammengesetzt ist.

Splines werden unter anderem dazu genutzt, glatt zwischen Knotenpunkten zu interpolieren, was darin begründet ist, dass die verschiedenen Arten von Splines an ihren Knotenpunkten diverse, wünschenswerte Eigenschaften ($n - 1$ fach stetige Differenzierbarkeit) aufweisen. Im Folgenden werden einige Formen von Splines und die verwandten Splinevolumina vorgestellt, die in der Simulation vor allem zur Darstellung von Instrumenten und Berechnung von Deformationen Anwendung finden.

Hermite Splines

Eine einfache Form von Splines sind die Familie der Kubischen Hermite Splines (denen als Subfamilie auch die Kardinal Splines und Catmull Rom Splines angehören). Benannt nach dem französischen Mathematiker Charles Hermite, handelt es sich bei den Hermite Splines um Splines der dritten Ordnung ($k = 3$), bei denen sich jedes der Splinepolynome in einer Hermite Form darstellen lässt.

Die Hermite Form besteht aus zwei Kontrollpunkten und zwei Kontrolltangenten pro Polynom. Der Spline wird in Subintervalle unterteilt, wobei jedes Subintervall k genau zwischen zwei Punkten x_k und x_{k+1} liegt. Die Kurve zwischen x_k und x_{k+1} ist parametrisiert auf dem Bereich $(0,1)$. Die Berechnung der Splinekurve geschieht wie folgt:

Auf jedem Subintervall mit Startpunkt $p_0 (= x_k)$, Endpunkt $p_1 (= x_{k+1})$ und den Start- und Endtangenten m_0 und m_1 ist die verbindende Kurve gegeben durch:

$$p(t) = h_{00}(t)p_0 + h_{10}(t)m_0 + h_{01}(t)p_1 + h_{11}(t)m_1 \quad (3.11)$$

wobei t , aufgrund des normalisierten Intervalls als $t \in [0, 1]$ definiert ist, und die Hermite Basisfunktionen als

$$\begin{aligned} h_{00}(t) &= 2t^3 - 3t^2 + 1 \\ h_{01}(t) &= t^3 - t^2 + t \\ h_{10}(t) &= -2t^3 + 3t^2 \\ h_{11}(t) &= t^3 - t^2 \end{aligned}$$

definiert sind. Sind die Tangenten m_0 und m_1 nicht von extern vorgegeben, lassen sie sich durch die angrenzenden Punkte interpolieren. Die zu x_k gehörende Tangente m_k berechnet sich in dem Fall durch

$$m_k = \frac{1}{2}(1 - c)(x_{k+1} - x_{k-1}). \quad (3.12)$$

In diesem Fall spricht man von Kardinalsplines. Der Parameter c nimmt Einfluss auf die Länge der Tangente und darüber direkt auf die Glattheit des resultierenden Hermite Splines. Wird für c der Wert 0 gewählt, bezeichnet man den Spline als Catmull-Rom Spline.

Hermite Splines sind vor allem schnell zu berechnen und zeichnen sich dadurch aus, dass jeder Kontrollpunkt durchquert wird. Sie eignen sich also vor allem für Approximationen von Wegen deren direkte Kontrollpunkte als Wegpunkte bekannt sind.

Bezierkurven

Bezierkurven sind parametrisch modellierte Kurven, die Anfang der 1960er Jahre von den beiden Mathematikern Pierre Bezier und Paul de Casteljaeu entwickelt wurden.

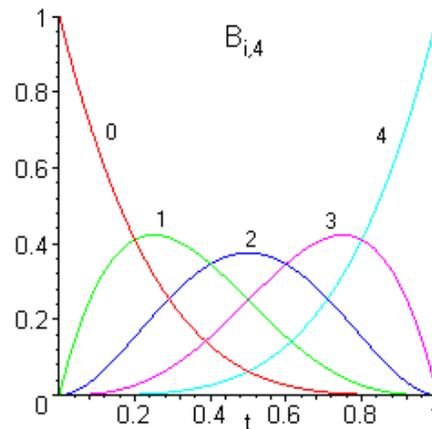


Abbildung 3.20: Die 5 Bernstein Basispolynome vom Grad 4. Deutlich ist zu sehen, dass sich die Summe der Polynomwerte an jedem Punkt zu genau 1 aufaddiert.

Eine Bezierkurve (oder Bezierspline) n -ten Grades erfordert mindestens $n + 1$ Kontrollpunkte p_i , die gemeinsam das sogenannte Kontrollpolygon bilden. Wie bei den oben beschriebenen Hermitesplines ist es abschnittsweise definiert und wird im Intervall $t \in [0, 1]$ beschrieben durch

$$p(t) = \sum_{i=0}^n B_{i,n}(t)p_i \quad (3.13)$$

$B_{i,n}$ bezeichnet dabei das i -te Bernsteinpolynom n -ten Grades, welches gegeben ist durch

$$B_{i,n} = \binom{n}{i} t^i (1-t)^{n-i} \quad (3.14)$$

Abbildung 3.20 zeigt exemplarisch die Bernsteinpolynome $B_{i,4} \forall 0 \leq i \leq 4$. Deutlich ist zu erkennen, dass die zu einem Grad gehörigen Bernsteinpolynome sich an jedem Punkt zu genau 1 aufaddieren.

Im Gegensatz zu Hermite Polynomen passieren Bezierkurven nicht jeden Kontrollpunkt und sind je nach Grad nicht als lokal einzustufen, da auch entfernte Kontrollpunkte noch Einflüsse auf die Kurve haben. Daher sind sie nur sehr bedingt zur

Approximation von durch Punkte gegebene Kurven geeignet, wohl aber zur sehr glatten Interpolation, wie sie zum Beispiel in Deformationsvolumina (siehe Kapitel 5.4) zur Anwendung kommt.

Splinevolumina

Die eindimensionalen Bezierkurven lassen sich mittels ineinander verschachtelter Summen zu Splinepatches oder Splinevolumina erweitern.

Ein zweidimensionales Splinepatch vom Grad n berechnet man demzufolge mit der Formel

$$p(s, t) = \sum_{i=0}^n \sum_{j=0}^n p_{i,j} B_{i,n}(s) B_{j,n}(t) \quad (3.15)$$

Konsequent die Formel erweiternd, kann ein dreidimensionales Splinevolumen über

$$p(s, t, u) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n p_{i,j,k} B_{i,n}(s) B_{j,n}(t) B_{k,n}(u) \quad (3.16)$$

berechnet werden. Die Splinepatches bzw. -volumina weisen die gleichen Eigenschaften wie normale Splinekurven auf. Sie sind $n - 1$ fach stetig differenzierbar und werden in der Regel zur glatten Interpolation genutzt.

3.5 Physikalische Grundlagen

Hauptaufgabe der Simulation ist es, die Wirklichkeit eines beschränkten Systems abzubilden. Um dieses Ziel zu erreichen, müssen physikalische Gegebenheiten analysiert und anschließend in berechenbare Modelle überführt werden.

3.5.1 Physik der Instrumente

Um eine realistische Simulation von Kathetereingriffen zu ermöglichen, ist es vor allem notwendig, ein möglichst exaktes und in der physikalischen Realität begründetes Modell der Instrumente aufstellen zu können. Zu diesem Zwecke werden im Folgenden einige grundlegende Ansätze vorgestellt, die in ihrer Folge zu der in dieser Arbeit gegebenen Formulierung der Instrumentensimulation führen werden.

Hamiltonsches Prinzip

Das Hamiltonsche Prinzip (in seiner Verallgemeinerung auch als "Prinzip der kleinsten Wirkung" bekannt) sagt aus, dass die Entwicklung $q(t)$ eines Systems, dessen Zustand durch N generalisierte Koordinaten $\mathbf{q} = (q_1, q_2, \dots, q_N)$ beschrieben werden kann, vom Zustand $\mathbf{q}_1 \stackrel{\text{def}}{=} \mathbf{q}(t_1)$ zum Zustand $\mathbf{q}_2 \stackrel{\text{def}}{=} \mathbf{q}(t_2)$ zu den Zeiten t_1 und t_2 ein Minimum des Aktionsfunktionals

$$\mathcal{S}[\mathbf{q}(t)] \stackrel{\text{def}}{=} \int_{t_1}^{t_2} L(\mathbf{q}, \dot{\mathbf{q}}, t) dt \quad (3.17)$$

darstellt. $L(\mathbf{q}, \dot{\mathbf{q}}, t)$ bezeichnet dabei die Lagrangsche Funktion des Systems. Stark vereinfacht und weniger formal ausgedrückt besagt das Prinzip, dass ein System in der Natur immer den Drang dazu hat, extramal oder optimal abzulaufen. Eine rollende Kugel wird beispielsweise immer der steilsten Neigung nach beschleunigen - ein Lichtstrahl durch unterschiedliche Medien immer den Weg der geringsten Eigenzeit gehen und damit die Gesetzmäßigkeiten der Brechung und Reflexion erfüllen.

3.5.2 Bewegungsgleichung

Da das Wirkungsintegral (3.17) einen Minimalwert einnehmen soll, und somit in erster Ableitung 0 ergibt ($\delta\mathcal{S} = 0$) führt eine funktionale Ableitung des Wirkungsintegrals über der Lagrangschen Systemfunktion zu einer Gleichung der Form²⁵:

$$\frac{\partial L}{\partial \mathbf{q}} - \frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} = 0. \quad (3.18)$$

,auch Euler-Lagrange Differentialgleichung genannt.

Unter Zufügung eines dissipativen Terms (hier Reibung) und einer Darstellung mit kartesischen anstelle von generalisierte Koordinaten erhält man die bekannte Bewegungsgleichung ohne Zwangsbedingungen:

$$m \frac{\partial x}{\partial t^2} + k \frac{\partial x}{\partial t} + F(x) = 0 \quad (3.19)$$

m bezeichnet darin die Masse eines (vorläufig punktförmigen) Objekts, k den auf die Bewegung wirkenden Reibungskoeffizienten. $F(x)$ sei die Summe aller inneren und äußeren auf das Objekt wirkenden Kräfte an Stelle x . Demnach gilt, dass die Bewegung eines (punktförmigen) Objektes immer im Gleichgewicht zu den wirkenden Kräften steht, bzw. äquivalent die Energie bezogen auf die Bewegung minimiert wird.

Bei der Simulation von Instrumenten werden allerdings keine punktförmigen Objekte betrachtet - insofern ist es nötig, nicht nur Bewegungen sondern auch Deformationen durch die zugrundeliegenden Gleichungen abzubilden. Hierfür wird die Elastizitätstheorie genutzt.

Elastizitätstheorie

Elastizität beschreibt die Eigenschaft von Stoffen, unter dem Einfluss von Kräften ihre Form zu verändern. Kehrt ein Körper nach Wegfallen der Krafteinwirkung wieder in seine ursprüngliche Form zurück, spricht man von einem elastischen Prozess mit Einwirkung auf einen elastischen oder dehnbaren Körper. Bleibt die Form des Körpers verändert, wird der Prozess als inelastisch bezeichnet.

²⁵Der Beweis ist in der bekannten Literatur nachzulesen, weshalb an dieser Stelle auf eine weitergehende Herleitung verzichtet wird.

Das Verhalten von elastischen Körpern wird durch das Hookesche Gesetz²⁶ approximiert²⁷. Es geht davon aus, dass die elastische Verformung proportional zur anliegenden mechanischen Spannung ist. Für die Systeme, die dem Hookeschen Gesetz unterliegen, ist die Deformation immer proportional zur anliegenden Kraft und es entsteht bei einem dreidimensionalen System mit anliegenden Spannungen P_{1-3} entlang der Hauptachsen und der Poissonschen Elastizitätszahl μ eine Deformation der Form:

$$e_1 = \frac{1}{M} \{P_1 - \mu(P_2 + P_3)\} = \frac{1 + \mu}{M} \left\{ P_1 - \frac{\mu}{1 + \mu} (P_1 + P_2 + P_3) \right\} \quad (3.20)$$

$$e_2 = \frac{1}{M} \{P_2 - \mu(P_3 + P_1)\} = \frac{1 + \mu}{M} \left\{ P_2 - \frac{\mu}{1 + \mu} (P_1 + P_2 + P_3) \right\} \quad (3.21)$$

$$e_3 = \frac{1}{M} \{P_3 - \mu(P_1 + P_2)\} = \frac{1 + \mu}{M} \left\{ P_3 - \frac{\mu}{1 + \mu} (P_1 + P_2 + P_3) \right\} \quad (3.22)$$

$$(3.23)$$

M bezeichnet darin das lineare Elastizitätsmodul des Materials [105]. Vereinfacht ausgedrückt gilt:

$$\vec{F} = -k \vec{x} \quad (3.24)$$

F bezeichnet dabei die anliegende Kraft. x ist die Distanz, um die das Material gedehnt wird und k beschreibt eine Elastizitäts- oder Federkonstante.

Die Energie E , die dementsprechend in einem gedehnten Material aufgebaut ist, lässt sich beschreiben durch

$$E = \frac{1}{2} k x^2 \quad (3.25)$$

3.5.3 Lösung der Bewegungsgleichung

Die eigentliche Simulation geschieht durch die Lösung der Bewegungsgleichung. Im Allgemeinen gibt es dafür drei grobe Ansätze, deren primäre Vor- und Nachteile je nach Anwendungsgebiet gegeneinander abgewogen werden müssen.

Direkte Minimierung des Hamiltonfunktional

Der offensichtliche Weg entspricht der direkten Minimierung des Hamiltonfunktional. Da diese Minimierung im kontinuierlichen Fall aufgrund von Rechenleistung oder fehlender Formulierungen in der Regel keine Option darstellt, kann die Gleichung diskretisiert und durch den Verzicht auf möglicherweise zu vernachlässigende Teile weiter vereinfacht werden.

²⁶nach Sir Robert Hooke (englischer Physiker)

²⁷Approximiert, da das Hookesche Gesetz auf kleine Deformationen beschränkt ist. Bei größeren Deformationen verliert das Hookesche Gesetz in der Regel eine Gültigkeit

Eine derartig denkbare Vereinfachung wäre der Verzicht auf die zeitliche Abhängigkeit, was einer sogenannten quasi-statischen Formulierung entspricht. Voraussetzung hierfür ist, dass Trägheit und Reibung im Vergleich zu den übrigen auftretenden Kräften nur eine verschwindend geringe Rolle spielen.

Im Fall der Instrumentensimulation ist das der Fall, da sowohl die Trägheit (Instrumente haben eine sehr kleine Masse, was zu einem entsprechend kleinen Trägheitsmoment führt) als auch die Reibung (Instrumente sind speziell beschichtet um große Reibungskräfte zu vermeiden) in Relation zu den Elastizitätskräften gesetzt, an Bedeutung verlieren. In der vorliegenden Arbeit wird daher eine quasi-statische Formulierung des Problems verwendet, um Instrumente realistisch zu simulieren.

Weiterhin muss dabei darauf geachtet werden, dass bei einer multimodalen Funktionsfläche die Lösung (die Form des Instruments) so weit eingegrenzt (bzw. durch evtl. Randbedingungen spezifiziert) werden sollte, dass sich das Attraktionsgebiet immer innerhalb des gleichen Minimums bewegt.

Finite Elemente Methode

Eine weitere Möglichkeit eine solche Differentialgleichung zu lösen, besteht in der Anwendung der Finite Elemente Methode (FEM). Zu diesem Zweck wird die ursprüngliche Differentialgleichung in ihrer schwachen Variationsform reformuliert und die gesuchte Funktion in endlich kleine und endlich viele (finite), stückweise lineare Teilgebiete diskretisiert. Mittels dieser Diskretisierung wird durch die FEM das kontinuierliche Problem in ein diskretes Problem überführt, um anschließend eine approximative Lösung berechnen zu können, wobei sich die approximierte Lösung mit Erhöhung des Diskretisierungsgrades der exakten (kontinuierlichen) Lösung annähert.

Zu diesem Zweck wird für jedes der Elemente eine dem Problem entsprechende mathematische Beschreibung gewählt, wobei darauf zu achten ist, dass an den Übergängen zwischen den Elementen (problemabhängig zu spezifizierende) physikalisch motivierte Stetigkeitsbedingungen einzuhalten sind.

Resultierend erhält man ein Gleichungssystem, das mit geeigneten Verfahren (in der Regel) numerisch gelöst werden kann.

Lösung der Euler Lagrange Differentialgleichung mit numerischen Verfahren

Ist eine schnelle Lösung gefragt, bei der nicht zwingend die Genauigkeit im Vordergrund steht, ist es häufig möglich, die Differentialgleichung zu diskretisieren und dabei komplexe Terme wegfällen zu lassen. Anschließend kann die Gleichung unter Verwendung numerischer Verfahren (bsp. Runge-Kutta, s.o.) gelöst werden. Diese - auch als Feder Masse Modelle - bezeichneten Vorgehensweisen bringen häufig schnelle Lösungen, leiden aber, je nach Ansatzgebiet, unter der Vereinfachung der Differentialgleichungen.

Für viele physikalische Probleme ist es nicht ohne erhebliche Abweichungen von der Realität möglich, die komplexeren Terme zu übergehen. Für einfache Bewegungen wie beispielsweise (punktförmige) Partikel ist der Ansatz aber ein gangbarer und sinnvoller Weg.

3.5.4 Fluidsimulation

In der Simulation kardiologischer Interventionen spielt der Blutfluss, bzw. dessen mathematische Abbildung eine große Rolle. Defekte und Gefäße werden durch Kontrastmittel visualisiert, welches wiederum mit dem fließenden Blut verwirbelt und durch den Körper transportiert wird.

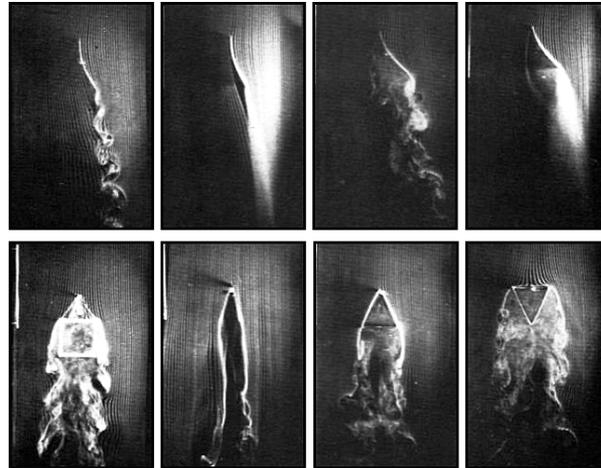


Abbildung 3.21: Verschiedene Anteile laminarer und turbulenter Strömungen in Flussbildern

Bei der Simulation von Kontrastmittelfluss muss zuerst zwischen laminaren (parallel ausgerichteten Strömungsgeschwindigkeitsvektoren) Strömungen, wie sie in kleinen Gefäßen vorherrschen und turbulenten (differierend ausgerichteten Strömungsgeschwindigkeitsvektoren) Strömungen, die vermehrt in Hohlräumen (Herzkammern) und großen Gefäßen (Aorta) auftreten, unterschieden werden. Abbildung 3.21 zeigt eine Gegenüberstellung verschiedener Anteile laminarer und turbulenter Strömungen.

Laminare Strömungen

Solange die Ausbreitung des Kontrastmittels näherungsweise laminar geschieht, lässt sich das Ohmsche Gesetz aus der Elektrodynamik entsprechend adaptiert für die Hämodynamik nutzen. In der Kardiologie tritt dieser Effekt beispielsweise, aber nicht darauf beschränkt, in den Koronarien auf.

Ein Volumenstrom F (Stromstärke I im Ohmschen Gesetz) bestimmt sich demnach aus dem Strömungswiderstand R (elektrischem Widerstand) und Druckdifferenz δp zwischen Quelle (Herzkammer) und Senke (Kapillaren/Venen). Die Druckdifferenz entspricht im Ohmschen Gesetz der Spannung U .

Hinzukommend entspricht der Volumenstrom dem Volumen, das pro Zeiteinheit [mls^{-1}] durch ein Gefäß mit dem Querschnitt $A = \pi \cdot r^2$ strömt. Er ergibt sich aus dem Querschnitt des Gefäßes und der gemittelten linearen Strömungsgeschwindigkeit \vec{v} . Es folgt:

$$\mathcal{F} = \frac{\delta p}{R} = A \cdot \vec{v} = \pi \cdot r^2 \cdot \vec{v} \quad (3.26)$$

Aufgrund der Inkompressibilität (nahezu) von Flüssigkeiten, ändert sich der Volumenstrom entlang eines nicht verzweigenden Gefäßes nicht. Bei variierendem Querschnitt eines Gefäßes, verändert sich die Strömungsgeschwindigkeit antiproportional zum Durchmesser. Beschrieben wird dieses Verhalten in der Kontinuitätsgleichung:

$$\mathcal{F} = A_1 \cdot \vec{v}_1 = A_2 \cdot \vec{v}_2 = \textit{konstant} \quad (3.27)$$

Solange konsequent laminare Strömungen angenommen werden, kann also in der Simulation die Strömungsgeschwindigkeit anhand einer Betrachtung der Durchmesseränderung innerhalb eines Gefäßabschnittes berechnet werden.

Bei laminaren Strömungen entsteht ein so genanntes parabolisches Geschwindigkeitsprofil, das sein Maximum im Axialstrom aufweist. Herrscht keine innere Reibung F_{innere} (was in der laminaren Strömung angenommen wird), entspricht der Zusammenhang zwischen maximaler und mittlerer Fließgeschwindigkeit $\vec{v} = \frac{1}{2} \cdot v_{maximal}$.

Turbulente Strömungen

Treten bestimmte äußere Bedingungen ein (beispielsweise eine nicht glatte Struktur der Gefäßoberfläche, ein nicht röhrenförmiges Profil oder erhöhte Strömungsgeschwindigkeiten), kann die laminare Strömung in eine turbulente Strömung mit erheblich höherer innerer Reibung umschlagen. Schichten, die vorher übereinander gegliedert sind, vermischen sich, und es treten Wirbel auf (siehe Abbildung 3.21).

Aufgrund der auftretenden Abflachung des parabolischen Gefäßprofils beschleunigt sich die mittlere Fließgeschwindigkeit und ergibt sich fortan aus $\vec{v} \approx 0.8 \cdot v_{maximal}$.

Auch hängt die Stromstärke nicht mehr linear, sondern annähernd proportional zur Quadratwurzel vom Druckgefälle ab. Daraus folgt, dass für eine Verdopplung der Stromstärke ein ca. um das Vierfache erhöhter Druck benötigt wird.

Die Wahrscheinlichkeit für das Auftreten einer Turbulenz wird durch die dimensionslose Reynoldszahl Re beschrieben, die vom Radius r , der mittleren Fließgeschwindigkeit \vec{v} , der Massendichte ρ sowie der Viskosität η abhängt.

$$Re = \frac{2r \cdot \vec{v} \cdot \rho}{\eta} \quad (3.28)$$

Der Wert, bei dem ein Übergang von laminarer zu turbulenter Strömung zu beobachten ist, wird allgemein als *kritische Reynoldszahl* bezeichnet. Er ist nicht berechenbar und muss für jedes Strömungsproblem individuell und experimentell ermittelt werden. Auch ist der Übergang nicht zwingend, sondern stark von der Anfangsbedingung der Strömung abhängig, so dass die Reynoldszahl nur als Tendenz zum Auftreten einer turbulenten Strömung gewertet werden kann.

Zur Simulation turbulenter Strömungen ist eine Kenntnis der sogenannten Navier-Stokes²⁸-Gleichungen unentbehrlich.

Die Navier-Stokes-Gleichungen beschreiben das Verhalten von Fluidsubstanzen (Flüssigkeiten, Gasen) unter Bewegung. Für die meisten Probleme nehmen sie die Form nichtlinearer, partieller Differentialgleichungen an, was ihre Lösung in den meisten Fällen schwierig gestaltet. Bis heute ist es nicht gelungen, die Existenz einer globalen Lösung für die allgemeinen Navier-Stokes Gleichungen nachzuweisen²⁹.

Die allgemeinen Navier-Stokes Gleichungen sind, ausgehend von der Erhaltung von Masse, Moment und Energie in einem begrenzten Kontrollvolumen, beschrieben als:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \eta \Delta \mathbf{u} + (\lambda + \eta) \nabla(\nabla \cdot \mathbf{u}) + \mathbf{f} \quad (3.29)$$

In dieser Gleichung bezeichnet ρ die Dichte des Fluids, u die Geschwindigkeit, die Stoffkonstanten η und λ die (bekannte) Viskosität, p den Druck und der Vektor \mathbf{f} die Volumenkraftdichte (Gravitation oder Corioliskraft).

Glücklicherweise kann im Falle einer Kontrastmittelsimulation von inkompressiblen Fluids ausgegangen werden, die sich dadurch charakterisieren, dass die Dichte entlang der Teilchenbahnen konstant ist.

$$\nabla \cdot \mathbf{u} = 0 \quad (3.30)$$

Zusammen mit dem Erhaltungssatz der Masse

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (3.31)$$

ergeben sich die vereinfachten Navier-Stokes-Gleichungen für inkompressible Fluids:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0 \quad (3.32)$$

Dabei entspricht ν einer Konstante für die Viskosität. Zum besseren Verständnis der Gleichung hilft es, sie in ihre Einzelkomponenten zerlegt zu betrachten.

$$\overbrace{\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}}^{\text{Masseträgheit}} = \underbrace{-\nabla p}_{\text{Druck Gradient}} + \underbrace{\nu \nabla^2 \mathbf{u}}_{\text{Viskosität}} + \underbrace{\mathbf{f}}_{\text{Weitere Kräfte}} \quad (3.33)$$

Instationäre Beschleunigung
Konvektive Beschleunigung

²⁸Benannt nach dem französischen Physiker Claude-Louis Navier und dem irischen Mathematiker George Gabriel Stokes

²⁹Das Problem, einen solchen Beweis zu erbringen, steht auf der Liste der sieben wichtigsten offenen Probleme der modernen Mathematik. Seine Lösung ist vom Clay Mathematics Institute mit einem Preisgeld von 1.000.000\$ dotiert (Millenium Prize Problems).

Da, wie oben bereits beschrieben, in der Regel keine analytische Lösung der gegebenen Gleichungen existiert (Kriechströmung stellt eine der wenigen Ausnahmen aus der Realität dar), werden die Navier-Stokes Gleichungen meist unter Zuhilfenahme verschiedener numerischer Verfahren gelöst.

3.6 Computergrafik & Geometrie

Viele in der Simulation verwendete grundlegende Techniken stammen aus dem Gebiet der Computergrafik. Große Gefäße und Kavernen³⁰ sind (unter anderem) durch dreidimensionale Dreiecksnetze repräsentiert. Instrumente werden aus verknüpften und gegeneinander rotierten Gelenken berechnet. Ein grundlegendes Verständnis der linearen Algebra und geometrischer Algorithmen wird daher im Folgenden vorausgesetzt, spezialisiertere Verfahren werden aber eingehend erläutert.

3.6.1 Quaternionen

Quaternionen wurden erstmals 1853 von dem irisch-englischen Mathematiker William Rowan Hamilton beschrieben. Ähnlich der komplexen Zahlen stellen sie eine Erweiterung der reellen Zahlen - allerdings mit drei imaginären Anteilen (i.d.R. repräsentiert als i , j und k) - dar. Da Quaternionen sehr nützliche Eigenschaften in Bezug auf Rotationen im dreidimensionalen Raum haben, finden Sie eine weit verbreitete Anwendung in der Computergrafik.

Ein Quaternion ist ein Quadrupel $q = (a, b, c, d)$ mit $a, b, c, d \in \mathbb{R}$. Aufgrund Ihres Entdeckers, wird die Menge der Quaternionen mit \mathbb{H} bezeichnet. Addition, Subtraktion und Multiplikation sind auf der Menge der Quaternionen wohldefiniert und lassen sich mit der Kenntnis einiger Regeln leicht herleiten.

Betrachtet man Quaternionen analog zu den komplexen Zahlen, so lassen sie sich in der Form

$$q = a + b \cdot i + c \cdot j + d \cdot k \quad (3.34)$$

schreiben. Mithilfe der sogenannten *Hamilton-Regeln*:

$$i^2 = j^2 = k^2 = -1 \quad (3.35)$$

$$i \cdot j = k, j \cdot k = i, k \cdot i = j \quad (3.36)$$

$$j \cdot i = -k, k \cdot j = -i, i \cdot k = -j \quad (3.37)$$

können Addition:

$$q_0 + q_1 = (a_0 + a_1, b_0 + b_1, c_0 + c_1, d_0 + d_1) \quad (3.38)$$

³⁰Hohlräume

Subtraktion:

$$q_0 - q_1 = (a_0 - a_1, b_0 - b_1, c_0 - c_1, d_0 - d_1) \quad (3.39)$$

und Multiplikation:

$$q_0 \cdot q_1 = ((a_0 \cdot a_1 - b_0 \cdot b_1 - c_0 \cdot c_1 - d_0 \cdot d_1), \quad (3.40)$$

$$(a_0 \cdot b_1 + b_0 \cdot a_1 + c_0 \cdot d_1 - d_0 \cdot c_1), \quad (3.41)$$

$$(a_0 \cdot c_1 - b_0 \cdot d_1 + c_0 \cdot a_1 + d_0 \cdot b_1), \quad (3.42)$$

$$(a_0 \cdot d_1 + b_0 \cdot c_1 - c_0 \cdot b_1 + d_0 \cdot a_1)) \quad (3.43)$$

für zwei gegebene Quaternionen $q_0 = (a_0, b_0, c_0, d_0)$ und $q_1 = (a_1, b_1, c_1, d_1)$ definiert werden.

Ebenso analog zu den komplexen Zahlen existieren für jede Quaternion $q = (a, b, c, d)$ eine konjugierte Quaternion $\bar{q} = (a, -b, -c, -d)$, ein Betrag $|q| = \sqrt{q \cdot \bar{q}}$ und eine Inverse $q^{-1} = \frac{\bar{q}}{|q|^2}$, was eine Division über Multiplikation mit der Inversen ermöglicht.

Die in der Computergrafik meist genutzte Eigenschaft von Quaternionen ist die Möglichkeit, Rotationen durch sie auszudrücken. So repräsentiert die Quaternion $q = (\cos(\frac{\alpha}{2}), x \cdot \sin(\frac{\alpha}{2}), y \cdot \sin(\frac{\alpha}{2}), z \cdot \sin(\frac{\alpha}{2}))$ eine sogenannte Rotationsquaternion für eine Drehung um den Winkel α um den (normierten) Vektor (x, y, z) .

Die eigentliche Rotation für den Punkt p berechnet sich dann nach der Formel:

$$p_{rot} = q \cdot p \cdot \bar{q} \quad (3.44)$$

Zur Multiplikation einer Quaternion mit dem dreidimensionalen Vektor p , kann die Quaternion $q = (a, b, c, d)$ folgendermaßen in eine 3×3 Matrix M umgewandelt werden:

$$M = \begin{pmatrix} 1 - 2b^2 - 2c^2 & 2ab - 2cd & 2ac + 2bd \\ 2ab + 2cd & 1 - 2a^2 - 2c^2 & 2bc - 2ad \\ 2ac - 2bd & 2bc + 2ad & 1 - 2a^2 - 2b^2 \end{pmatrix} \quad (3.45)$$

Für eine genaue Herleitung der gegebenen Formeln sei an dieser Stelle auf die gängige Literatur verwiesen. Der große Vorteil der Rotationsdarstellung über Quaternionen, verglichen mit alternativen Möglichkeiten der Rotationsberechnung (Achsen Winkel Notation, Euler Winkel), ist die Möglichkeit, zwischen Rotationen - und somit auch Rotationsmatrizen auf herkömmliche Weise interpolieren zu können.

Eine alternative, intuitivere Betrachtungsweise für Quaternionen liefert die geometrische Algebra. Da dieser Ansatz für die vorliegende Arbeit aber nicht weiter relevant ist, soll an dieser Stelle nur auf die entsprechenden Literaturquellen (siehe [97]) verwiesen werden.

3.7 Grafische Simulation

Neben der Berechnung der Physik, wird ein Großteil der Simulation durch spezialisierte Grafikalgorithmen bestritten, die zum größten Teil direkt auf der Grafikhardware ausgeführt werden. In diesem Abschnitt wird eine kurze Einführung in die verwendete Grafikschnittstelle - OpenGL - gegeben.

3.7.1 OpenGL

Der Begriff OpenGL (Open Graphics Library) bezeichnet eine ursprünglich von SGI³¹ entwickelte und spezifizierte Schnittstelle zur Programmierung von 2D und 3D Computergrafik. Der erstmals 1992 veröffentlichte Standard wird seither stetig vom OpenGL ARB³² zertifiziert und erweitert.

Microsoft, das sich als Unternehmen im März 2003 aus dem ARB verabschiedet hat, entwickelt seit 1995 mit Direct3D einen auf ähnlichen Konzepten aufbauenden, direkten Konkurrenten zur OpenGL Schnittstelle. Da im Rahmen dieser Arbeit aber ausschließlich OpenGL (Die unterstützten Plattformen - Linux und Mac OS X bieten keine Unterstützungen für den Direct3D Standard) verwendet wurde, soll auf einen ausführlichen Vergleich und eine Besprechung von Direct3D an dieser Stelle verzichtet werden.

OpenGL bietet Softwareentwicklern die Möglichkeit, Hardwarefunktionen der Grafikkarte direkt aufzurufen und somit Geometrien, Texturen und Spezialeffekte auf dem Bildschirm zu rendern. Derzeit sind dazu ca. 250 Befehle im OpenGL Standard zertifiziert. Da die Implementierung der API in der Regel aber vom Hersteller der Grafikkarte übernommen wird, stehen diesen 250 offiziellen Befehlen noch eine herstellerabhängige Menge weiterer, nicht vom ARB zertifizierter und stärker an die Hardware gebundener Befehle gegenüber.

OpenGL stellt, aus Sicht des Programmiermodells gesehen, eine große Zustandsmaschine dar. Der Programmierer setzt Parameter, die die Darstellung der Objekte während des Renderingvorgangs beeinflussen. Diese Parameter können beispielsweise Texturierungen, Farbwerte oder Skalierungen repräsentieren. Jeder Parameter bleibt, getreu dem Muster eines Zustandsautomaten, so lange erhalten, bis er erneut verändert oder zurückgesetzt wird.

Seit der ersten Vorstellung hat sich der OpenGL Standard stark verändert. 3D Texturen, Tiefenformate und die Verwendung mehrerer Renderziele sind dabei nur einige, wenige Beispiele, die dem Standard über die Jahre hinzugefügt worden sind. Mit Einführung der Ende 2007 erwarteten Version 3.0 (Longs Peak) wird zudem die festverdrahtete Funktionalität der Renderpipeline (die Verkettung der Berechnungseinheiten, durch die geometrische Informationen transformiert und in Pixelwerte gewandelt werden) endgültig abgeschafft. Funktionen, die auf Pixel und Vertices³³ angewendet werden sollen, müssen dann als individuelle Shaderprogramme vorliegen.

Für eine eingehendere Besprechung von OpenGL und die Erläuterung der relevanten Fachbegriffe sei an dieser Stelle auf die entsprechende Fachliteratur verwiesen, die zur Reflektion des wandelnden Standards in regelmäßigen Abständen vom ARB herausgegeben wird [208][170][178].

Shader

Shader stellen eine noch direktere Programmierschnittstelle zur Grafikkarte als der OpenGL Standard dar. Durchlief bis vor wenigen Jahren noch jeder Vertex und jeder resultierende Pixel die gleichen Stufen der Renderingpipeline (Matrixtransformati-

³¹Silicon Graphics

³²Architecture Review Board

³³3D Koordinaten

on, Rasterisierung, Beleuchtung...), so wird die Funktionalität heute durch sogenannte Shaderprogramme bestimmt.

Dabei unterscheidet man zwischen Vertex- und Fragmentshadern, die entsprechend ihrem Namen Einfluss auf die Berechnungen der Vertices³⁴ und Fragmente³⁵ haben. Zur Programmierung haben sich, neben Assembler, die C ähnlichen Hochsprachen Cg (Nvidia), HLSL (Microsoft), Renderman (Pixar) und GLSL (OpenGL ARB) auf dem Markt etabliert.

Im Rahmen dieser Arbeit wurde aufgrund der größten Plattformunabhängigkeit GLSL als Shadersprache gewählt. Die später vorgestellten Algorithmen lassen sich aber mit wenig Aufwand in jede der anderen Sprachen übersetzen.

³⁴3D Koordinaten mit zusätzlichen Informationen

³⁵Pixel mit Tiefeninformation

Kapitel 4

Stand der Forschung

Das nachfolgende Kapitel widmet sich dem aktuellen (Januar 2008) Stand der Technik. Um eine Vollständigkeit zu gewährleisten, wird zu Beginn kurz der aktuelle Stand in der medizinischen Ausbildung vorgestellt. Anschließend werden verwandte und vergleichbare Simulationssysteme eingeordnet und ihre Unterschiede (und Gemeinsamkeiten) zur vorliegenden Arbeit soweit möglich diskutiert.

Im darauf folgenden Kernteil wird insbesondere auf die Simulation von Instrumenten, Kontrastmittel und Grafik sowie auf weitere technisch relevante Themen eingegangen und derzeitige Entwicklungen besprochen.

4.1 Simulationssysteme

Der allgemeine Nutzen von Simulationen in der medizinischen Aus- und Fortbildung wurde in einer Vielzahl von Studien immer wieder untersucht und bestätigt [19][200][78][134][44][96][181][33][202][76]. Generell wurde gezeigt, dass Interventionszeiten verkürzt, das Auftreten von durch menschliches Versagen induzierten Komplikationen verringert und das Komplikationsmanagement erheblich verbessert werden können. Die Schlussfolgerung - darüber sind sich die Autoren der diversen Veröffentlichungen einig - ist eine Notwendigkeit der Verstärkung der simulationsgestützten Ausbildung in der Medizin.

4.1.1 Nutzen in der Kinderkardiologie

Speziell in der Kinderkardiologie liegen über den Einsatz von Simulatoren mangels geeigneter Trainingssysteme noch nicht ausreichend Daten vor, um eine qualitative Untersuchung zu rechtfertigen.

Wohl gibt es aber Forschungsergebnisse, die die Erfahrung von Kardiologen mit der Komplikationsrate und der durchschnittlichen Dauer von Eingriffen korrelieren und darüber Rückschlüsse auf den zu erwartenden Nutzen der Simulation in diesem Bereich zulassen [171][43][102].

Die aktuellen Trainingsrichtlinien für Katheterinterventionen im pädiatrischen Bereich sehen so zum Beispiel klar definierte Minimalwerte für Basis- und Fortgeschrittenentraining in der Aus- und Weiterbildung interventionell arbeitender Kardiologen

Gesamtinterventionen pro Jahr	100
Davon interventionell	20
Art der Intervention	
Septumdefektmessung mit Ballon	5
Andere	k.A.

Tabelle 4.1: Empfohlene minimale Anzahl an jährlichen Basisinterventionen [102].

(siehe Tabelle 4.1.1 und 4.1.1) vor, [102] um einen, die routinierte Behandlung des Patienten gewährleisten, Erfahrungsstand erhalten zu können.

Gesamtinterventionen pro Jahr	200
Davon interventionell	100
Art der Intervention	
Septumdefektmessung mit Ballon	5
Transseptale Punktion	10
Pulmonalklappendilatierung	10 (5 Neugeborene)
Aortenklappendilatierung	10 (5 Neugeborene)
Pulmonalarteriendilatierung	10
Pulmonalarterienstenting	10
Coarkialdilatierung	10
Coarkialstenting	5
Coarkialverschluss	10
Collateralverschluss	10
Ductusverschluss	10
ASD Verschluss	10

Tabelle 4.2: Empfohlene minimale Anzahl an jährlichen Fortgeschritteneninterventionen [102].

4.1.2 Schulungssysteme

In der konventionellen kardiologischen Ausbildung wird schon seit längerer Zeit auf diverse Simulationen und simulationsähnliche Hilfsmittel zurückgegriffen. Es gibt anatomische Präparate und spezielle Bücher, die mit Hilfe von Stereobrillen einen räumlichen Eindruck der Anatomie ermöglichen [205] [141]. Aufgrund fehlender Interaktivität sind diese Methoden allerdings eher unter der Rubrik Lehr-/Schulungssysteme einzustufen.

4.1.3 Mechanische Modelle

Mechanische Modelle sind ein derzeit in der Medizin noch weit verbreitetes Ausbildungsinstrument. Die Modelle haben allerdings meist den Nachteil, dass sie teuer, schlecht wartbar und in der Regel statisch oder nur schlecht animiert sind.

Die Firma Mecora [21] stellt beispielsweise ein anatomisches Koronarmodell mit Namen *CoroSim* her, das zur Simulation in der interventionellen Kardiologie und neuerdings auch in der pädiatrischen Kardiologie eingesetzt werden kann (siehe Abbildung 4.1).



Abbildung 4.1: Pulsatiles Corosim Modell der Firma Mecora

Die maßstabsgetreue Abbildung des menschlichen Herzens (mit Aorta und den wichtigsten Koronargefäßen) ist elastisch, transparent und wird pulsatil mit Wasser durchströmt. Der mechanische Simulator kann sowohl direkt visuell als auch im Katheterlabor und unter Röntgendurchleuchtung genutzt werden.

CoroSim ist ein anatomisches Koronarmodell zur Simulation für die interventionelle Kardiologie. Es bietet eine maßstabsgetreue Abbildung der wichtigsten Koronarien. Diese sind elastisch, transparent und werden pulsatil mit Wasser durchströmt. Somit eignet sich CoroSim gut zur Schulung interventioneller Eingriffe. An den integrierten Stenosen kann sowohl visuell, als auch unter radiologischen Realbedingungen im Katheter-OP die Positionierung von Kathetern und Stents geübt werden [21].

Die Nachteile des Modells liegen vor allem in der Natur kongenitaler Herzfehler begründet. Da viele der angeborenen Defekte topologische Änderungen mit sich bringen, wird für jeden Defekt ein neues Modell benötigt. Zudem ist das Modell vollständig geschlossen - einmal implantierte Plastiken können also nur mit sehr hohem Aufwand wieder aus dem Modell entfernt werden. Die Pumpanimation wird von vielen Kardiologen als nicht ausreichend beschrieben und die Elastizität der Gefäße ist laut der Meinung befragter Kardiologen weit von der Wirklichkeit entfernt.

Ein weiterer Aspekt ist das (um realen Bedingungen gerecht zu werden) benötigte Katheterlabor. Abgesehen von Kosten, die bei Belegung der Geräte entstehen, ist zu beachten, dass sowohl der Auszubildende als auch der Ausbilder während des Trainings der gleichen Röntgenstrahlung ausgesetzt sind, wie das bei einer realen Intervention der Fall wäre. Ein Anfänger kann also nicht beliebig oft und lange üben, weil die Richtwerte für Strahlenbelastung einzuhalten sind.

Ein vergleichbares Modell mit ähnlichen Nachteilen wird von der Firma Sawbones [22] vertrieben (siehe Abbildung 4.2).

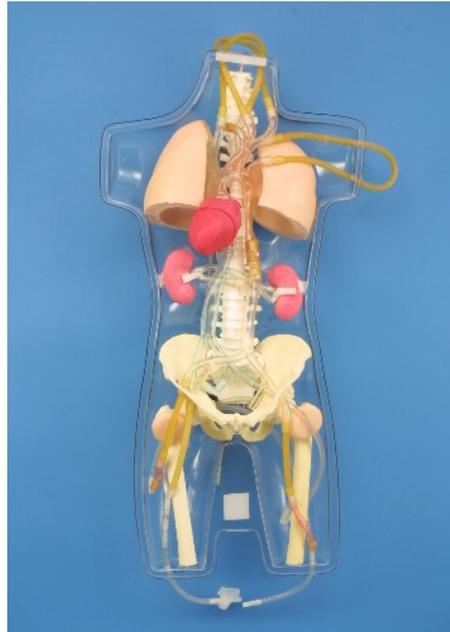


Abbildung 4.2: Pulsatiles Modell der Firma Sawbones

4.1.4 Tierversuche

Eine weitere Möglichkeit, Eingriffe zu trainieren, bildet der Tierversuch. Allerdings haben nur wenige Tiere eine dem Menschen so ähnliche Anatomie, dass sie gegenüber einem mechanischen Modell einen ernsthaften Mehrnutzen hätten. Das Tier mit der in Bezug auf Kardiologie vergleichbarsten Anatomie ist das Schwein, an dem auch Tierversuche zur Ausbildung durchgeführt werden [174].

Abgesehen davon, dass Schweine aber sehr viel anfälliger für Durchblutungsstörungen als Menschen sind, sind Tierversuche teuer und aufwendig. Wie beim mechanischen Modell können kongenitale Herzfehler nicht einfach erzeugt, sondern müssten gegebenenfalls chirurgisch implantiert werden. Das Problem der Belastung durch Röntgenstrahlung ist wie beim mechanischen Modell gegeben.

Betrachtet man hinzukommend noch die ethische Problematik (Tierversuche sind in vielen Ländern verboten, in Deutschland ist der Tierschutz seit 21. Juli 2002 Teil des Grundgesetzes), wird schnell deutlich, dass der Tierversuch bei bestehenden Alternativen keine geeignete Methode für die Ausbildung mehr darstellen kann.

4.1.5 Ausbildung am Patienten

Das derzeitige Mittel der Wahl für die Ausbildung in der Kinderkardiologie stellt die direkte Ausbildung am Patienten dar [102]. Der Anfänger lernt, indem er - unter professioneller Überwachung - selbst am Patienten Eingriffe durchführt und Erfahrung sammelt. Auch wenn dadurch natürlich ein sehr guter Trainingseffekt erreicht wird, braucht es in der Regel eine Vielzahl von Eingriffen, bis aus dem Anfänger ein erfahrener Arzt mit entsprechender Routine geworden ist. Zeit, in der Patienten unter

längeren Untersuchungszeiten (Röntgenstrahlung, Kontrastmittelverbrauch), erhöhten Komplikationsraten und (im Extremfall) falschen Diagnosen oder unangemessenen Therapien zu leiden haben.

Davon abgesehen, werden Anfänger hauptsächlich auf Standardfällen geschult. Patienten mit anatomischen Besonderheiten kommen im klinischen Alltag nur selten vor und können dann auch nur von wenigen Studenten untersucht oder gar behandelt werden [154].

4.1.6 Simulationssysteme

Große Hoffnungen für eine zukünftige effizientere Ausbildung ruhen derzeit auf den verschiedenen Simulationssystemen. In der herkömmlichen Kardiologie haben sich derzeit fünf Systeme (Cathi, CathLabVR, Vist, ANGIO Mentor und SimSuite) mit verschiedenen Stärken, Schwächen und Trainingsstrategien auf dem Markt etabliert. Daneben existieren noch Forschungsergebnisse zu den experimentellen Systemen ICard, daVinci und EVE von denen die beiden erstgenannten aber nicht mehr in ihrer ursprünglichen Form als Simulationssysteme weiterentwickelt werden.

Auf dem kinder-kardiologischen Sektor ist das Feld noch weitgehend leer - einzig SimSuite bietet zur Zeit eingeschränkt die Simulation einer ASD Intervention (allerdings derzeit für den beschränkten Fall von Erwachsenenherzen) an. Laut Aussagen der einzelnen Hersteller (mit Ausnahme CATHI GmbH) besteht zur Zeit auch kein gesteigertes Interesse für diesen Markt, da er im Vergleich zur traditionellen interventionellen Kardiologie einen deutlich erhöhten Forschungs- und Kostenaufwand, aber nur ein vergleichsweise geringes finanzielles Potential bietet.

Es folgt eine kurze, beschreibende Auflistung der derzeit verfügbaren Systeme - die Angaben beruhen (soweit nicht anders durch Quellen angegeben) auf persönlichen Beobachtungen¹ und Experimenten. Technische- und Implementierungsdetails können daher, wenn überhaupt, meist nur sehr grob angegeben werden. Aufgrund der Natur dieser Arbeit, beschränken sich die hier dargestellten Beobachtungen auf technische Gegebenheiten.

CATHI GmbH - Cathi

Das Cathi System[20] (siehe Abbildung 4.3) befindet sich als ursprünglich universitäres Projekt seit 2000 in Entwicklung[119]. 2003 wurde es von der zu diesem Zweck gegründeten CATHI GmbH übernommen.

Cathi verwendet Trainingsdaten, die aus biplanen Röntgendatensätzen rekonstruiert wurden[164]. Die Röntgensimulation der Gefäßdarstellung wird (auf der CPU) exakt gerechnet und stützt sich für die finale Darstellung auf eine SDL Umgebung. Durchleuchtungslängen werden exakt berechnet und abgebildet. Die Animation der Gefäße geschieht auf Keyframebasis.

Das System ist derzeit auf die Simulation von Eingriffen in den Koronarien beschränkt und spezialisiert. Eine Kathetersimulation findet nicht statt, Drähte und andere Instrumente innerhalb der Koronarien folgen der Mittellinie der Gefäße. Es gibt keine entsprechenden physikalischen Berechnungen.

¹Stand zur PCR 2007 in Barcelona

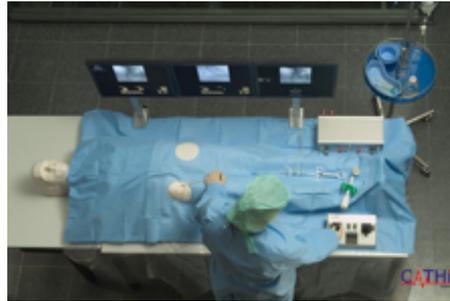


Abbildung 4.3: Cathi Simulator der Firma CATHI GmbH

Das Tracking der Instrumente geschieht optisch, Force Feedback ist in der aktuellen Version passiv (Variabilität des mechanischen Widerstands) möglich. Ein aktives Force Feedback (Rückstellkraft) ist derzeit nicht geplant - für das Gros an Katheterinterventionen aber laut Aussage diverser Kardiologen auch nicht notwendig.

Immersion - CathLabVR

Das CathLabVR System ist der Nachfolger des AccuTouch Endovascular Simulators vom selben Hersteller [8]. Immersion sieht seine Stärken laut eigenen Angaben vor allem im realistischen Force Feedback, das von Kardiologen jedoch häufig als zu stark empfunden wird. Eine mögliche Begründung dafür ist der Ursprung des mechanischen Teils des Systems in der Endoskopie, wo die Wirkung der Rückstellkräfte um ein Vielfaches stärker empfunden wird als in der Kardiologie.



Abbildung 4.4: CathLabVR Simulator der Firma Immersion

Dort hat die Firma auch ihre Wurzeln mit einer Vielzahl an Patenten zur Umsetzung von kontrolliertem User Feedback. Neben CathLabVR bietet Immersion noch eine Reihe weiterer Simulatoren im medizinischen Umfeld (Endoskopie, Koloskopie...) an.

Über das CathLabVR System selbst gibt es leider keine wissenschaftlichen Veröffentlichungen - auch konnte es nicht auf der PCR 2007 begutachtet werden. Die Herstellerangaben versprechen branchentypisch ein sehr gutes Katheterverhalten und

eine sehr realistische grafische Umsetzung. Auf der Homepage von Immersion veröffentlichte Screenshots (siehe 4.4) und Videos [9] lassen jedoch Zweifel an diesen Behauptungen aufkommen.

Bei einer qualitativen Studie (Stand Juni 2007, Veröffentlichung Dezember 2007 unter Prof. Dr. G. Völker) in der CathLabVR mit anderen Systemen verglichen wurde, konnte eine stark eventbasiert arbeitende Instrumentensimulation beobachtet werden.

Beispielsweise führt eine Annäherung an ein Zielgebiet (Einmündung der Koronarien) ohne weiteres Zutun des Nutzers zu einer Sprung und einer Verankerung des Katheters. Nicht begründbare Sprünge, teilweise über Gefäßgrenzen hinweg, waren auch an anderen Stellen der Intervention zu beobachten.

Die Darstellung des Hintergrundes und der Gefäße geschieht über Rendering von Drahtgittermodellen und ist daher nur eine grobe Annäherung an die Realität. Durchleuchtungslängen und Kontrastmittelkonzentrationen werden vernachlässigt, was zu einer vollständigen Schwärzung von durchflossenen Bereichen führt.

Neben den Eingriffen in den Koronarien lassen sich mit dem System auch Herzschrittmacherinterventionen und Eingriffe in den Carotiden simulieren. Eingriffe in den Herzkammern sind nicht möglich.

Mentis - VIST

Das VIST Trainingssystem[13] (siehe Abbildung 4.5) beruht auf dem 2000 von Dawson et al. vorgestellten ICTS (Interventional Cardiology Training System) Simulator [151][56][63]. Seit der Kommerzialisierung des Systems gab es keine weiteren Veröffentlichungen über die technischen Details.

Das System erlaubt neben Koronarinterventionen auch die Simulation von Herzschrittmacherinterventionen sowie Eingriffe in den Carotiden und im neurologischen Bereich. Wie auch schon bei CathLabVR sind Eingriffe in den Herzkammern nicht möglich.

VIST verwendet Trainingsdaten aus 4D CT- oder MR-Daten, die in einer OpenGL Umgebung durch animierte NURBS² Oberflächen dargestellt werden. Die Darstellung des Kontrastmittels geschieht über die Opazitätswerte der Oberflächen - die Durchleuchtungslänge wird nicht berücksichtigt. (Die Dichte hängt nicht von der relativen Kontrastmittelkonzentration ab - dünne Gefäße erscheinen in der gleichen Dichte wie dickere Gefäße). Verwirbelungen treten nicht auf.

Der Röntgenhintergrund wird bei VIST als Oberflächengrafik berechnet und anschließend durch Shader mit diversen heuristischen Effekten versehen. Ein animierter Hintergrund wird durch Keyframes im Hintergrundmesh realisiert (Zwerchfellbewegung), korreliert jedoch nicht mit der Herzbewegung. Es gibt im VIST System keine Volumengrafik.

Soweit ersichtlich beruht die Instrumentenberechnung auf einem Feder-Massemodell (Siehe 4.2.3). Die Reaktionsgeschwindigkeit ist entsprechend gut, der Realitätsgrad in großen Gefäßen (bsp. Aorta) aber nach eigener Beobachtung und der Aussage von erfahrenen Kardiologen eingeschränkt. Nach Aussage eines VIST Mitarbeiters (Stand 2007) wird ein Freiheitsgrad (nach eigener Beobachtung vermutlich Torsion) vernachlässigt. Teilweise konnten ungenau getrackte und vor allem zwischen

²Non Uniform Rational B-Spline



Abbildung 4.5: VIST Simulator der Firma Mentice

den Gefäßen springende Instrumente beobachtet werden.

Zudem fährt die virtuelle Röntgenkamera automatisch die Position der Katheterspitze an - nach eigenen Beobachtungen wird jeweils nur ein distaler Bereich des Katheters berechnet. Weiter hinter liegende Bereiche fallen aus der Berechnung heraus.

Hinzukommend wurde beschrieben, dass während der Navigation in der Aorta Schwierigkeiten, die in der Natur auftreten, nicht im Simulator reproduzierbar waren und der Katheter sich stark an einem optimalen Weg orientiert.

Das Tracking der Instrumente geschieht mechanisch, Instrumente werden im System deutlich spürbar von einer Art Teleskoparm gegriffen und darüber geführt. Systembedingt ist daher selbst bei sehr feinen Drähten ein Widerstand spürbar.

Über den positiven Nutzen beim Einsatz des VIST Trainingssystems gibt es eine Reihe quantitativer und qualitativer Untersuchungen [111][51].

Simbionix - ANGIO Mentor

ANGIO Mentor (siehe Abbildung 4.6) ist von Aufbau und Nutzbarkeit dem VIST System sehr ähnlich.



Abbildung 4.6: ANGIO Mentor Simulator der Firma Simbionix

Laut Aussage eines Simbionix Mitarbeiters (Stand Juni 2007) ist auch der AI-

gorithmus zur Katheterberechnung größtenteils mit dem des Vist Systems identisch und teilt daher dessen Defizite. Eine experimentelle Prüfung auf der PCR 2007 in Barcelona bestätigt dies. Das bei VIST auftretende Springen der Instrumente konnte jedoch nicht reproduziert werden, was eine Weiterentwicklung des Algorithmus vermuten lässt. Auch bei ANGIO Mentor fällt auf, dass das System optimierten Wegen zu folgen scheint.

Graphisch gleicht das System ebenso stark dem VIST Trainingssystem. Unterschiede fallen vor allem in der Oberfläche und der stärkeren Synchronisierung zur EKG Simulation auf.

Wie auch Immersion und Vist bietet Symbionix neben dem vaskulären Simulator, noch weitere Simulatoren (Endourologie, Endoskopie, Chirurgie) an. Untersuchungen über den klinischen Nutzen des Simulators zeigen deutlich positive Tendenz auf [67][66]. Eine Simulation von Eingriffen in den Kavernen ist derzeit nicht möglich.

Medical Simulation Corporation - SimSuite

Das von der Firma Medical Simulation Corporation entwickelte und vertriebene Gerät SimSuite (siehe Abbildung 4.7) [12] unterscheidet sich in Philosophie, Anwendung und Ausführung deutlich von den bisher vorgestellten Konkurrenzprodukten.

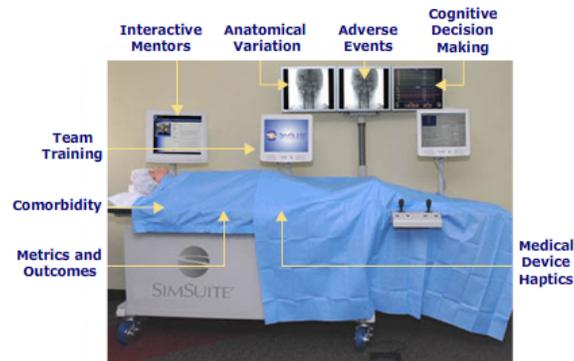


Abbildung 4.7: SimSuite Simulator der Firma Medical Simulation Corporation

Aussagen eines MSC³ Mitarbeiters zufolge, versteht sich SimSuite eher als Schulungssystem mit einem stärkeren Fokus auf der didaktischen als auf der Simulationsebene. So wird zwar (Stand Juni 2007 - eigene Beobachtung) die Bewegung der Instrumente simuliert, verläuft aber innerhalb eines Bereich vorgegebener Bahnen mit nur wenigen Variationsmöglichkeiten. Auch der Verlauf einer Intervention folgt stets vorgegebenen, didaktisch sinnvollen Wegen. Komplikationen treten stets an identischen Stellen auf und müssen für einen korrekten Ablauf der Simulationsprozedur auf vorgegebene Weise behandelt werden.

Ein ähnliches Konzept wird bei der graphischen Umsetzung verfolgt. SimSuite basiert hier auf aufgenommenen Realfilmen. An festgelegten Punkten werden Videos zu Ultraschallaufnahmen eingeblendet, die Hintergrundgraphik besteht aus aufgenommenen, originalen Röntgendaten. Instrumente werden in einer Direct3D Umgebung

³Medical Simulation Corporation

überlagernd dargestellt. Kontrastmittelgabe ist nur an bestimmten Punkten möglich und wird dann ebenfalls als aufgenommener Film gezeigt. Aufgrund der fest definierten Hintergründe können nur einige wenige Röntgenpositionen dargestellt werden. Die in der Kinderkardiologie gewünschte und benötigte Freiheit ist mit dem System nicht realisierbar.

Simsuite ist derzeit die einzige Simulationsumgebung auf dem Markt, die das Training von ASD Eingriffen erlaubt [3]. Auch hier wird allerdings das oben aufgezeigte didaktische Konzept verfolgt - eine realistische Simulation des Eingriffs mit allen zugrundeliegenden Freiheitsgraden ist daher nicht möglich.

Der Nutzen des Trainings an Simsuite konnte in einer Reihe klinischer Studien als positiv gezeigt werden [62][157]. Tieferegehende technische Information über das System sind leider nicht verfügbar.

The Simulation Group - EVE

EVE (Endovascular Training System) der Simulation Group [11] ist ein Simulator, der sich derzeit noch in Entwicklung befindet (siehe Abbildung 4.8). Entsprechend rar sind bisherige Informationen. Das Projekt konzentriert sich derzeit hauptsächlich auf neurologische Interventionen, kann aber theoretisch auf die Anwendung im kardiologischen Sektor erweitert werden.



Abbildung 4.8: Screenshot aus dem EVE System der Fa. The Simulation Group

Der Algorithmus zur Instrumentensimulation baut auf der Arbeit von Cotin et al. [68] auf. Der Algorithmus basiert grundlegend auf einer Finite Elemente Methode (siehe Abschnitt 4.2.4) und erlaubt eine schnelle Berechnung der Instrumentendeformation. Ein 2007 auf [11] veröffentlichtes Video zeigt aber diverse Schwächen im Realitätsgrad der Bewegung auf. Zudem kann nicht beurteilt werden, ob die Geschwindigkeit auf aktuellen Rechnern zur nutzergesteuerten Interaktivität ausreicht.

Kontrastmittelfluss wird anhand einer Netzwerkmethod berechnet und augenscheinlich über Veränderung der Oberflächendichten dargestellt. Dennoch werden Durchleuchtungslängen berücksichtigt und - soweit das anhand der Bilder beurteilbar ist - korrekt simuliert.

Bisher liegen noch zu wenige Informationen - und vor allem keine Erfahrungen - vor, um das System näher beurteilen zu können.

4.2 Instrumentensimulation

Einer der Kernpunkte in der Simulation katheterbasierter Eingriffe, ist die schnelle und akkurate Simulation des Instrumentenverhaltens. Durch Wandkontakte und innerer Rückstellkraft des Instrumentenmaterials induzierte Kräfte, führen zu Deformationen der Instrumente und der Umgebung.

Eine Simulation muss im Kontext dieser Arbeit den folgenden Ansprüchen genügen:

1. **Echtzeitfähigkeit:** Instrumente müssen innerhalb interaktiver Zeiten ein visuelles Feedback bieten. Nutzer müssen das Gefühl einer direkten 1:1 Umsetzung ihrer Eingaben erhalten.
2. **Realitätsgrad:** Instrumente müssen sich ausreichend realistisch verhalten. Da eine vollständig objektive Beurteilung über das realistische Verhalten der verschiedenen Verfahren oft nicht möglich ist, gilt als Maßstab, dass geschulte Nutzer (Ärzte mit ausreichender Erfahrung) das Verhalten als realistisch empfinden.
3. **Flexibilität:** Instrumente müssen flexibel gestaltbar, das Simulationssystem möglichst frei in der Definition der Instrumentenparameter sein. In der Kinderkardiologie werden eine Reihe von Instrumenten (bsp. Coil) verwendet, die über besondere Deformationseigenschaften (Spirale) verfügen - das Modell darf in Bezug auf die Form der Instrumente keine signifikanten Einschränkungen machen.
4. **Bedingungen an die Simulationsumgebung:** Allen nachfolgend gezeigten Verfahren ist gemein, dass sie von den jeweiligen Autoren nur im Zusammenspiel mit Gefäßeingriffen getestet und optimiert wurden und daher oft direkt auf die Modellierung der äußeren Strukturen als Röhrenmodelle aufbauen. Für die Simulation in der Kinderkardiologie ist aber eine nicht näher zu spezifizierende freie Topologie (Herzkammern) gegeben, die nicht mehr auf einfache Strukturen abzubilden ist.
5. **Zusammenspiel mehrerer Instrumente:** Während der Navigation werden in der Kinderkardiologie häufig mehrere, ineinander verschachtelte Instrumente verwendet die ineinander rotiert werden und sich gegenseitig beeinflussen. Der Algorithmus muss in der Lage sein, ein solches Verhalten effizient und realistisch abzubilden.

Im Folgenden werden verschiedene publizierte Verfahren ausführlich vorgestellt und auf ihre Nutzbarkeit im vorliegenden Projekt hin untersucht.

Die generelle Problemstellung ist formal durch die mathematische Repräsentation des Instruments sowie der umgebenden Strukturen und Lösung des daraus resultierenden Modells definiert. In allgemeiner Form lässt sich das Problem als dynamisches System in Form einer Differentialgleichung 2. Ordnung beschreiben:

$$\frac{d^2x}{dt^2} = H\left(t, x, \frac{dx}{dt}, Y, k\right) \quad (4.1)$$

wobei t die zeitliche Abhängigkeit, Y die Beeinflussung durch externe Parameter, k einen der Reibung entsprechenden Koeffizienten und x den eigentlichen, von der gewählten Repräsentierung abhängigen Parametervektor bezeichnet.

H beschreibt eine nicht lineare und problemabhängig zu spezifizierende Funktion (im einfachsten Falle entspräche $H = F/m = a$) die der Form einer Differentialgleichung zweiter Ordnung zur Beschreibung der Deformation unter dem Wirken externer Kräfte (Reibung, Trägheit, vorschiebende Kraft) genügt.

Unter Vernachlässigung dissipativer Kräfte (bsp. Reibung) entspricht dieses Modell der Formulierung einer Energieminimierung nach dem Hamiltonformalismus (siehe 3.5.1) auf dem gegenwärtige Instrumentensimulationen in verschiedenen Variationen aufbauen.

Der Argumentation aus dem Grundlagenkapitel zum Hamiltonschen Prinzip (3.5.1) folgend, kann Gleichung 3.19 als Basis verwendet und nach der Beschleunigung aufgelöst werden.

Da die Masse der simulierten Instrumente sehr klein ist, werden die Trägheitskräfte (bezogen auf typische Beschleunigungskräfte während einer Intervention) vernachlässigbar gegenüber den elastomechanischen Kräften der Instrumente und der Gefäßwände.

Analog kann für die Reibung eines Instruments argumentiert werden (Instrumente sind in der Regel mit einer extrem glatten, reibungsarmen Oberfläche versehen), weshalb auch diese in der Simulation in der Regel ignoriert werden kann.

Als Konsequenz dieser beiden Vereinfachungen verschwinden jedoch die zeitabhängigen Terme. Als physikalische Konsequenz tritt die Relaxation instantan ein und das System befindet sich zu jedem Zeitpunkt in einem statischen Gleichgewicht. Dadurch ergeben sich Folgen für die Wahl der Schrittweite des gewählten Lösungsverfahrens, die gewährleisten muss, dass sich die diskreten Lösungen mit den Lösungen einer kontinuierlichen Bewegung in Einklang befinden.

Ein typischer Weg dies zu erreichen ist es, zu gewährleisten dass sich die Lösung stets im Attraktionsgebiet des gleichen lokalen Optimums befindet. Dafür stellt die detaillierte Analyse der Größe der Attraktionsgebiete eine für die Wahl der Schrittweite wesentliche Information dar (siehe dazu auch Kapitel 5.1.8). Weitere Methoden, die Optimierung robuster werden zu lassen, wie etwa eine Anpassung des Funktionsraums, können genutzt werden, um eine größere Schrittweite zu ermöglichen.

Entsprechend dieser Diskussion können die verschiedenen Ansätze aus der Literatur eingeordnet werden. Die Simulation von Instrumenten basiert dabei hauptsächlich auf drei verschiedenen Ansätzen (siehe auch Kapitel 3.5.3). Direkte Optimierungsmethoden auf den durch die wirkenden Kräfte resultierenden Energien (die unter Vernachlässigung der zeitlichen Komponente in der Regel quasi-statisch berechnet werden), der Anwendung klassischer FEM⁴ Algorithmen und diversen Variationen von Feder Masse Modellen. Alle drei Entwicklungen sollen im Folgenden vorgestellt werden.

4.2.1 Quasi-Statische Mechanik

Auf die Simulation von Instrumenten angewandt, beschreibt der Begriff die Gruppe der Verfahren, die versuchen, das oben genannte Modell über eine Folge statischer

⁴Finite Elemente Methode

Zustände abzubilden und dabei zeitliche Entwicklungen, wie sie über Trägheit und Reibung entstünden, vernachlässigen. Derzeit unterliegen mit Ausnahme der Feder Masse Modelle alle Instrumentensimulationen den Gesetzmäßigkeiten einer Quasi-Statistischen Betrachtungsweise.

Lösungen des Modells können über die Auswertung semi-analytischer Gleichungen [30] oder numerischer Verfahren [68] erfolgen, in denen in der Regel die Berechnung eines Gleichgewichtszustand zwischen dem Zusammenspiel äußerer (deformierbare Umgebung) und innerer (deformierbares Instrument) Kräfte geschieht.

4.2.2 Direkte, semi-analytische Optimierung

Ein erstes auf die Instrumentensimulation spezialisiertes Modell wurde 2002 von Alderliesten et al. [30] vorgestellt. Instrumente werden in n durch $n - 1$ Gelenke miteinander verbundene Segmente fixer Länge diskretisiert und die durch interne und externe Kräfte induzierten Energien durch Lösung einer analytischen Gleichung [117] minimiert, um die Kräfte in ein statisches Gleichgewicht zu bringen. Dabei ist die interne Biegeenergie der Gelenke durch

$$U_b(\bar{x}_i) = \frac{1}{2} c_i \theta_i^2 \quad (4.2)$$

gegeben. θ_i beschreibt dabei den Winkel zwischen zwei Segmenten, der am Gelenk i auftritt. Die Konstante c_i bestimmt die Steifheit im Gelenk i . Entsprechend wird die Gesamtbiegeenergie eines Instruments bei Alderliesten als harmonische Approximation durch die Gleichung

$$U_{Tb} = \sum_{i=0}^{n-1} U_b(\bar{x}_i) \quad (4.3)$$

bestimmt.

Dem gegenüber steht die durch den Kontakt mit der Gefäßwand induzierte externe Energie eines Gelenks

$$U_w(\bar{x}_i) = \frac{1}{2} k d^2 \quad (4.4)$$

bei der Hookes Gesetz als Approximation der durch Deformation der Gefäße entstehenden Energie genutzt wird. Die Gesamtenergie berechnet sich abermals als harmonische Approximation durch Summation über die Gelenke.

Geometrisch werden die Instrumente bei Alderliesten in einer Notation repräsentiert, die Veränderungen abhängig von vorhergehenden Gelenken als relative Translationen α_i beschreibt und so Abhängigkeiten zwischen der Position eines Gelenks und der Konfiguration seiner Vorgänger ($j < i$) herstellt. Anschließend wird die Beschreibung der Instrumente auf eine 2D Darstellung im 3D Raum reduziert, die als freie Variablen nur noch die Länge der Translation (a_i) und einen Rotationswinkel (ψ_i) enthält.

$$\alpha_i = -\frac{\alpha_i^2}{2\lambda} \hat{\mathbf{e}}_i^{(P)} + \frac{\alpha_i}{2\lambda} \sqrt{4\lambda^2 - a_i^2} \cos \psi_i \hat{\mathbf{e}}_i^{(Q)} + \frac{\alpha_i}{2\lambda} \sqrt{4\lambda^2 - a_i^2} \sin \psi_i \hat{\mathbf{e}}_i^{(S)} \quad (4.5)$$

Das zu lösende System beruht bei Alderliesten (siehe [116]) auf den beiden fundamentalen Gleichungen:

$$\forall i : \frac{\partial U_{tot}}{\partial a_i} = 0 \quad (4.6)$$

und

$$\forall i : \frac{\partial U_{tot}}{\partial \psi_i} = 0 \quad (4.7)$$

wobei a_i der Länge einer Translation des Gelenks i und ψ_i einem Torsionswinkels (Rotation um Gelenksachse) im Gelenk i entspricht.

Gelöst wird das System durch eine aufwendig hergeleitete, semi-analytische Gleichung, die bei geringen Schrittlängen a_i und ψ_i und somit nach 4.5 α_i für jedes Gelenk direkt berechnet:

$$a_i = \frac{|\tilde{G} - c_i \tilde{\Upsilon}| \lambda}{2G^P - c_i \partial} \quad (4.8)$$

$$\psi_i = \beta_i + \pi \quad (4.9)$$

Auf eine Replizierung der Herleitung sei mit einem Verweis auf die ausführliche Dokumentation in [116] an dieser Stelle verzichtet. Dem Verständnis halber sei jedoch erwähnt, dass der Term die oben diskutierte Aufrechterhaltung des Gleichgewichts zwischen den internen und externen Energien eines Gelenks herbeiführt.

\tilde{G} bezeichnet darin die transversale Komponente der Summe der Gradienten über der externen Energie aller Gelenke mit einem Index von $i \geq i + 1$. G^P drückt den longitudinalen Anteil aus. Die longitudinale Komponente der Biegeenergie in Gelenk i wird durch den Term $c_i \partial$ repräsentiert, während $c_i \tilde{\Upsilon}$ wiederum für den transversalen Anteil steht.

β_i ist der in Polar Koordinaten ausgedrückte Winkel, der durch den Vektor $\tilde{G} - c_i \tilde{\Upsilon}$ repräsentiert wird und den Ausgleich in der transversalen Ebene zwischen den durch Kontakt mit Gefäßwänden entstehenden Gradienten und der mit dem Gelenk i verbundenen Biegeenergie herstellt.

Ein Zustand minimaler Energie wird erreicht, wenn ψ_i gleich dem Wert von $\beta_i + \pi$ ist. Es ist leicht zu sehen, dass ein Gleichgewicht zwischen lokaler Biegeenergie und durch Kontakt induzierten Energien ($\tilde{G} = c_i \tilde{\Upsilon}$) dazu führt, dass a_i gleich 0 wird und das $i - te$ Gelenk sich somit in einem stabilen Zustand befindet.

Die implizite Berechnung von Torsionseffekten⁵ ist somit prinzipiell möglich, wird aber in [30] bewusst vernachlässigt (die Torsionskonstante für Führungsdrähte wird als unendlich angenommen und aus der Berechnung herausgenommen).

Dem Algorithmus liegt ein voll physikalisches Modell zugrunde, das technisch gesehen nur auf dissipative Kräfte verzichtet und somit hohe Genauigkeiten in der Simulation aufweisen sollte. Weiterhin ist durch die Fixierung von Segmentlängen keine Simulation von Streckungen oder Dehnungen der Gelenk möglich, was aber in der Regel bei Instrumenten in der Kardiologie auch keine größere Rolle spielt.

⁵ Axiale Verdrehung eines Katheters, ein Effekt, der in der Kardiologie häufiger vorkommt

Eine Erweiterung des Algorithmus zur Behandlung von dissipativer Reibung, die 2007 veröffentlicht wurde [31] benötigt, abhängig von der Länge der Iterationsschritte (Vorschiebegeschwindigkeit des Instruments) und Länge der Gelenke Zeiten von Minuten bis zu mehreren Stunden um das Instrument um eine Distanz von 175 mm zu bewegen (ohne Rotation).

Die Zeitmessungen sind leider auch nicht konkret genug nach Iterationen aufgeschlüsselt, so dass aus den Messungen nicht klar hervorgeht, ob die Zeiten aufgrund einer einmaligen Berechnung oder aufgrund der Summation vieler einzelner Zeitschritte entstanden sind.

Die Berechnung der durch Reibung induzierten Kräfte selbst, bewegt sich noch auf sehr theoretischem Niveau. Da keine Untersuchungen zum tatsächlichen Effekt der Reibung bei Katheterinterventionen vorliegen, basieren die in [32] verwendeten Konstanten auf nicht belegbaren Werten und stellen derzeit nicht mehr als einen grundlegenden Beweis des Konzeptes dar.

Der Algorithmus wurde, mehr als alle anderen Verfahren, experimentell auf seine Genauigkeit untersucht [116][32][31] und kann dort auch sehr gute Ergebnisse aufweisen.

Dennoch ist das Verfahren gegenwärtig nicht für eine Anwendung in der Echtzeitsimulation geeignet. Es krankt trotz diverser Versuche es zu beschleunigen [45] an der extrem hohen Berechnungszeit (mehrere Sekunden pro Berechnungsschritt) und der geringen möglichen Translationslänge pro Iterationsschritt (< 0.3 mm). Gleichung 4.8 ist bei Translation und Rotation des Instruments mehrfach interaktiv auf allen Gelenken zu berechnen, obliegt einer starken Beschränkung der Vorschiebeschrittweite (andernfalls kann aufgrund während der Herleitung angenommener Approximationen keine sichere und genaue Konvergenz erzielt werden) und wird intern durch aufwendig zu berechnende Gradienten dominiert.

Mit dem derzeitig verwendeten geometrischen Modell wäre auch keine Beschreibung spiralenbildender Instrumente (wie etwa Coils) möglich. Eine Anpassung wäre allerdings denkbar. über die Wahl der verwendeten Kollisionsdetektion, und ob damit weitere Einschränkungen einhergehen, wird bei Alderliesten et al. keine nähere Aussage getroffen.

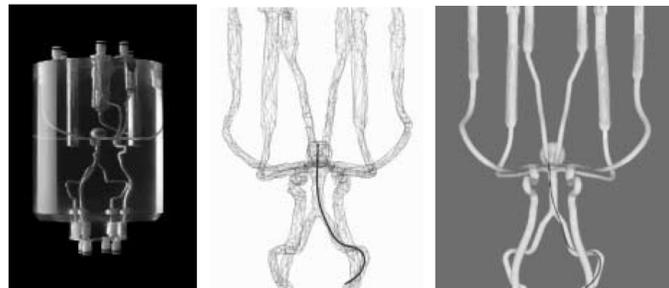


Abbildung 4.9: Führungsdrahtsimulation von Alderliesten et al. [30].

Möglichkeiten der Beschleunigung werden in [31] untersucht - es werden verschiedene numerische Optimierungsverfahren wie konjugierte Gradienten und evolutionäre Algorithmen mit dem semi-analytischen Verfahren aus [116] verglichen - dennoch

wird durch die resultierenden Zeiten die gewünschte Echtzeitfähigkeit noch nicht erreicht. Ferner verliert der Algorithmus unter Verwendung der numerischen Optimierung viel von seiner proklamierten Genauigkeit. Wünschenswert wäre eine Kombination der unter Verwendung des semi-analytischen Verfahrens erzielten Genauigkeit mit einer echtzeitfähigen Geschwindigkeit.

Die Publikationen nennen keine direkte Möglichkeit, verschachtelte Instrumente zu simulieren. Generell wären jedoch, aufgrund der verwandten geometrischen Repräsentation der Instrumente, ähnliche Ansätze wie die in dieser Arbeit neu vorgestellten Verfahren (siehe Kapitel 5.1) denkbar.

4.2.3 Feder Masse Modelle

Feder Masse Modelle werden in physikalischen Berechnungen hauptsächlich für die Simulation von Fäden, Seilen und Stoffen verwendet. 2005 zeigten Taskiran et al. in [194] ein reines Feder Masse Modell für die Simulation von Haar. Um die axiale Steifigkeit der Haarstränge zu erhöhen, werden Federn über mehrere Massepunkte hinweg verteilt. Das Modell ist zwar prinzipiell schnell (mehrere Haarstränge können parallel auf der aktuellen Rechnergeneration simuliert werden), kann aber nicht die grundlegend für Katheter und Drähte benötigten nicht linearen Steifigkeiten abbilden. Auch sind externe Kräfte und Kollisionserkennungen sehr einfach gehalten.

Aufbauend auf ein 2002 veröffentlichtes Modell von Kukuk und Geiger [122] haben Guilloux et al. 2005 in [92] ein verwandtes Modell (Home Spring Modell [129]) für die Verwendung in der Simulation von Führungsdrähten vorgestellt. Fixe Segmente werden über Rotationsfederkräfte an Massepunkten (Gelenken) miteinander verknüpft.

Unter der Einwirkung externer Kräfte erhöht sich die Belastung auf den Federn, die dann entsprechend ihrer Federstärke (Instrumentensteifigkeit) nachgeben und ein steifes Instrument abbilden können. In [93] wird ein streng heuristisches Verfahren verwendet, dass strikt Gelenke bei Wandkontakt bis zum Erreichen eines Schwellwertwinkels abknicken lässt, und sich anschließend iterativ auf distale Elemente fortsetzt. Anschließend wird eine numerische Relaxation durchgeführt.

Das System kann mit den bekannten Verfahren zur numerischen Integration gelöst werden. In [93] wird allerdings nur der Einsatz eines expliziten Euler-Verfahrens untersucht.

Das Verfahren von Guilloux sollte ursprünglich auch zur Pfadplanung bei Führungsdrähten eingesetzt werden - bisher sind aber keine weiteren diesbezüglichen Veröffentlichungen erschienen.

Die aktuellste Arbeit zum Einsatz von Feder Masse Modellen in der Instrumentensimulation wurde 2007 von Wang et al. vorgestellt [204] und fügt weitere Kräfte zum Ansatz von Guilloux hinzu. Geometrisch werden Instrumente durch eine Menge von K miteinander verknüpften Knoten repräsentiert, die durch ihre Position $\mathbf{x}_i = (x_i^1, x_i^2, x_i^3)$, einen Torsionswinkel q_i und die Verknüpfung mit ihren Nachbarknoten definiert werden. Ferner ist jedem Knoten eine Masse m_i und ein polares Trägheitsmoment I_i zugeordnet. Mehrere Federn zwischen den Knoten produzieren Kräfte um Streckung, Kompression, Biegung und Torsion zu simulieren. Die Dynamik jedes Knoten wird bestimmt durch

$$\tau_i = I_i \cdot \ddot{q}_i(t) \quad (4.10)$$

$$\mathbf{F}_i = m_i \cdot \ddot{\mathbf{q}}_i(t) \quad (4.11)$$

, was formal einer der Beschreibung nach der Euler Lagrange Differentialgleichung unter Vernachlässigung der Reibung entspricht.

Die Torsionskraft τ_i entlang des Instruments wird durch eine Summe aus Streckung und Reibung berechnet ($\tau_i = \tau_i^s + \tau_i^r$), während die Kraft \mathbf{F}_i als Summe aller simulierten Kräfte berechnet wird:

$$\mathbf{F}_i = \mathbf{F}_i^s + \mathbf{F}_i^b + \mathbf{F}_i^t + \mathbf{F}_i^r + \mathbf{F}_i^c + m_i \mathbf{g} \quad (4.12)$$

wobei \mathbf{F}_i^s die aus den streckenden Federn, die mit Knoten i verbunden sind, resultierende Kraft darstellt, \mathbf{F}_i^b die Biegekraft, die durch die Rotations- (angulären) Federn entsteht, \mathbf{F}_i^t , die Kraft, die durch Torsionsfederkräfte hervorgerufen wird. Dissipative Kräfte werden durch die Reibungskraft \mathbf{F}_i^r simuliert. Der Kontakt mit Gefäßwänden induziert eine weitere Kraft \mathbf{F}_i^c während $m_i \mathbf{g}$ die wirkende Gravitationskraft darstellt.

\mathbf{F}_i^c wird bei Wang wie folgt festgelegt:

$$\mathbf{F}_i^c = \kappa^c e^{\kappa^0 d} \cdot \mathbf{o}_i \quad (4.13)$$

\mathbf{o}_i ist dabei ein Richtungsvektor, der in Richtung der resultierenden Kraft zeigt und die Konstanten κ^c und κ^0 stellen heuristisch ermittelte Konstanten zur Justierung der Kraft dar. Die e Funktion stellt einen klassischen Strafterm in Form einer exponentiellen "Penalty Funktion" dar.

Werden Knoten durch externe Einwirkung (Rotation, Translation durch den Nutzer) bewegt, so wird die Position dieser direkt bewegten Knoten (in der Regel der proximale Knoten) als fix angesehen und anschließend die resultierenden Kräfte nach 4.12 berechnet. Die neuen Positionen der Knoten werden anschließend über einfache Euler Integration bestimmt, was bei derart steifen Systemen in der Regel zu instabilen Lösungen führen kann (siehe unten).

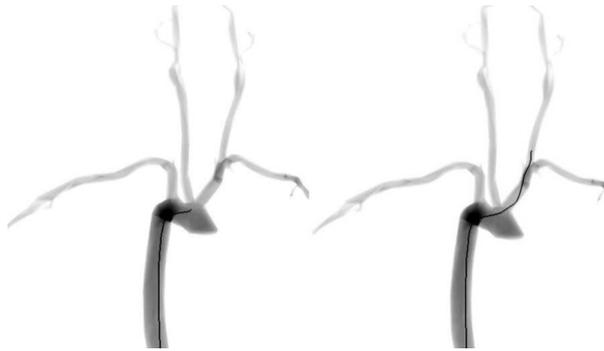


Abbildung 4.10: Führungsdrahtsimulation von Wang et al. [204].

Wang et al. zeigen in ihren Publikationen eine sehr performante (Updaterate 500Hz bei 120 Knoten) Methode um Katheter und Führungsdrähte zu simulieren, was neben den verhältnismäßig einfachen Gleichungen auch in der Verwendung einer effizienten Kollisionsdetektion begründet ist (siehe dazu auch Abschnitt 4.3).

Die Autoren der Publikation machen keine Aussage über die verwendete Schrittweite. Allgemein ist bei Feder Masse Systemen mit hohen inhärenten Steifigkeiten aber von einer sehr kleinen Schrittweite ($\ll 1mm$) auszugehen [84] was die Geschwindigkeit des Gesamtsystems wieder relativiert.

In der Publikation werden keine objektiven Angaben über die Genauigkeit des Verfahrens gemacht. Allerdings wird im Fazit das Verhalten der simulierten Instrumente als eine der Limitationen des Verfahrens genannt.

Ein möglicher Grund dafür findet sich bei der Betrachtung der speziellen Eigenschaften von Feder Masse Systemen, die hinreichend untersucht wurden [84]. Feder Masse Systeme bilden grundsätzlich nur eine grobe Approximation der physikalischen Gegebenheiten ab [193]. Speziell inkompressible Materialien mit schlechten Dehnungs-/Streckungseigenschaften sind aufgrund der nicht fixen Verbindung nur schwer realistisch zu modellieren.

Ein weiteres Problem stellen die umgebenden Strukturen dar. Feder Masse Modelle werden unter schlechten Randbedingungen schnell numerisch instabil. Derartige Randbedingungen sind etwa eine schwer durchdringbare Umgebung oder hohe Eigensteifigkeit wie es bei der Instrumentensimulation häufig vorkommt [84]. Erschwerend kommt hier der physikalisch nicht motivierte Einsatz des exponentiellen Strafterms hinzu, der schon bei kleinen Eindringtiefen sehr hohe Rückstellkräfte erwirkt und somit eben diese schlechten Randbedingungen herbeiführt.

Trotz seiner augenscheinlich hohen Geschwindigkeit ist der Algorithmus daher nicht für die Zwecke anspruchsvoller Echtzeitsimulation von Instrumenten geeignet. Im Vergleich mit der Qualität anderen Methoden stellen Feder Masse Modelle in der gegenwärtigen Form keine wirklich brauchbare Alternative für den angestrebten Verwendungszweck dar.

Es existiert in der Literatur derzeit keine Publikation über die Simulation verschachtelter Instrumente mittels Feder-Masse Methode, was einen weiteren Hinderungsgrund für die Verwendung in der vorgestellten Simulation bildet.

4.2.4 Finite Elemente Methode

Die FEM wurde ursprünglich zu Beginn der 60er Jahre in der Luft- und Raumfahrtindustrie, kurze Zeit später auch im Fahrzeugbau eingesetzt [53]. Die Idee hinter FEM besteht darin, das zu berechnende Gebiet in eine endliche Zahl kleiner, über Randbedingungen und partielle Differentialgleichungen miteinander verknüpfte Elemente zu zerteilen. Aus diesen ergibt sich ein großes Gleichungssystem über dessen Lösung man auf die Resultate für die Einzelelemente rückschließen kann.

Erstmals zur Simulation kardiologischer Instrumenten eingesetzt wurde die FEM 1999 unter Lawton et al. [127]. Eine Fortführung und detailliertere Version des Verfahrens findet sich unter [128], wo Rotationsgruppen und Cosserat Modelle in der Berechnung der FEM-Gleichungen verwendet werden. Das Modell ist allerdings nicht in der Lage, Torsionen, die innerhalb des Katheters auftreten, zu simulieren. Über die Genauigkeit sind keine qualitativen Untersuchungen veröffentlicht worden.

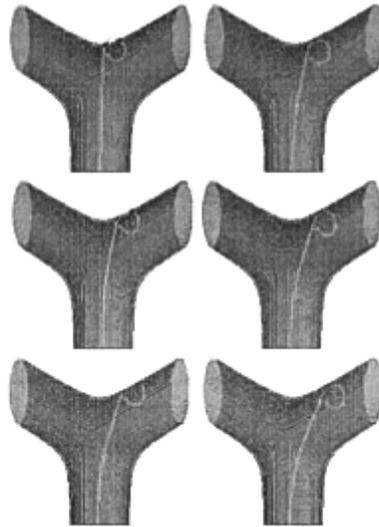


Abbildung 4.11: Kathetersimulation von Lawton et al. [128].

Weitere Anwendung findet die FEM in [65], wo Pai et al. ebenfalls Cosserat Modelle nutzen, um dünne, steife Materialien (im Kontext chirurgische Fäden) zu simulieren.

Ein weiteres virtuelles Katheter Modell, das auf einer lineare elastischen FEM Repräsentation basiert, wurde unter Verwendung von finiten Balkenelementen 2001 von Nowinski und Chui [52] vorgestellt. Balkenelemente bieten den Vorteil, dass sich Trägheitsmoment und polarer Trägheitsmoment querschnittsbasiert modellieren lassen und somit hohle Instrumente verschiedener Querschnitte und mechanischer Eigenschaften abgebildet werden können. Das Hauptproblem dieses linearen Modells besteht darin, dass sich die starken Nicht-Linearitäten, die während der Navigation eines Instruments in komplexen Umgebungen auftreten, damit nicht ausreichend abbilden lassen.

Eine Weiterentwicklung des Modells von Lawton wurde 2005 von Duriez et al. [68] gezeigt. Das Modell verwendet im Gegensatz zu Lawtons Algorithmus keine Cosserat Modelle, sondern erstellt die für die FEM benötigten Strukturmatrizen auf Grundlage klassischer Balkenmodelle. Nicht-Linearitäten werden durch inkrementelle Lösungen eines linearen Modells zu Echtzeitbedingungen gelöst.

Jedes Balkenelement wird in dem Ansatz von Duriez entsprechend der bekannten FEM Balkentheorie durch eine 12×12 symmetrische Matrix K_e repräsentiert (die genaue Herleitung findet sich in [68]).

Aus den Strukturmatrizen der einzelnen Gelenke wird durch Summation eine globale Steifheitsmatrix $[K]$ erstellt, die im quasi-statischen Fall ein Gleichgewicht unter der Bedingung

$$[K]U = F \tag{4.14}$$

herstellt. $[K]$ ist dabei, begründet dadurch, dass jeder Knoten maximal mit zwei Nachbarelementen geteilt wird, eine Bandmatrix. U repräsentiert eine Spaltenmatrix

$$K_e = \frac{E}{l} \left[\begin{array}{cccc|cccc} A & & & & \leftarrow [K_i]_{(1,1)} & & & [K_i]_{(1,2)} \\ 0 & \frac{12I_z}{l^2(1+\phi_y)} & & & & & & \\ 0 & 0 & \frac{12I_y}{l^2(1+\phi_z)} & & & & & \\ 0 & 0 & 0 & \frac{GJ}{E} & & & & \\ 0 & 0 & \frac{-6I_y}{l(1+\phi_z)} & 0 & \frac{(4+\phi_z)I_y}{1+\phi_z} & & & [K_i]_{(2,2)} \\ 0 & \frac{6I_z}{l(1+\phi_y)} & 0 & 0 & 0 & \frac{(4+\phi_y)I_z}{1+\phi_y} & & \downarrow \\ \hline -A & 0 & 0 & 0 & 0 & 0 & A & \\ 0 & \frac{-12I_z}{l^2(1+\phi_y)} & 0 & 0 & 0 & \frac{-6I_z}{l(1+\phi_y)} & 0 & \frac{12I_z}{l^2(1+\phi_y)} \\ 0 & 0 & \frac{-12I_y}{l^2(1+\phi_z)} & 0 & \frac{6I_y}{l(1+\phi_z)} & 0 & 0 & 0 & \frac{12I_y}{l^2(1+\phi_z)} \\ 0 & 0 & 0 & \frac{-GJ}{l} & 0 & 0 & 0 & 0 & \frac{GJ}{E} \\ 0 & 0 & \frac{6I_y}{l(1+\phi_z)} & 0 & \frac{(2-\phi_z)I_y}{1+\phi_z} & 0 & 0 & 0 & \frac{6I_y}{l(1+\phi_z)} & 0 & \frac{(4+\phi_z)I_y}{l(1+\phi_z)} \\ 0 & \frac{6I_z}{l(1+\phi_y)} & 0 & 0 & 0 & \frac{(2-\phi_y)I_z}{1+\phi_y} & 0 & \frac{-6I_z}{l(1+\phi_y)} & 0 & 0 & 0 & \frac{(4+\phi_y)I_z}{1+\phi_y} \end{array} \right]$$

mit räumlichen Verschiebungen, die mit den externen Kräften (F) korrespondiert.

Würde dieses Modell als vollständig angenommen, könnten damit nicht die für Instrumente typischen, nicht linearen geometrischen Eigenschaften simuliert werden. Daher wird in [68] vorgeschlagen, zu jeden Berechnungsschritt die Matrix $[K_e]$ unter Verwendung der letzten Lösung zu aktualisieren und neue lokale Steifheitsmatrizen zu generieren. Auf diese Weise erfolgt jeder Schritt linear, während für das globale Instrument ein nicht lineares Verhalten erzielt werden kann.

Um zu verhindern, dass Instrumente sich inelastisch verhalten (ohne Einwirkung äußerer Kräfte würden die Instrumente mit der beschriebenen Adaption des Algorithmus nur in den letzten stabilen - nicht jedoch in den ursprünglichen Zustand relaxieren), wird in das System vor der Lösung eine Dämpfungskraft ($F_t = -\alpha[K_t](X_t - X_0)$, $0 < \alpha \leq 1$) eingebracht, die mit dem Dämpfungskoeffizienten des Instrumentenmodells korreliert.

Die Formulierung basiert auf physikalischen Gegebenheiten und stellt, abgesehen von der Einbringung der (eher heuristischen) Dämpfungskraft, eine realistische Modellierung von Instrumenten dar.

In Bezug auf Geschwindigkeit, leiden FEM Modelle häufig unter der hohen Anzahl der freien Variablen in den entstehenden Gleichungssystemen (bei 100 Gelenken sind schon über 1000 freie Variablen im Gleichungssystem). Versucht man solche Gleichungssysteme über Invertierung und ohne iterative Verfahren zu lösen, können Echtzeitbedingungen auf heutigen Rechnern nicht mehr erreicht werden.

Kommen wie in [68] weitere Beschränkungen hinzu, ist eine Lösung, auch auf Grundlage iterativer Verfahren, nur noch schwer in Echtzeit zu erreichen.

Die Lösung von 4.14 erfolgt daher in [68] nicht direkt sondern unter Zuhilfenahme einer Substruktur Analyse. Das gesamte Instrument wird in mehrere Substrukturen unterteilt, wobei jede Substruktur ein oder mehrere Balkenelemente beinhaltet. Jede Substruktur wird dann - unter der Annahme, dass die Grenzelemente fixiert sind separat gelöst. In einem zweiten Schritt werden anschließend die Grenzelemente relaxiert.

Unter Verwendung dieser Techniken werden Geschwindigkeiten von 25 ms pro Frame bei einem aus 100 Gelenken bestehenden Instrument erreicht. Die relaxierte Länge der dabei verwendeten Gelenke wird in der Publikation leider nicht angegeben, was einen Vergleich mit den anderen Verfahren erschwert.

In [130] wird erstmals in der Literatur zur Instrumentensimulation ein Algorithmus vorgeschlagen um mehrere verschachtelte Instrumente effizient zu berechnen.

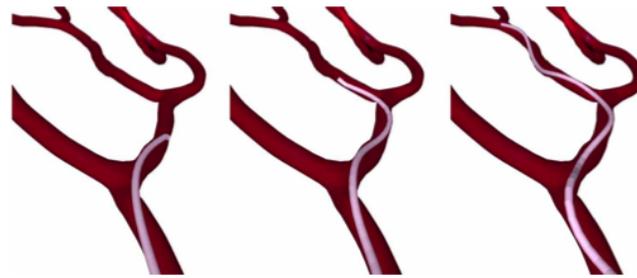


Abbildung 4.12: Kathetersimulation von Duriez et al. [68].

Die Biegesteifigkeit und resultierende Form wird mittels der empirischen Halpin-Tsai Gleichungen [95] berechnet, wobei allerdings davon ausgegangen wird, dass die Instrumente sich nur in der Steifigkeit und der Stärke der Vorbiegung voneinander unterscheiden.

Die Behandlung von Rotationen verschachtelter Instrumente wird nicht näher untersucht. Die in [130] dargestellten Bilder zeigen einen Draht, der einen Katheter vollständig streckt - eine echte Überlagerung (in Form einer partiellen Deformation) ist im gezeigten Fall nicht festzustellen.

Die Simulation von spiralförmigen Instrumenten (Coil) ist gegenwärtig mit dem System nicht möglich, wird aber nach [130] derzeit untersucht. Zum aktuellen Zeitpunkt ist keine Veröffentlichung über die Ergebnisse der Untersuchungen bekannt.

Duriez et al. verwenden in ihrem Verfahren ein System der Kollisionsdetektion, das auf einer Röhrenförmigen Beschreibung der Umgebung von Krissian et al. [121] aufbaut. Entsprechend ist das System ohne Eingriffe in das Kollisionsdetektionsverfahren nicht in Bereichen außerhalb röhrenförmiger Gefäße verwendbar.

2007 wurde durch Allard et al. das SOFA⁶-Framework zur medizinischen Simulation vorgestellt [34]. Neben einer Reihe weiterer Methoden zur physikalischen Berechnung, ist auch der FEM-Algorithmus von Duriez zu Teilen in dem Paket enthalten.

Die Neuerung besteht in der Verwendung der CUDA⁷-Architektur moderner Grafikkarten, die eine Berechnung der Gleichungen auf der GPU erlaubt und somit den Prozessor entlastet.

Derzeit gibt es allerdings keine Veröffentlichungen über die möglichen Leistungssteigerungen bei der Instrumentensimulation unter Verwendung der CUDA Architektur. Die frei verfügbaren Quellen sind auch nicht ausreichend, um die Simulation ohne Weiteres nachzubilden.

4.2.5 Andere Modelle

Neben den oben vorgestellten Modellen existieren noch weitere Algorithmen, die sich im weitesten Sinne mit der Simulation von Kathetern beschäftigen. Diese sind in der Regel aber nicht geeignet, um in einer Echtzeitsimulation verwendet zu werden. Viel-

⁶Simulation Open Framework Architecture

⁷Computer Unified Device Architecture

fach war dabei die Echtzeitfähigkeit kein primäres Ziel der Entwicklung. Ein Algorithmus, um einen kollisionsfreien optimalen Weg für die Pfadplanung von Katheterinterventionen in einer virtuellen Umgebung zu finden, wird beispielsweise in [81] von Gayle et al. vorgestellt. Das Verfahren lässt sich aber nicht zur inkrementellen Simulation von Instrumenten nutzen, das stets nur statisch der optimale Pfad berechnet wird, der vom Realpfad abweichen kann.

4.2.6 Fazit

	Direkte, (semi-analytische) Optimierung	Finite Elemente Methode	Feder Masse Methode
Echtzeitfähigkeit	○	●	●
Realitätsgrad ¹	●	★	○
Flexibilität ²	★	★	○
Simulationsumgebung ³	○	○	○
Verschachtelte Instrumente ⁴	○	●	○

¹ Die Angaben beziehen sich darauf, dass nur für das Verfahren von Alderliesten genaue Untersuchungen in Bezug auf die Nähe zur Realität vorliegen.

² Keines der Verfahren bietet derzeit die Möglichkeit, Coils zu simulieren. Die Formulierungen erlauben aber teilweise eine einfache Adaption.

³ Alle Verfahren vereinfachen die Kollisionsdetektion über die Annahme, in röhrenförmigen Strukturen zu operieren.

⁴ Das Verfahren von Duriez ist derzeit das einzige, das die Simulation verschachtelter Instrumente untersucht hat.

Tabelle 4.3: Vergleich der wichtigsten Verfahren in Bezug auf ihre Verwendbarkeit in der kinderkardiologischen Simulation. - ○: Bedingung nicht ohne weitergehenden Forschungsaufwand erfüllbar. ●: Bedingung erfüllt oder mit nur geringem Aufwand erfüllbar. ★: Bedingung mit moderaten Anpassungen des Systems theoretisch erfüllbar.

Es zeigt sich nach Prüfung des aktuellen Stands der Technik, dass keines der bekannten Verfahren ohne Weiteres für die Kinderkardiologie übernommen werden kann. Als Fazit steht in Kapitel 5.1 die Entwicklung eines neuen Verfahrens, das die besten Eigenschaften der einzelnen Methoden in Bezug auf die Simulation in der Kinderkardiologie vereint.

4.3 Kollisionsdetektion

4.3.1 Allgemeine Verfahren

Sowohl bei der Berechnung der Instrumentensimulation als auch bei der Simulation des Kontrastmittelflusses (siehe Abschnitt 4.5) ist die Kollisionsdetektion ein entscheidender und - in Bezug auf Performanz - dominierender Faktor. Die Ansprüche an eine Kollisionsdetektion sind eine hohe Geschwindigkeit, Robustheit und Flexibilität in Bezug auf die kollidierenden Primitive. Im Falle der Instrumentensimulation ist

zusätzlich die schnelle Bestimmung des Abstands zwischen einem Punkt (Instrumentenpunkt) und einem Objekt (umliegende Struktur) gefragt.

Naiv ausgeführt, bedeutete die Kollisionsberechnung auf N Objekten ein Problem der Komplexität $O(N^2)$. Daher unterteilt sich eine Kollisionsdetektion üblicherweise in zwei Phasen - der sogenannten breiten und der schmalen Phase⁸. Die breite Phase sortiert Objekte in räumlichen Hierarchien und bildet Gruppen von Objekten, die möglicherweise miteinander kollidieren könnten. In der schmalen Phase werden die verbleibenden Kollisionskandidaten auf mögliche Kollisionen hin überprüft [71].

Viele Verfahren zur 3D Kollisionsdetektion (bsp. Lin-Canny [132]) nutzen zu der räumlichen auch zeitliche (Objekte bewegen sich zwischen zwei Zeitabschnitten in der Regel nur über kurze Distanzen) Kohärenzen aus, um potentielle Kollisionspartner schnell zu detektieren und auszuwerten.

Die Verwendung von Proxys ist ein weit verbreitetes Standardverfahren für die Abschätzung in der breiten Phase. Objekte werden bei dieser Methode durch einfach zu berechnende, vollständig umschließende Primitive approximiert. Das bekannteste Beispiel dazu sind AABBs⁹. Diese, an den Hauptachsen des verwendeten Koordinatensystems ausgerichteten Begrenzungsquader, enthalten ein Objekt vollständig und können dazu genutzt werden auf einfache Weise eine erste Abschätzung für eine Kollision durchzuführen. Abbildung 4.13 zeigt eine solche Approximation. Nur Objekte, die in Boxen liegen, deren Projektionen sich auf *jeder* der Hauptachsen überlagern, müssen in der schmalen Phase auf Kollision geprüft werden. Ansonsten entspricht der Test den Anforderungen des Separating Axis Theorem [29] und eine Kollision kann ausgeschlossen werden.

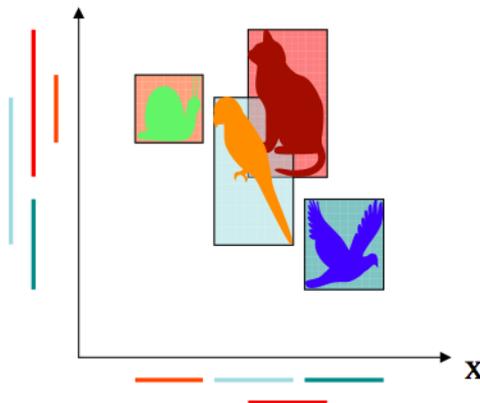


Abbildung 4.13: ABBs im exemplarischen Beispiel.

Neben AABBs sind auch OBB¹⁰, Kugeln oder minimale Ellipsoide [72] häufig genutzte Proxies, die sich vor allem durch den Aufwand während ihrer Erstellung und die Geschwindigkeit während der Echtzeitabfrage unterscheiden. Außerdem muss beachtet werden, dass einige Proxies (z. bsp. Kugeln) für rigide Objekte besser geeignet

⁸Broad Phase & Narrow Phase

⁹Axis Aligned Bounding Boxes

¹⁰Oriented Bounding Boxes

sein können, weil sie wünschenswerte Eigenschaften wie etwa Invarianz bei Rotationen mit sich bringen und nicht ständig neu berechnet werden müssen.

Eine weitere bekannte Methode für die Behandlung der breiten Phase sind BVHs¹¹ wie Octrees. Objekte werden in eine dreidimensionale geometrische Baumstruktur einsortiert, indem ein kubisches Volumen, das alle Objekte der Szene enthält rekursiv immer weiter in acht Oktanten unterteilt wird, bis eine vorgegebene Rekursionstiefe erreicht ist oder die Anzahl der im Knoten enthaltenen Primitive unter einen festgesetzten Schwellwert gefallen ist [175]. In der schmalen Phase müssen dann nur noch die Primitiven im Oktanten des zu prüfenden Objektes getestet werden.

Der Nachteil des Verfahrens ist, dass die Quader des Octrees in der Regel das zu umschließende Objekt nicht gut approximieren. Primitive müssen entweder in mehrere benachbarte Quader einsortiert (und damit gedoppelt) oder aber unterteilt werden, was zu einem Anwachsen der Gesamtpolygonzahl führt.

Aus diesem Grund gibt es weitere, verwandte Verfahren die versuchen auf verschiedenen Wegen das Objekt besser zu approximieren oder aber die anfallenden Tests zu beschleunigen.

Sphere-Trees, bei dem Objekte in Hierarchien von Kugeln eingeteilt werden (siehe Abbildung 4.14 eignen sich vor allem für schnelle Kollisionsdetektion, da Distanztests und Überlappungstests auf Kugeln sehr performant ausgeführt werden können. Allerdings sind viele Typen von Objekten durch Kugelmengen nur sehr schlecht zu approximieren, was in der breiten Phase zu vielen falsch positiv getesteten Resultaten führen kann.

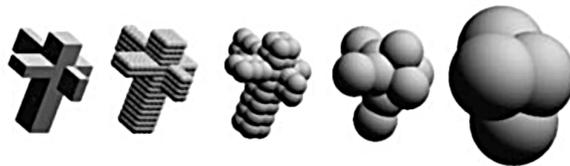


Abbildung 4.14: Sphäretrees im exemplarischen Beispiel (Bild aus [155])

Abhilfe kann eine optimierte Erstellung des Sphäretrees schaffen, wie sie etwa in [156] von Palmer et al. gezeigt wurde. Hier wird von Objekten erst eine Skelettierung erstellt. Werden die Kugeln für den Sphäretree auf dieser Skelettierung gewählt, erhält man eine Approximation, die nahe am möglichen Optimum liegt. Die Skelettierung selbst ist gegeben, durch die Zentren der maximalen Kugeln in einem Objekt X . Eine Kugel S gilt als maximal in X , falls S vollständig in X enthalten ist und keine andere Kugel in X existiert die S umschließt. Das Problem der optimierten Sphere Tree Erstellung ist allerdings recht zeitaufwendig. Sollen Objekte in Echtzeit angepasst werden, können allerdings neuere, empirische Algorithmen wie etwa das 2004 in einer Arbeit von Kumar et al. vorgestellte Verfahren [123] gute Ergebnisse liefern.

Swept Sphere Volumes haben sich in den letzten Jahren mit einem sehr guten Verhältnis zwischen Objekt Approximation und schnellen Kollisionstests hervorgetan [126]. Statt Kugeln oder Quadern werden Gruppen von Swept Sphere Volumes

¹¹Bounding Volume Hierachies

als Proxies verwendet. Diese Volumina entstehen aus der Minkowski Summe eindimensionaler Objekte (bsp. Punkt Linie, Rechteck) und eine Kugel. Die entstehenden Objekte zeigt Abbildung 4.15.

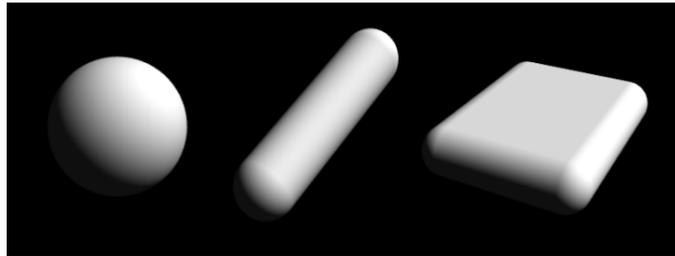


Abbildung 4.15: Swept Sphere Volumes aus Punkt, Linie und Rechteck im Vergleich

Eine weitere mögliche Hierarchie für die breite Phase ist der C-Tree bei dem Kugeln und konvexe Polyeder die Basis für die Hierarchie bilden [209]. Die Approximation an die Objekte ist bei diesem Verfahren deutlich besser, die Einteilung muss aber manuell erfolgen.

Es existieren eine Menge weiterer Verfahren für die Behandlung der breiten Phase wie etwa Shell-Trees [120], oder Zonotopen [91]. Diese haben aber aufgrund ihrer sehr speziellen Anforderungen keine große Verbreitung gefunden und sind für diese Arbeit auch nicht weiter relevant.

Für die schmale Phase sind im Rahmen dieser Arbeit besonders Algorithmen von Interesse, die eine zeitliche Kohärenz ausnutzen. Der Katheter bewegt sich zwischen zwei Zeitabschnitten in der Regel nur um wenige Millimeter - das Kollisionsgebiet kann also entsprechend gut abgeschätzt werden.

Bekannt ist hier vor allem der Lin-Canny Algorithmus [132]. Das Verfahren sieht vor, dass Features (Kanten, Vertices...) von polygonalen Objekten, die in einer vergangenen Suche die geringste Distanz hatten als Ausgangspunkt für die nächste Distanzabschätzung genutzt werden - wird ausgehend von einem Feature in einer Breitensuche über die Oberfläche ein neues Minimum gefunden, gilt dieses als neue minimale Distanz. Der Algorithmus in seiner ursprünglichen Form versagt allerdings bei Durchdringungen [71] und bei nicht konvexen Objekten, wie es bei der Gefäßumgebung der Fall ist.

Eine Erweiterung auf nicht konvexe Objekte und Durchdringungen stellt der V-Clip Algorithmus von Mirtich et al. [147] dar. Allerdings ist es dabei nötig alle nicht konvexen Objekte in konvexe Untergruppen zu unterteilen, was einen Teil der Komplexität auf die breite Phase verlagert.

Auf ähnliche Weise arbeitet auch die frei verfügbare I-Collide Bibliothek von Cohen et al. [54]. Auch hier ist allerdings die Behandlung nicht konvexer Objekte nicht ohne Unterteilung möglich.

In letzter Zeit gibt es vermehrt Bestrebungen, Kollisionsdetektion auf der GPU und, mit verbesserter Geschwindigkeit, im Bildraum auszuführen. Ein Beispiel dafür ist das Cullide [89] Verfahren von Govindaraju et al. Die gegenwärtigen Limitierungen (keine Messung der Durchdringungstiefe möglich, die Genauigkeit ist auf die Auflösung des Bildraums begrenzt) machen das Verfahren für die angestrebten Ziele aller-

dings unbrauchbar.

4.3.2 Verfahren in der kardiologischen Simulation

Ein Test der gängigen Verfahren und freien Bibliotheken ergab, dass die geläufigen Verfahren zur Kollisionsdetektion zwar für Partikelsimulation (siehe 5.5 gute Ergebnisse erzielen, für die harten Anforderungen der Instrumentensimulation in nicht konvexen anatomischen Umgebungen aber nicht ausreichend sind.

Die meisten der bekannten Simulationssysteme aus dem kardiologischen Bereich bauen daher auf spezialisierten Gefäßmodellen auf, um die Kollisionsdetektion in der schmalen Phase zu vereinfachen. So verwendet das Cathi System von Kornmesser et al. Zylinder Kugel Modelle [118], ebenso wie die Simulationsumgebung von Duriez et al. [130], in der Röhrenmodelle für Gefäße verwendet werden [121].

Kollisionsdetektionen auf Röhren werden bei Duriez in der breiten Phase durch manuell erstellte, an das Gefäßsystem angepasste Baumhierarchien und (in der schmalen Phase) durch einfache Abstandsberechnungen (Punkt / Zylinder) ausgeführt. Die Einschränkung auf röhrenförmige Primitive schließt das Verfahren in der vorliegenden Form für die Simulationssoftware dieser Arbeit aus.

Da die vorgestellte Simulation Dreiecksrepräsentationen für anatomische Strukturen verwendet wird im Folgenden nur noch auf solche Algorithmen eingegangen, die Kollisionsdetektionen im 3D Raum mit Dreiecksnetzen ausführen können.

Das neue Simulationssystem von Wang et al. [204] baut 2007 zwar auf Dreiecksnetzen auf und verwendet eine klassische BVH¹² Methode in der breiten Phase. Instrumentenknoten werden durch Kugeln approximiert. Die das Gefäßsystem bildenden Dreiecke sind durch entlang der Hauptachsen des Koordinatensystem ausgerichtete Bounding Boxen angenähert, die in einem vorberechnenden Schritt in eine Baumstruktur einsortiert werden.

Der Nachteil des Verfahrens ist, dass die Baumstruktur im Vorfeld berechnet werden muss und das System daher keine dynamischen Änderungen der Topologie während der Laufzeit erlaubt. Zudem ist das Verfahren bei Wang möglich, weil auf ein aufwendiges Optimierungsverfahren zugunsten einer Feder-Masse Implementierung verzichtet wird.

Abgesehen davon kann die Methode als ausreichend schnelles und robustes Verfahren der Kollisionsdetektion im generellen Fall (wie etwa bei Partikelkollision) gut eingesetzt werden. Für die Instrumentensimulation wurde in dieser Arbeit eine andere Form der Distanzberechnung gewählt, die vollständig unabhängig von der üblichen Kollisionsdetektion arbeitet.

4.4 Distanzberechnungen

Die realistische Simulation von Kathetern und anderen Instrumenten basiert in der Regel auf der Optimierung von Funktionen in hohen Dimensionen oder der Lösung großer Gleichungssysteme. Die Leistungsfähigkeit des verwendeten Rechners stellt dabei einen stark limitierenden Faktor dar, der über Optimierung der Funktionsberechnung und geschickt gewählte Abschätzungen deutlich verbessert werden kann.

¹²Bounding Volume Hierarchy

Gut 80% der Ressourcen während der Berechnungen werden für die Auswertung der auf die Instrumente wirkenden externen Kräfte und der damit einhergehenden Bestimmung der Eindringtiefe der Instrumente aufgewendet. Eine Optimierung in diesem kritischen Punkt stellt damit eine gewichtige Komponente dar. Das Problem ist prinzipiell eng verwandt mit Abstandsschätzungen und -berechnungen.

Eine besonders große Bedeutung spielen diese Berechnungen der Eindringtiefe in der Robotik und in der Pfadplanung, wo in Echtzeit Abstände zwischen deformierbaren Objekten geschätzt werden müssen. Gilbert et al. konnten in [86] einen Algorithmus vorstellen, der eine robuste Berechnung der Entfernung zweier konvexer Objekte im \mathbb{R}^3 erlaubt. Cameron zeigte in [48], dass sich das Verfahren theoretisch in linearer Komplexität lösen lässt. Die Beschränkung auf konvexe Objekte schließt eine Verwendung in der Simulation allerdings aus.

Cameron et al. zeigten in [49] ein heute als Standard geltendes Verfahren auf Grundlage der Minkowski-Summe, das aber abermals auf konvexe Formen beschränkt bleibt. Kim et al. schlugen in [110] eine inkrementelle Fassung des Algorithmus vor, die besonders gut funktioniert, wenn die zu prüfenden Objekte eine hohe zeitliche Kohärenz aufweisen.

Redon et al. präsentierten 2006 in [165] einen Algorithmus, der im Gegenteil zu den bisher vorgestellten Verfahren, auch auf Objekten beliebiger Topologien schnell und robust funktioniert - die Berechnung basiert auf einer geschickten Wahl der Kollisionsdetektion und einer anschließenden Partitionierung der kollidierenden Objekte - funktioniert allerdings nicht als reine Entfernungsabschätzung, da der Algorithmus bei nicht vorhandener Kollision zu keinem Ergebnis führen kann. Eine Verwendung in der kardiologischen Simulation ist somit nicht direkt möglich.

Parallel zeigten Sud et al. 2006 einen Weg, um die Eindringtiefe deformierbarer Objekte ohne Vorwissen in Echtzeit unter Zuhilfenahme diskreter Voronoi-Diagramme zweiter Ordnung zu berechnen [191]. Der Algorithmus ist derzeit aber noch nicht schnell genug, um in einer interaktiven Anwendung genutzt zu werden - Benchmarks zeigen, dass die Berechnung von 10 deformierbaren Objekten mit insgesamt ca. 5500 Dreiecken noch etwa 120 ms auf einem High End PC benötigt.

4.4.1 Distanzfelder

Da das Instrumentenmodell in der vorgestellten Arbeit durch diskrete Punkte repräsentiert wird, kann die Abschätzung der Eindringtiefe im Allgemeinen schnell und präzise durch ein im Vorfeld generiertes Distanzfeld geschehen. Eine erste Erwähnung von Distanzfeldern (in 2D) findet sich in [169], wo Rosenfeld und Pfaltz sie erstmalig zur Chamfer Distanz Transformation in Verbindung mit der Skelettierung einer Graphik nutzten.

Distanzfelder sind im Allgemeinen durch Position und Ausdehnung fest im \mathbb{R}^3 orientierte, diskrete Volumen, wobei jedes Feld (Voxel) des Volumens einen Wert erhält, der bezogen auf eine Objektoberfläche S die minimale Distanz zwischen (euklidischer) Voxelposition p und der Oberfläche δS des Objekts S repräsentiert.

$$dist_S(p) = \inf_{x \in \delta S} \|x - p\| \quad (4.15)$$

Von Interesse ist in der Regel auch das vorzeichenbehaftete Distanzfeld, bei dem die Voxelwerte der Funktion

$$\text{dist}_S(p) = \text{sgn}(p) \inf_{x \in \delta S} \|x - p\| \quad (4.16)$$

genügen. $\text{sgn}(p)$ stellt dabei die Orientierung des Punktes p (in- oder außerhalb des Objektes S) dar:

$$\text{sgn}(p) = \begin{cases} -1, & \text{wenn } p \in S \\ 1, & \text{sonst} \end{cases} \quad (4.17)$$

Ausgangspunkte für die Berechnung von Distanzfeldern sind üblicherweise entweder 3D-Gittermodelle oder Volumenmodelle. Der einfachste denkbare Ansatz ist dabei die Brute Force Methode. Bei dieser Methode wird für jede Voxelcoordinate p im Volumen ν die Distanz zu jedem Strukturprimitiv in S (bsp. Dreieck) berechnet und anschließend die kürzeste Distanz gespeichert. Der Ansatz ist zwar in der Regel robust, leidet aber unter extrem hohen Berechnungszeiten (polynomiell wachsend mit der Anzahl der Voxel und der Strukturprimitiven), so dass je nach verwendeten Ausgangsdaten verschiedene Möglichkeiten entwickelt wurden, um die Berechnung von Distanzfeldern zu beschleunigen.

Bei Vorliegen von Dreiecksnetzen, der häufigsten Repräsentation dreidimensionaler Objekte, muss im Vorfeld geprüft werden, ob die Bedingungen für die Berechnung eines Distanzfeldes erfüllt sind. Die Geometrie muss 2-manifold (keine Schnitte mit sich selbst, jede Kante grenzt an genau zwei Dreiecke, jeder Vertex wird von genau einem Kreis von Dreiecken umschlossen), geschlossen und bezogen auf ihre Oberflächennormalen gleichmäßig orientiert sein (nur wichtig bei vorzeichenbehafteten Feldern).

Payne und Toga zeigten in [158] einen Algorithmus zur schnellen Berechnung von Distanzen zu Dreiecken, wo sie auch diverse Methoden wie begrenzende Boxen für die Dreiecke oder Sortierung der Dreiecke in Baumstrukturen zur Beschleunigung der zur Generierung eines Distanzfeldes benötigten Kalkulationen vorschlugen. Allgemein ist der Algorithmus aber noch zu langsam um auf eine externe Vorberechnung und Speicherung der Daten verzichten zu können.

Geuzéic stellte in [90] einen anderen - als Meshsweeper benannten - Algorithmus vor. Begrenzte Boxen werden für geschickt gewählte Dreiecksgruppen erzeugt und anhand dieser Boxen minimale und maximale Distanzen zu Dreiecksgruppen berechnet. Die Gruppen werden in eine Prioritätsqueue einsortiert. Anschließend werden Gruppen, deren minimale Distanz die maximale Distanz bereits vorhandener Gruppen übersteigt, aus der Queue entfernt und somit die Rechenkomplexität minimiert. Der Algorithmus ist prinzipiell schnell, kann aber direkt keine Vorzeichen generieren und ist damit nicht für die Simulationszwecke geeignet.

Andere Methoden nutzen die Charakteristik des zu berechnenden Objekts aus. Bei feinen Strukturen haben einzelne Dreiecke nur in unmittelbarer Nähe maßgeblichen Einfluss auf das Distanzfeld. Mit zunehmender Entfernung wird die Wirkung der Dreiecke gemindert, und Approximationen bieten eine hinreichende Genauigkeit. Ein solches Verfahren wurde von Dachille et al. in [60] vorgestellt, wo Dreiecke bei zunehmender Entfernung zu approximierenden Gruppen zusammengefasst werden. Mauch

et al. schlugen in [142] und [143] vor, Features in Dreiecksnetzen zu detektieren und zu Polyedern (den sogenannten Charakteristiken) zusammenzufassen.

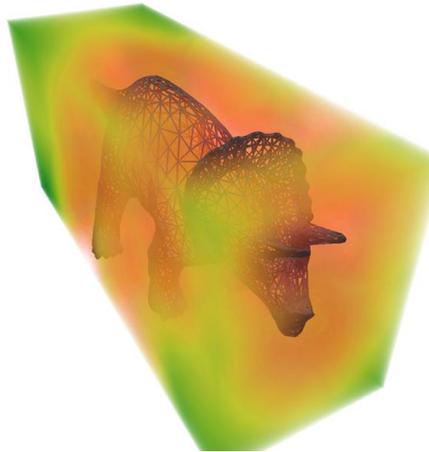


Abbildung 4.16: Beispielhafte Darstellung eines Distanzfeldes - berechnet nach einem Algorithmus von Sud et al. [191]. Rot gefärbte Bereiche sind nahe dem Objekt gelegen, grün gefärbte weiter entfernt.

Die Methode wurde von Sigg et al. aufgegriffen und derart implementiert, dass die GPU während des Rasterisierungsprozesses die Dreiecke in Charakteristiken konvertiert und die Distanzen in einem zweiten Rendschritt im Fragmentshader berechnet werden [183]. Die Methode ist schnell und kann ggf. auf Vorzeichenbehaftung erweitert werden.

Eine weitere auf Hardwarebeschleunigung basierende Methode präsentierten Sud et al. in [191], wo die räumliche Kohärenz von Primitiven ausgenutzt wird, um die zu testende Anzahl zu limitieren. Sud zu Folge ist die Implementierung um zwei Größenordnungen schneller als eine reine Softwarefassung des Verfahrens von Mauch ([142]).

Zur Berechnung des Vorzeichens schlugen die meisten Autoren, unabhängig von der Methode der Distanzberechnung vor, eine Scanlinien-Konvertierung zu verwenden ([104][142][158][190]). Jede Schicht des resultierenden, das gesamte Objekt umfassenden, Distanzfeldes in einer Richtung (bsp. Z-Ebenen) wird einzeln in eine 2D-Fläche konvertiert. Die Schnittlinie mit den Dreiecken resultiert in einer Kontur des Objekts. Den Punkt p kreuzende Schnittlinien in der X-Achse der Schicht werden dann genutzt, um das Vorzeichen zu bestimmen.

Eine auf den Oberflächennormalen der Dreiecke basierende Methode zur Bestimmung des Vorzeichens zeigten Aanæs and Bærentzen 2005 in [39] und [25]. Vertex- und Kantennormalen werden als achsengewichtete Pseudonormalen [195] berechnet. Die Methode funktioniert robust, solange das Modell wohldefiniert ist und keine Einschlüsse oder degenerierten (sehr kleine Oberfläche/Winkel) Dreiecke vorliegen, was bei automatisch aus CT-Daten generierten Meshes nicht immer garantiert werden kann¹³.

¹³Während der Konvertierung wurden mehrere externe Tools (Slicer, Cinema4D, Maya) genutzt, die im Schritt der Dreiecksminimierung stellenweise degenerierte Dreiecke erzeugt haben

Eine Methode, wie Distanzfelder auch noch aus Dreiecksgittern, die die oben genannten Voraussetzungen nicht erfüllen, erzeugt werden können, veröffentlichten Nooruddin et al. 2003 in [153]. Löcher werden geglättet, indem die Scanlinien-Konvertierung aus mehreren Projektionen heraus durchgeführt und die Ergebnisse entsprechend gewichtet werden - der Algorithmus funktioniert stabil, verschlingt aber viel Zeit.

2006 zeigten Erleben und Dohmann einen auf Scanlinien Konvertierung aufbauenden Algorithmus, der auf GPU implementiert war. Außerdem wurde die Oberfläche durch Tetraeder approximiert, was als guter Kompromiss zwischen Rechengeschwindigkeit und Genauigkeit gewertet werden konnte [70].

Alle vorgestellten Methoden generieren standardisierte Distanzfelder mit dem üblichen Nachteil der immens großen Datenmengen. Will man viele Felder oder hochaufgelöste Daten im Speicher halten, stößt man schnell an Ressourcenlimitierungen.

4.4.2 Adaptive Distanzfelder

Distanzfelder werden in der Regel als dreidimensionales Volumen repräsentiert. Da die Vielzahl natürlicher Modelle sich aber durch ein glattes Distanzfeld (besonders in größer werdender Entfernung zum Objekt) auszeichnen, ist diese Methode in Bezug auf den Verbrauch von Speicherressourcen und Zugriffsgeschwindigkeit (Cache) im Vergleich zu anderen Verfahren nicht ideal implementierbar. Frisken et al. schlugen daher in [85] vor, auf Octrees ([144]) basierende Strukturen zu verwenden und so die Auflösung der Distanzfelder zu verbessern. In [161] überarbeiteten die Autoren ihre Repräsentation von Distanzfeldern erneut in Bezug auf Zugriffsgeschwindigkeit - die eigentliche Idee der octreebasierten Speicherung blieb jedoch erhalten.

Ju et al. schlugen 2002 in [106] eine neue, schnellere und teilbasierte Generierung von adaptiven Distanzfeldern vor. Zeitgleich wurde eine ebenso auf Octrees fundierende ADF¹⁴ Methode von Barentzen et al. vorgeschlagen. Distanzwerte werden anstelle von dem in [85] vorgeschlagenen Verfahren an den Eckpunkten der Octreezellen gespeichert. Zu diesem Zwecke wird eine zusätzliche 3D-Hashtabelle generiert, die die Distanzwerte sichert und auf die die Zellecken verweisen. Auf diese Weise wird eine mehrfache Speicherung von Distanzen (benachbarte Octreezellen teilen Distanzwerte) vermieden.

4.5 Kontrastmittelsimulation

Auch wenn eine realistische Flussberechnung in der Kinderkardiologie eine geringere Bedeutung als die Instrumentensimulation einnimmt (im Gegensatz zur herkömmlichen Kardiologie sind Defekte meist größer und Diagnostiken zu Lokalität und Ausmaß des Defekts schon im Vorfeld durch MRT oder Ultraschall bekannt.¹⁵), ist eine angemessene Visualisierung wichtig für den Lerneffekt.

Die bekannten Verfahren, Flüssigkeiten zu simulieren, lassen sich grob in die Bereiche der Partikelmodelle (bsp. Smooth Particle Hydrodynamics) der gridba-

¹⁴Adaptive Distanzfelder

¹⁵Bei größeren Defekten (ASD/VSD) ist es üblich, Messungen mit speziellen Instrumenten - Messballonen vorzunehmen. Dafür ist allerdings kein Kontrastmittelfluss nötig.

sierten Verfahren (bsp. zelluläre Automaten) und der Textursynthese einteilen. Klar zu unterscheiden ist auch zwischen den physikalischen (in der Regel auf Navier-Stokes-Gleichungen aufbauenden) und den eher heuristischen Methoden (bsp. Perlin-Rauschen). In den letzten Jahren wurde zudem viel Aufwand betrieben, um Flüssigkeitssimulationen auf GPUs abzubilden.

Da die Simulationen von Flüssigkeiten und Gasen technisch recht eng miteinander verwandt sind und ggf. Teilbereiche beider Techniken für die Visualisierung von Kontrastmittel in Betracht gezogen werden können, werden beide Elemente im Folgenden häufig generalisierend unter dem Begriff Fluids zusammengefasst.

4.5.1 Partikelmodelle

Partikelmodelle gehen davon aus, dass Objekte als Gruppe einzelner Objekte (Partikel) mit festem Ort, Masse und Eigenschaften, die die Wechselwirkungen zwischen den Partikeln beschreiben, simuliert werden können. Findet keine Wechselwirkung zwischen den Elementen statt, spricht man von nicht gekoppelten Partikelsystemen.

Auch wenn Partikelsysteme schon seit langer Zeit fester Bestandteil der Computergraphik sind (1961 wurden beispielsweise in dem Spiel *Space War* nicht gekoppelte Partikelsysteme auf einem Oszilloskop dargestellt), wurden sie erstmals 1983 formell von Reeves et al. [166] zur Verwendung in der Computergraphik mit dem Ziel, diffuse Objekte zu simulieren, vorgestellt. Desbrun und Cani [64] sowie Tonnesen et al. [196] zeigten später weiterentwickelte Möglichkeiten, um weiche, zusammenhängende Objekte mit Partikeln zu simulieren.

Da Flüssigkeiten allerdings nur durch eine große Menge an Partikeln dargestellt werden können und theoretisch jedes Partikel mit jedem anderen über die aufwendig zu berechnenden Navier-Stokes-Gleichungen wechselwirkt ($O(n^2)$), waren Partikelmodelle zur Simulation von Fluiden über lange Zeit nicht ohne Weiteres einsetzbar.

Murta und Miller zeigten 1999 in [152] einen einfachen Ansatz, um das Verhalten von Partikeln heuristisch an das Verhalten an Flüssigkeiten anzunähern. Partikel werden über ein *Lennard-Jones-Potential* angezogen und abgestoßen. Große Partikel können sich unter Kraftaufwand in kleine Partikel teilen (Splitting) und wieder zusammenfügen. Die Übergänge werden mit einem einfachen Euler-Verfahren berechnet.

Das Verfahren kann visuell aber nur wenig überzeugen. Die Heuristiken verhalten sich bei längerer Betrachtung unnatürlich und können nicht alle Fluidphänomene korrekt darstellen. Zudem werden über die Potentialberechnungen hohe rechnerische Komplexitäten erzeugt.

Verwandt dazu ist die Idee der Smooth-Particle-Hydrodynamics. Diese Form der Partikelsimulation wurde 1977 von Monaghan und Gingold [87], aufbauend auf einem Artikel von Lucy [137] entwickelt und war ursprünglich für die Simulation astrophysischer Fragestellungen gedacht. Dank der sehr allgemeingültigen Formulierung des SPH Modells ist eine Verwendung in vielen anderen Simulationsrichtungen jedoch gut möglich, so dass die Modelle bereits 1995 von Stam und Fiume [189] zur Darstellung von Feuer und Gasphänomenen eingesetzt wurden.

Die grundlegende Idee der SPH ist es, nicht jedes Partikel mit jedem anderen wechselwirken zu lassen, sondern den Wirkungsbereich der Partikel mit zunehmendem Abstand bis auf 0 zu dämpfen. Außerhalb des Wirkungsbereiches wird keine Wechselwirkung mehr berechnet, so dass die Komplexität des Verfahrens stark abnimmt. Der

Wirkungsbereich wird über den sogenannten Wirkungskern bestimmt.



Abbildung 4.17: Fluidsimulation mit SPH von Müller et al. in verschiedenen Rendermodi. Partikel werden über Smooth Particle Hydrodynamics berechnet und anschließend gesplattet oder als zusammenhängendes Mesh gerendert.

Müller et al. [149] schlagen 2003 in ihrer Fluidsimulation mit SPH vor, die Partikel in ein dreidimensionales Hash einzusortieren, um die wechselwirkenden Elemente schnell gruppieren zu können. Die Geschwindigkeit ist allerdings mit ca. 200 ms pro Bild (inklusive Oberflächenrekonstruktion) bei 1300 Partikeln noch deutlich zu langsam, um in Echtzeitanwendungen genutzt werden zu können).

In [150] präsentieren die gleichen Autoren 2004 eine Erweiterung ihres Verfahrens auf die Anwendung zur Blutsimulation in der virtuellen Chirurgie. Takasi et al. zeigen in [35] eine GPU Variante des Algorithmus von Müller et al.

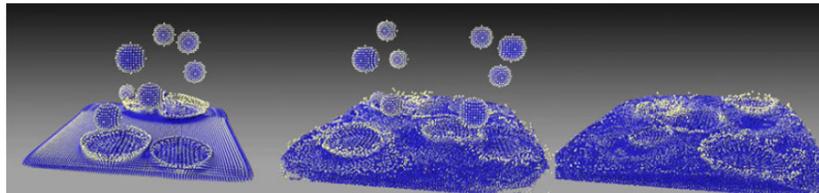


Abbildung 4.18: Fluidsimulation mit SPH auf GPU von Zhang et al. Smooth Particle Hydrodynamics werden auf der GPU gerechnet.

2007 zeigten Zhang und Solenthaler auf der Siggraph eine verbesserte GPU Implementierung [211], die durch bessere Sortierung der Elemente (in Buckets) bis zu 60000 Partikel bei einer Bildrate von 20 Bildern pro Sekunde (bei Vollaustattung) berechnen kann. Allerdings wird dabei auf Kollisionsberechnung und entsprechende Kollisionsantworten nahezu vollständig verzichtet.

4.5.2 Gridbasierte Methoden

Eine andere Art, Flüssigkeiten zu simulieren ist es, den ganzen Raum in eine feste oder variable Gitterstruktur einzuteilen. Jeder Knoten des Gitters verfügt über Eigenschaften (Kräfte, Dichte...) und kann dann - in der Regel mittels Integrationsmethoden über die Zeit gelöst werden. Der Nachteil ist, dass sich im dreidimensionalen Fall die

Komplexität des Gitters mit $O(n^3)$ erhöht, schon niedrig aufgelöste Strukturen also zu exponentiell hohen Rechenzeiten führen.

Versuche, die Berechnung gridbasierter Methoden unter Echtzeitbedingungen durchzuführen, gibt es seit ca. 1990. Kass und Miller haben in [108] einen Algorithmus vorgestellt, der die Oberfläche eines Fluids unter der vereinfachenden Annahme eines flachen Gewässers zweidimensional berechnet und die Höhe separat über ein Höhenfeld simuliert. Obwohl sich der Algorithmus nur bedingt für die Berechnung eines dreidimensionalen Fluids eignet, wurde er von Raghupathi 2002 zur Simulation von Blut in einem Simulator für minimal invasive Chirurgie verwendet [163].

1996 zeigten Foster und Metaxas in [73][74] erstmals eine Methode, die Navier-Stokes-Gleichungen in einer Echtzeitsimulation (1-2 FPS) zu implementieren. Verwendet wurde ein niedrig (16^3) aufgelöstes Gitter, auf dem Druck und Dichte der Flüssigkeit grob mittels numerischer Integration berechnet wurden. Der Ansatz wurde zur feineren Visualisierung mit einem Algorithmus zum Tracken der sich verändernden Oberfläche kombiniert. Die Methode ist allerdings schon auf niedrigen Auflösungen zu aufwendig, um in interaktiven Anwendungen Gebrauch zu finden.

1999 stellte Jos Stam [189] eine effizientere und vor allem stabile Methode vor, um die Navier Stokes Gleichungen auf einem fixen Gitter zu lösen. Stam verwendet einen Semi-Lagrangischen Ansatz, um die Gleichungen zu berechnen. Die notwendige Erhaltung der Masse wird durch Anwendung der Helmholtz-Hodge-Dekomposition auf das Geschwindigkeitsfeld erreicht. Der Algorithmus wurde 2000 und 2003 von Stam selbst erweitert und verbessert, so dass auch Umfließungen von Objekten gut berechnet werden können [185] [187].

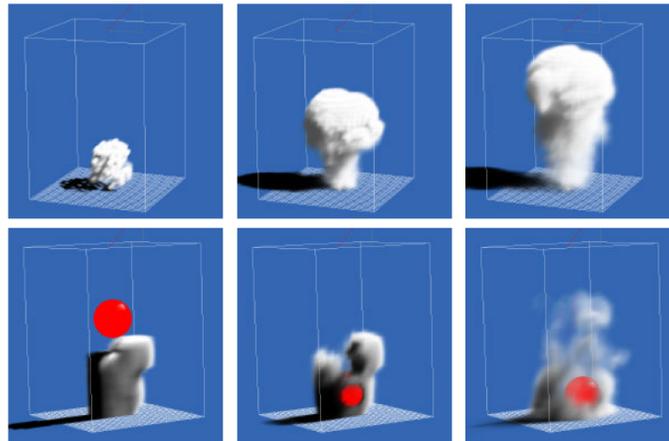


Abbildung 4.19: Gridbasierte Fluidsimulation von Jos Stam. Navier Stokes Gleichungen werden in einem Semi-Lagrangischen Ansatz auf einem fixen 3D Gitter berechnet.

2001 zeigte der selbe Autor eine weitere Methode, in der sein in [189] vorgestellter Algorithmus mit Hilfe einer schnellen Fourier-Transformation (FFT) weiter beschleunigt werden kann [186]. Die hohe Komplexität von $O(n^3)$ bleibt aber erhalten.

Die Methode von Stam wurde 2003 von Zatoryi et al. [210] verwendet, um ausfließendes Blut in einem Hysteroskopie Simulator nachzubilden. Aus Geschwindigkeits-

gründen wird aber auf eine echte 3D Simulation verzichtet und der Effekt nur visuell auf einem 2D Gitter (55×55 Zellen) abgebildet.

2007 konnte erstmals eine vollständig auf GPU implementierte Berechnung der Navier-Stokes-Gleichungen von Crane et al., basierend auf Stams Algorithmus, interaktive Frameraten bei nutzbarer Gittergröße (128^3) (und Vollaustattung der Grafikkarte) erreichen [59].

Eine weitere gridbasierte Technik basiert auf den 1966 durch Von Neumann vorgestellten zellulären Automaten [201]. Auf einem fixen Gitter werden Regeln aufgestellt, beispielsweise unter welchen Bedingungen Dichte zwischen den Gitterzellen ausgetauscht wird. Ansätze, das Verhalten von Fluids durch zelluläre Automaten zu simulieren, finden sich in den Arbeiten von Wei et al. [206][207] von 2004, wo die Lattice-Boltzmann-Methode angewandt wird, um Feuer und Gase auf zellulären Automaten ähnlichen Strukturen zu simulieren.

Zelluläre Automaten werden, trotz ihrer häufigen Defizite in Bezug auf exakte (realistische) Simulation, vor allem aufgrund ihrer guten Eignung zur Implementierung auf GPU immer bedeutsamer [207].

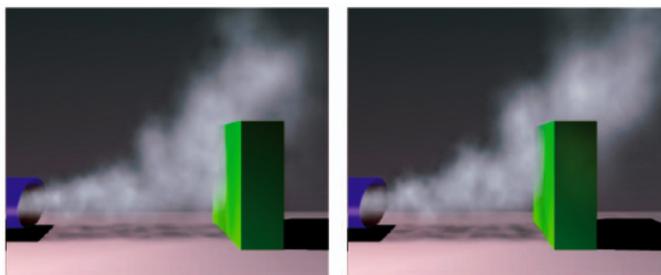


Abbildung 4.20: Gridbasierte Fluidsimulation von Wei auf Basis der Lattice-Boltzmann Methode

4.5.3 Textursynthese

Über die Zeit haben sich auch andere, eher heuristische Verfahren, zur Simulation und Visualisierung von Fluids etabliert. Einen großen Teil davon nehmen Methoden ein, die aus dem Bereich der Textursynthese stammen.

Als Textursynthese bezeichnet man ein Teilgebiet der Computergrafik in dem Bild-daten mit ausreichend statistisch wiederholbaren Eigenschaften (Texturen) in beliebiger Dimension erzeugt werden sollen, die in bestimmten Kriterien (bsp. visuell oder statistisch) nicht von natürlichen Vorlagen zu unterscheiden sind. Grundsätzlich wird dabei zwischen prozeduralen Syntheseverfahren, bei denen keine reale Vorlage genutzt wird (Texturen werden allein aufgrund von Formeln generiert), und vorlagenbasierte Verfahren unterschieden, die aufgrund verschiedener Verfahren versuchen, den Stil einer gegebenen Vorlage nachzubilden.

Die wahrscheinlich bekannteste prozedurale Technik basiert auf dem 1985 vorgestellten Perlin-Rauschen [159]. Mehrere glatte Funktionen unterschiedlicher Frequenz und Amplitude werden addiert, um eine geglättete Rauschfunktion zu erhalten. Im

2000 zur Simulation endoskopischer Eingriffe von Kühnapfel et al. vorgestellten Verfahren [47] wird 3D-Perlin-Rauschen (mit Zeit als dritter Dimension) verwendet, um animierte 2D-Texturen für sich entlang vordefinierter Splines bewogender Billboard Partikel zu generieren.

Der Nachteil der Methode ist, dass sie aufgrund ihrer heuristischen Herkunft stark parametrisch justiert werden muss, um ansprechende Resultate zu generieren. Sie lässt sich nur schwer auf wechselnde Umgebungen und Druckverhältnisse anpassen und erfordert für jedes neue Modell eine erneute Parametrisierung. Einen großen Vorteil bietet jedoch die im Vergleich hohe Geschwindigkeit des Verfahrens.

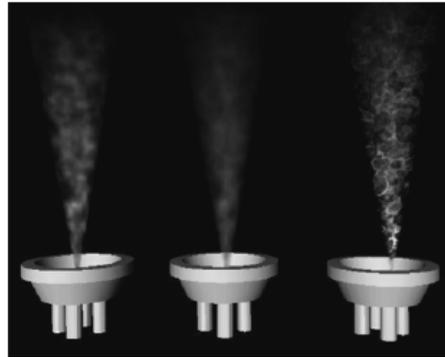


Abbildung 4.21: Mit Perlin Rauschen simulierter Rauch von Kuehnappel et al.

Perlin selbst zeigte 2001 in [160] ein "Flow Noise" genanntes Verfahren bei dem die einzelnen Funktionen durch räumliche Rotationen der Zentren gegeneinander verschoben werden um ein Fluid ähnliches Verhalten zu simulieren.

Das Verfahren muss allerdings mit geeigneten, physikalisch motivierten Fluidsimulationen kombiniert werden um wirklich ansprechende Effekte zu erzielen. 2006 zeigten Augustin et al. [36] ein solches, in 2D auf GPU implementiertes Verfahren, das bei geeigneter Anpassung dazu genutzt werden könnte, Fluidsimulationen auf niedrig aufgelösten Gittern optisch aufzuwerten und ohne sichtbaren Qualitätsverlust in höheren Auflösungen darzustellen (siehe Abbildung 4.22).

Neben den Ansätzen von Perlin existieren noch eine Reihe weiterer Verfahren, die aus Textursynthesen motiviert sind. Da es sich bei Fluidsimulationen um bewegte Texturen handelt, spricht man in der Regel auch von dynamischer oder temporaler Textursynthese.

Von Lai und Wu stammt ein Verfahren, das Texturen aus einer einzigen Bildvorlage über einen Min Cut Algorithmus (Teile aus einer Textur werden an möglichst idealen Trennkanten herausgelöst und mittels Alpha Blending neu zusammengesetzt) synthetisiert [124]. Anschließend wird zwischen diesen Texturen über einstellbare Parameter gemorphet, um Bilder mit einem zusammenhängenden, grafischen Fluss zu erzeugen. Das Verfahren ist im Paper auf 2D limitiert, ließe sich allerdings mit geringem Aufwand auf 3D portieren. Die Resultate sind allerdings nicht ausreichend ansprechend, um den Aufwand zu rechtfertigen. Die Autoren schlagen allerdings vor, die Morphingparameter in Zukunft physikalisch motiviert anzupassen, um den Flussbedingungen realer Phänomene gerecht zu werden.

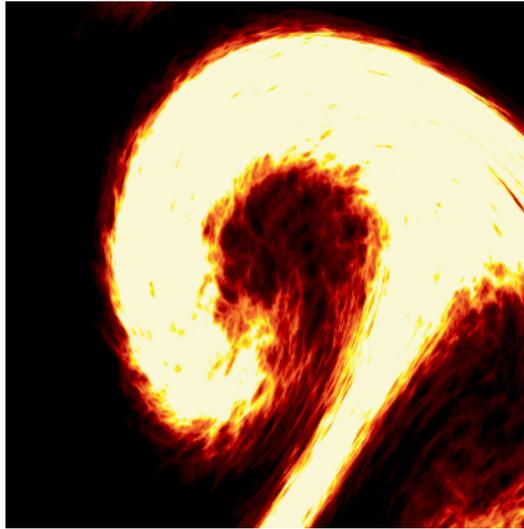


Abbildung 4.22: Flow Noise (modifiziertes Perlin Rauschen) wird verwendet, um eine niedrig aufgelöste Fluidsimulation in höherer Auflösung zu rendern (Bild von Augustin et al.[36]).

Einen sehr schnellen Algorithmus zur Synthese von Texturen zeigten 2000 Wei und Levoy. Texturen werden auf der Basis von MRF-Textur Modellen¹⁶ generiert, teilen aber nicht deren inhärenten Nachteil der oft extrem langen Berechnungszeit. Der Syntheseprozess ist in der Publikation grundlegend als Suche nach lokalen Ähnlichkeiten in einer Vorlagetextur definiert. Da die Suche den zeitaufwendigsten Teil der Synthese darstellt, wird der Suchprozess auf eine Suche nach dem am nächsten gelegenen Punkt abgebildet, indem Nachbarschaften als Vektoren in einem multidimensionalen Raum betrachtet werden. Die Suche selbst wird mittels der Erstellung eines binären Suchbaums nach dem TSVQ¹⁷ Verfahren beschleunigt. Somit ist es möglich, 2D Texturen in sehr schneller Zeit zu erzeugen und das Verfahren auf temporale (in Form von 3D) Texturen auszudehnen.

Die Ergebnisse verfügen allerdings prozessbedingt (Eine einzige 2D Textur dient als Eingabe) nur über lokale Ähnlichkeiten und sind daher für die Generierung der typischen Merkmale von Fluiden (Wirbel, Strömungen) eher wenig geeignet.

Soatto et al. zeigten 2001 in [172] ein Verfahren, um dynamisch animierte Texturen in 2D von nur wenigen Bildvorlagen zu erzeugen. Zu diesem Zweck werden dynamische Texturen als Realisierungen der Ausgabe eines dynamischen Systems definiert, das von einem unabhängig und gleichverteiltem (IID) Prozess angetrieben wird. Das System kann mittels "Total Likelihood" und "Prediction Error" Kriterien belehrt und anschließend synthetisiert werden. Das Verfahren arbeitet allerdings nur in 2D und ist nicht für den Echtzeitbetrieb ausgelegt.

Einen etwas andere Weg stellten Masiero et al. 2006 in [140] vor. Geht Soatto noch von einer linearen Repräsentation dynamischer Texturen aus, nutzt Masiero einen

¹⁶Markov Random Field

¹⁷Tree Structured Vector Quantization

Monte Carlo Ansatz um Texturen aus einem nicht explizit gegebenen nicht linearen Modell zu schätzen. Der vorgestellte 2D Ansatz hat aber weitere, negative Einflüsse auf die Rechenzeit und bedarf großer Mengen an Vorgabetexturen um brauchbare Ergebnisse zu erzielen.

Viele Bestrebungen gehen auch dahin, bereits bestehende Fluidsimulation mit Textursynthesen visuell durch Texturierung aufzuwerten.

So präsentierten Narain et al. 2007 ein Verfahren, um Texturen aus Bild- oder Videovorgaben, basierend auf physikalischen Charakteristiken zu synthetisieren. Die Ergebnisse erlauben die Generierung realistisch wirkender Texturen, die mit dem Fluss einer vorgegebenen Fluidsimulation korrespondieren und komplexere Effekte wie Schaum oder Blasenbildung simulieren können. Das Verfahren ist allerdings zu aufwendig, um in Echtzeit berechnet werden zu können.

4.6 Modellanimation

Da das Herz ein Organ ist, das sich ständig in Bewegung befindet, ist es für eine realistisch wirkende Simulation unumgänglich, eine Form der Bewegungssimulation zu implementieren. Da die Daten (nach entsprechender Vorverarbeitung) als 3D-Dreiecksgitter vorliegen, beschränkt sich der folgende Abschnitt auf die Vorstellung verschiedener Methoden zur Animation von Gittermodellen.

4.6.1 Physikalische Deformation

Auch in anderen VR Simulationssystemen ist die Deformation elastischen Gewebes ein zentraler Bestandteil. So präsentierten Cover et al. 1993 ein Trainingssystem für Laparoskopie bei dem Objekte über Spline Deformationen deformiert wurden [58]. Für ein ähnliches Trainingssystem nutzten Baur et. al 1998 spezielle Deformationsfunktionen um die Berührung und das Schneiden durch virtuelles Gewebe zu ermöglichen [41].

Logan et al. sowie Gibson et al. zeigten 1996 ein haptisches Trainingssystem für Arthroskopie mit deformierbaren Objekten, die durch Feder Masse und 3D-Chain-Mail Techniken simuliert wurden [135] und teilweise auf Volumenmodellen aufbauten [83].

Koch et al. stellten ebenfalls 1996 ein System zur Planung von gesichtschirurgischen Eingriffen vor, bei dem Gesichtsgewebe mittels Feder Masse und Finite Elemente Methoden simuliert wurde [114].

Szekely zeigte 1998 ein System um chirurgische Eingriffe in der Gynäkologie zu trainieren. Der Fokus des Systems liegt auf der realistischen, nicht linearen, elastodynamischen Deformation des Gewebes mittels Finite Elemente Methode, weshalb die Simulation in Echtzeit nur auf einem Rechencluster vieler vernetzter Prozessoren berechnet werden konnte [192].

Ebenso kann deformierbares Gewebe mit Hilfe freier Frameworks zur medizinischen Simulation (SOFA [34], Spring [148], Gipsi [88]) implementiert werden.

Allerdings haben alle vorgestellten Systeme gemein, dass die Deformation in der Regel auf das dynamische Wirken äußerer Kräfte zurückzuführen ist. Die Herzdeformation wird aber, wenn überhaupt, nur zu einem sehr geringen Teil durch die äußerst

weichen Instrumente beeinflusst, weshalb die selbsttätige und zyklische Animation des Herzmuskels in den Vordergrund rückt.

Für diese Art der Deformation existieren andere Verfahren, die zwar nicht physikalisch motiviert sind, optisch aber ebenso überzeugend sind und mit einem Bruchteil der Rechenzeit auskommen.

4.6.2 Manuelle Deformation

Die Animation von Gittermodellen lässt sich im gegebenen Kontext grob in Bereiche einteilen. Die direkte (Keyframe basierte) Animation, bei der Gitterpunkte direkt manipuliert werden und die indirekte (Deformer basierte) Animation, bei der sich die Gitterpunkte aufgrund der Konfiguration eines umschließenden und beschreibenden Objektes bewegen.

Direkte Animationen sind in der Regel nur unter großem manuellen Aufwand oder bei Vorliegen sehr gut trackbarer (bsp. Motion-Capturing) Daten zu erstellen. Zudem liegen nach Erstellen der Schlüsselbilder große Datenmengen vor, die entsprechend zu verarbeiten sind.

Indirekte Animationsverfahren haben dagegen den Vorteil, Bewegungen gut parametrisierbar und über eine vergleichsweise geringe Anzahl an Kontrollpunkten steuerbar zu machen. Nachteilig zeigt sich hingegen der Mangel, direkten Einfluss auf Vertices nehmen zu können.

Nicht jede Deformation lässt sich gut durch ein indirektes Verfahren approximieren. Komplexe Verformungen wie beispielsweise die Beine eines laufenden Lebewesens können nur mit viel Aufwand und entsprechend vielen Kontrollpunkten umgesetzt werden. Für die (relativ) gleichmäßig skalierende und rotierende Animation eines Herzens sind diese Techniken allerdings gut geeignet.

1986 zeigten Sederberg et al. ihr bis heute in Variationen verwendetes Verfahren zur Free Form Deformation [179] bei dem globale Deformationsvolumina inneliegende Objekte beeinflussen (siehe Abschnitt 4.6.3).

Ein weiteres Verfahren, zeigten 1992 Metaxas und Terzopoulos in [146] mit dem Unterschied, dass es zwar lokalere Einfluss als die FFD auf Objekte nehmen kann (lokale Deformationen werden als Finite Elemente Form Funktionen berechnet), jedoch auf die Deformation parametrisch beschreibbarer Objekte beschränkt bleibt. Da diese Einschränkung eine Anwendung im vorliegenden Fall verhindert, lohnt hier eine nähere Betrachtung der Free Form Deformation.

Andere, mehr auf die Animationsbehandlung durch Graphiker ausgelegte Verfahren, die häufig große Rechenressourcen verbrauchen und daher nicht für den Einsatz in der Simulation geeignet sind wie etwa optimierungsbasierte BCM¹⁸ Algorithmen [113] oder Control Handle basierte Verfahren, die nicht immer glatte (und somit im gegebenen Kontext visuell überzeugende) Deformationen erzeugen [133] sollen hier nicht weiter betrachtet werden.

4.6.3 Free Form Deformation

Ein häufig verwendetes Verfahren der indirekten Animation ist die Free-Form-Deformation, wie sie in [179] von Sederberg und Parry vorgestellt wurde. Ein Punkt p

¹⁸Boundary Constraint Modeling

innerhalb eines definierten Gitters wird mit Hilfe trivariater Bernsteinpolynome durch Anwendung der Vektorfunktion

$$FFD(u, v, w) = \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n P_{ijk} \theta_i^l(u) \theta_j^m(v) \theta_k^n(w) \quad (4.18)$$

in einen Zielraum deformiert. Dabei sind l , m und n die Grade der Bernstein Polynome in der U , V und W Achse. $P_{i,j,k}$ sind die Kontrollpunkte und θ_i^l bezeichnet die i^{th} Basisfunktion vom Grad l .

Probleme bei der Anwendung der FFD ergeben sich durch die nicht vorhandene Lokalität der FFD Funktion (die Veränderung eines beliebigen Kontrollpunkts beeinflusst die Deformation des gesamten Volumens).

Hsu et al. zeigten in [99] ein Verfahren, das es ermöglicht, unter aufwendiger Berechnung einer pseudoinversen Matrix, direkt animierte Objekte in FFD-animierte Objekte zu überführen. Das Verfahren konnte 2001 von Hu et al. deutlich beschleunigt werden, indem eine effizientere und explizite Lösung für das Problem gegeben wird [101].

Eine weitere Beschränkung der klassischen FFD ist das zwingend quaderförmige Kontrollgitter. Einen Weg, diesen Mangel zu beheben, präsentierte Coquillart in [55]. Die dort gezeigte EFFD¹⁹ kann über einen beliebig geformten Raum definiert werden. Zu diesem Zweck werden mehrere Beziervolumen zu einem globalen Raum kombiniert. Der Nachteil des Verfahrens ist allerdings, dass Freiheitsgrade verloren gehen, da sie verfahrensintern dazu verwendet werden müssen, die Kontinuität zwischen den Volumen zu gewährleisten. Dieses Problem wurde von MacCracken und Joy gelöst, indem das Volumen durch Subdivisions Oberflächen angenähert wurde [198].

Kobayashi und Ootsubo gehen in [112] einen anderen Weg, in dem das Volumen für die Deformation durch ein Dreiecksnetz repräsentiert werden kann. In ihrem t-FFD genannten Verfahren definiert jede Dreiecksfläche ein affines Koordinatensystem über das jeder Punkt p innerhalb des Volumens parametrisch beschrieben werden kann. Durch Deformation des Dreiecksnetzes ändert sich die Position der parametrisierten Punkte.

Ein weiteres Problem der FFD ist, dass sie zu Schnitten des deformierten Modells mit sich selbst führen kann (bei nicht beschränkter Bewegungsfreiheit der Kontrollvertices). Gain und Dodgeson schlagen daher in [77] vor, jeden Deformationsschritt auf Modellüberschneidungen zu testen und - bei positivem Test - die Überschneidungen durch iterative Korrektur der Deformation aufzulösen. Einen ähnlichen Ansatz gehen Hirota et al., die in [98] eine Methode zeigen, FFD-Deformationen volumenerhaltend durchzuführen.

Griessmair und Purgathofer schlagen vor, B-Spline-Basisfunktionen anstelle der Bernstein-Polynome zu verwenden, da sie lokaler sind und so auch größere Gitter ohne zusätzliche Rechenlast erlauben. Weitere Untersuchungen zur Verwendung anderer Basisfunktionen finden sich unter [107], wo Kalra et al. B-Spline basierte Basisfunktionen zur Simulation von Muskelbewegungen mit FFD-Netzen verwenden oder unter [125], wo die Verwendung von NURBS in FFDs gezeigt wird. In einem neueren An-

¹⁹Extendet Free Form Deformation

satz schlugen Botsch und Kobbelt vor, volumenbasierte radiale Basisfunktionen zur Deformation zu verwenden.

Im medizinisch technischen Umfeld wird die Free-Form-Deformation meist zur Nachbildung diffuser, räumlich begrenzbarer Deformationen eingesetzt. Rohlfing et al. zeigen in [168] den Nutzen der FFD, um verschiedene Bildgebungsmodalitäten (MR) nicht rigide zu matchen. In [34] wird die FFD in einem medizinischen Simulationsframework verwendet, und in [180] wird eine um die Behandlung von Diskontinuitäten erweiterte Version der FFD (D-FFD) mit FEM Methoden kombiniert, um Einschnitte in einem chirurgischen Umfeld zu simulieren.

Wenige Bestrebungen gibt es bisher, Free-Form-Deformationen auf GPU umzusetzen, da sich die Architektur moderner Grafikkarten nur bedingt für die benötigten Berechnungen eignet. 2005 wurde ein möglicher Ansatz von Schein et al. in [176] gezeigt.

Nur wenig ist in der Literatur zur Invertierung der FFD zu finden. Auch wenn eine Invertierung aufgrund der in [77] geschilderten Überschneidungsproblematik (die klassische FFD ist nicht bijektiv, damit also kein Diffeomorphismus) nicht eindeutig lösbar ist, lassen sich unter definiertem Ausschluss von Raumüberschneidung mit vertretbarem Aufwand inverse Approximationen zu einer Free-Form-Deformation finden. Die für diese Arbeit entwickelte und verwendete Methode wird in Kapitel 5.4 vorgestellt.

4.7 Simulations- und Systemarchitekturen

Auch wenn über die Architektur kommerzieller Simulationssysteme nur wenige Informationen öffentlich verfügbar sind, lässt sich der Stand der Technik in Bezug auf die Systemarchitekturen freier Simulationsframeworks prüfen.

Da die Entwicklung einer Simulation im Bereich der Medizin in der Regel viele Bereiche, von Grafik über Physik, numerische Integration, Kollisionsdetektion, User Interfaces etc. umfasst, existieren mehrere Projekte, die versuchen diese verschiedenen Disziplinen über ein offenes Framework einer größeren Forschergemeinde zur Verfügung zu stellen [148] [88] [34].

Die üblichen Nachteile solcher "Allzwecksimulationen" sind, dass die entstehende Codebasis häufig sehr allgemeingültig formuliert werden muss, um für große, disziplinenübergreifende Bereiche von Nutzen zu sein. Als Folge kann Code entstehen, der so generisch ist, dass problemspezifische Implementierungen deutlich erschwert werden [148].

Derzeit existieren mit Spring [148], GiPSi [88], Vrass²⁰ [109] und SOFA [34] vier bekannte Open Source Implementierungen von Simulationsframeworks, die jeweils unterschiedliche Philosophien verfolgen, was eine genauere Betrachtung nötig macht. Weiterhin existieren Frameworks die hauptsächlich auf die Simulation haptischen Feedbacks spezialisiert sind (Chai3D [5] und H3D [7]) und keinen Rahmen für eine vollständige medizinische Simulation bieten. Andere - inzwischen durch neuere Implementierungen ersetzte Frameworks wie etwa CAML [57] werden in dieser Zusammenfassung nicht weiter betrachtet.

²⁰Über Vrass sind leider nur wenige Informationen in Europa verfügbar, da das System zum aktuellen Zeitpunkt nur in Japan eingesetzt und entwickelt wird

4.7.1 Spring

Das erste frei verfügbare Framework für die Echtzeit Simulation von chirurgischen Eingriffen - Spring - wurde 2002 von Montgomery et al. vorgestellt [148]. Es existiert auf Windows und Linux Plattformen, nutzt POSIX Threading und OpenGL und unterstützt die Formate SMF, Wavefront OBJ, VRML, Mesh und Cyberware. Es ist in der Lage Gewebe (Soft-Tissue), einfache Rigid Objekte und chirurgische Fäden physikalisch zu simulieren. Kollisionsdetectionen werden über Bounding Spheres berechnet und einige zusätzliche Features (Stimmen Ein-/Ausgabe, Echtzeit Texturemapping von Videodaten, Stereo Displays) sind bereits im Framework implementiert.

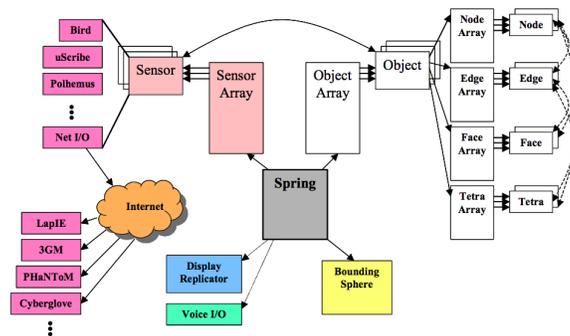


Abbildung 4.23: Grobe Darstellung des Spring Frameworks aus [148]

Die grobe Architektur von Spring ist in Abbildung 4.23 dargestellt. Der Hauptthread der Simulation geht von dem zentralen "Spring" Objekt aus. Der Simulationskern sieht vor, dass Objekte, die als Kanten, Flächen, Knoten oder Tetraeder repräsentiert werden (mehrfach Repräsentationen sind möglich) beinahe ausschließlich durch Feder Masse Physik berechnet werden. Rigide Objekte werden nur am Rande unterstützt, können aber über die Node Architektur mit Feder Masse simulierten Objekten verbunden werden.

Es existieren Löser für Runge Kutta 2. und 4. Ordnung sowie das explizite Euler Verfahren. Ferner wird ein quasi-statischer Löser implementiert, der davon ausgeht, dass sich das System stets im statischen Gleichgewicht befindet und Masseträgheit vernachlässigt. Für eine genauere Beschreibung des Systems wird hier auf [148] verwiesen.

Die Codebasis von Spring ist sehr auf chirurgische Simulation ausgelegt. Zudem sind nur Feder Masse Simulationen implementiert, deren interne Vergleichbarkeit auf den Einsatz verschiedener Lösungsverfahren beschränkt ist. Aufwendigere Verfahren wie FEM oder andere Mittel der Kollisionsdetektion können nicht ohne Weiteres zusätzlich implementiert werden.

4.7.2 GiPSi

Ein ähnliches Ziel wird vom 2004 von Cavusoglu et al. vorgestellten GiPSi (General Interactive Physical Simulation Interface) Framework verfolgt, jedoch unter Verwendung völlig anderer architektonischer Philosophien [88]. Während Visualisierung und

Interfaces nur einen sehr geringen Teil des Frameworks einnehmen, wird der große Fokus auf die physikalische Simulation gelegt, die abstrahiert in einem möglichst flexiblen Framework zusammengefasst wurde.

Alle zu simulierenden Objekte müssen von ihren entsprechenden Tools (Finite Elemente, Feder Masse..) abgeleitet werden und anschließend unter Zuhilfenahme spezieller Berechnungswerkzeuge (bsp. Differentialgleichungslöser) gelöst werden (siehe auch Abb. 4.24).

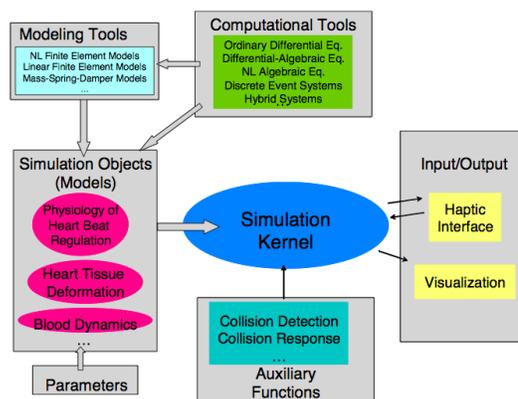


Abbildung 4.24: Grobe Darstellung des GiPSi Frameworks aus [88]

Sowohl Finite Elemente als auch Feder Masse Physik kann abgebildet werden. Objekte müssen dafür in einem Format (Domain Geometry) vorliegen, das das Objekt bereits in seiner diskretisierten Form abbildet. Sowohl traditionelle meshbasierte Objekte als auch aus Punktwolken bestehende Körper (bsp. Tetraeder Meshes) können abgebildet werden. Da Objekte jedoch miteinander agieren können muss von jedem Objekt mindestens eine allgemein gültige Darstellung in Form eines triangulierten Oberflächennetzes vorliegen.

Die Flexibilität des Systems wurde auf Seiten der physikalischen Simulation gewährleistet - graphische Ausgaben und andere Komponenten, die für ein offenes allgemein gültiges Framework eine ähnliche Erweiterbarkeit benötigt hätten, sind bei GiPSi aber eher starr orientiert.

4.7.3 Sofa

Das neueste und erfolgsversprechendste Framework für medizinische Simulation wurde 2007 von Allard et al. vorgestellt [34]. Bei Sofa handelt es sich um ein Kooperationsprojekt zwischen vier Forschergruppen (The Sim Group von der Harvard Medical School sowie die Gruppen Evasion, Alcove und Asclepios vom INRIA Institut), mit dem Ziel ein System zu schaffen, das die schnelle Entwicklung beliebiger medizinischer Simulationen in hoher Qualität und mit hoher Flexibilität ermöglicht. Gleichzeitig soll ein Austausch zwischen Forschergruppen gefördert und eine neue, auf XML basierende Beschreibungssprache für anatomische und biomechanische Datensätze geschaffen werden.

Das Basiskonzept hinter Sofa ist die mehrfach unterschiedliche Repräsentierung aller vorhandenen Modelle. So ist es beispielsweise möglich - und sogar erwünscht - jedes Objekt in mehreren Beschreibungen zu repräsentieren, die für die jeweilige Disziplin (Grafik, Kollisionsdetektion, Physik) ideal sind. Erreicht wird das über die Einführung sogenannter Mapping Vorschriften, die jede Repräsentation in ihre entsprechenden Gegenstücke - ggf. iterativ - überführen kann. Ein Beispiel für die verschiedenen Darstellungsformen zeigt Abb. 4.25

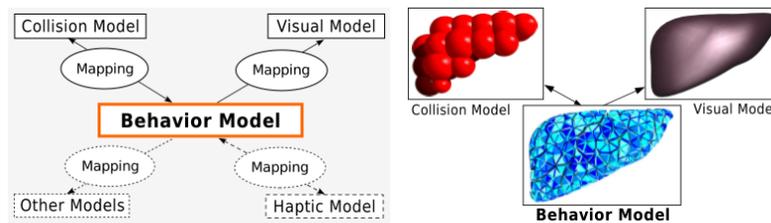


Abbildung 4.25: Darstellung der Multi Model Repräsentierung im SOFA Frameworks aus [34]

Um auf der Ebene der Verfahrenswahl (bsp. Löser für physikalische Simulation) eine hohe Modularität zu gewährleisten, werden die Komponenten, die die meisten physikalischen Simulationen gemein haben (Freiheitsgrade, Kraftfelder (bsp. FEM, SPH), Massen und Löser) als generische Primitive abstrahiert.

Das Sofa Modell bildet die gesamte interne Welt als Szenengraphen ab, wobei verschiedene Darstellungen (Mappings) von Objekten als Kinder der entsprechenden Objekte im Szenengraphen auftauchen (siehe Abbildung 4.26).

Moderne Hardwarearchitekturen wie das CUDA Framework für GPU Processing werden in Ansätzen von Sofa unterstützt. Eine Erweiterung auf Multiprocessing auf Threadbasis ist für eine der kommenden Veröffentlichungen geplant. Das gesamte System kann über XML definiert werden, so dass im Idealfall eine komplette Simulation nur über ihre Definitionen geschrieben werden kann ohne den konkreten C++ Code zu verändern.

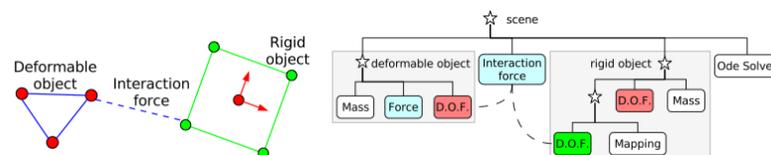


Abbildung 4.26: Szenengraph aus [34]. Links: Zwei interagierende Körper. Die Freiheitsgrade sind als Kreise, die Kräfte als Linien eingezeichnet. Durchgezogene Linien repräsentieren interne, gestrichelte Linien externe Kräfte. Rechts ist der zugehörige Szenengraph zu sehen.

Da das Sofa Framework noch sehr neu ist (bei Veröffentlichung der Spezifikationen war der Großteil der vorliegenden Arbeit bereits abgeschlossen), lässt sich noch

nicht beurteilen wie und ob das System in der zukünftigen Entwicklung von Simulationen eine Rolle spielen wird. Durch die sehr modulare Architektur und die saubere Definition der Schnittstellen, die eine wirkliche Erweiterbarkeit in allen Bereichen gewährleisten, ist die Chance aber gegeben, dass Sofa die zukünftige Entwicklung von medizinischen Simulationen maßgeblich beeinflussen kann.

4.8 Beschreibungssprachen

Um Geometrien in Anatomien effizient zu beschreiben, haben sich einige Beschreibungssprachen, die zumeist auf XML basieren, etabliert. Im Folgenden sollen diese kurz vorgestellt werden.

4.8.1 AnatML

Eine der bekannteren Sprachen zur Beschreibung von anatomischer Geometrie im menschlichen Körper ist AnatML [14]. Aus dem Physiome Projekt zur digitalen Beschreibung funktionaler Zusammenhänge zwischen biologischen Komponenten hervorgegangen, sollte die auf XML basierende Sprache auf einfache Weise eine Gliederungsstruktur zwischen Organen herstellen. Für vielfältigere Beschreibungen wird AnatML häufig mit weiteren XML basierten Beschreibungssprachen wie MeshML (Meshes), FieldML (Basis Funktionen und Parameterbeschreibungen) und RegionML (Containerformat) gekoppelt.

AnatML verwendet nur wenige Tags um die Anatomie zu beschreiben. Beschrieben werden Körperteile (`<body part>`), die die normalisierten Geometrien und CMISS Datenstrukturen enthalten, Körpergruppen (`<body group>`), die Links zwischen verbundenen Körperteilen herstellen und Platzierung (`<placement>`) Tags, die Körperteile im Raum orientiert platzieren.

Auch wenn die vorliegende Arbeit ein AnatML ähnliches Format verwendet, hat sich AnatML als spezialisierte Beschreibung von organischen Strukturen nicht durchgesetzt und wird heute kaum noch verwendet, da mit generalisierteren Beschreibungssprachen wie etwa RDF (Resource Data Format) oder Ontology generischere Lösungen vorliegen.

4.8.2 SSTML

Deutlich komplexer als AnatML gestaltet sich SSTML. Die "Surgical Simulation and Training Markup Language", die 2006 von Bacon et al. vorgestellt wurde hat zum Ziel, vollständige Simulationsvorgänge in einem XML basierendem Format zu beschreiben [38]. Dabei repräsentiert SSTML den gesamten Vorgang von den organischen Modellen (inkl. Gewebeeigenschaften) bis hin zu chirurgischen Prozeduren.

Als Weiterentwicklung bekannter Simulationsbeschreibungssprachen wie DIS²¹ oder HLA²² sind die Spezifikationen der Sprache offen einsehbar.

SSTML verwendet mehrere XML Container, jeder in einem eigenen Namespace, um Module für Haptisches Feedback, Darstellung, Tracking der Werkzeuge, die (chirurgische) Simulationsumgebung und Trainingsprozeduren separat zu beschreiben. Zur

²¹Distributed Interactive Simulation

²²High Level Architecture [61].

```

<surfacePropertyDeformationProcessorMap>
  <element>
    <key>kidney-surface</key>
    :
    <key>scalpel-surface</key>
    <value>
      <radiusOfInfluenceDeformationProcessor>
        <defaultRadiusOfInfluence>0.05
      </defaultRadiusOfInfluence>
    </radiusOfInfluenceDeformationProcessor>
    :
  </surfacePropertyDeformationProcessorMap>

```

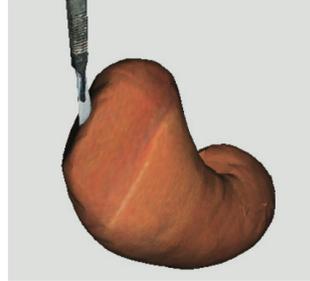


Abbildung 4.27: Beispielhafte SSTML Daten, die einen Deformationsalgorithmus beschreiben [38] und die resultierende Simulation aus [69].

eigentlichen Beschreibung werden für jeden Container vielfältige XML Tags definiert, die unter [15] genau beschrieben sind. Prozeduren können beispielsweise verschiedene gültige Vorgehensweisen als Baumstruktur erfassen - Simulationsumgebungen explizite Algorithmen, Interaktionszusammenhänge und Parameter der Strukturen genau beschreiben (siehe Abbildung 4.27).

SSTML stellt quasi einen Gegensatz zu AnatML dar. Wo AnatML durch Einfachheit hervorsteht, wird in SSTML eine enorme Vielfältigkeit und Komplexität angeboten. Daraus resultiert auch, dass SSTML bisher keine Verbreitung außerhalb der eigenen Simulationssoftware [69] gefunden hat. Eine Verwendung im vorliegenden Projekt wäre daher auch nicht von einem Nutzen, der den Aufwand der Implementierung rechtfertigen würde.

Kapitel 5

Physikalische Realisierung

Die physikalische Simulation der Instrumente und der anatomischen Umgebung bildet, neben der graphischen Darstellung und den technischen (Softwaredesign) Aspekten, den Hauptteil dieser Arbeit. Im folgenden Kapitel wird zuerst gezeigt, wie Instrumente (Katheter, Führungsdrähte, Occluder...) in der vorgestellten Arbeit simuliert werden.

Anschließend werden Mittel und Methoden vorgestellt, die Simulation zu beschleunigen und geeignete Datenstrukturen halbmanuell aus computertomographischen Aufnahmen zu gewinnen.

Im zweiten Teil wird die Simulation bewegter anatomischer Strukturen am Beispiel des Herzens vorgestellt. Zwei Methoden zur Bewegungssimulation werden vorgestellt und miteinander verglichen. Die erste basiert auf einer graphischen Approximation der Bewegungspfade. Die zweite berechnet in Echtzeit die physikalischen Auswirkungen der Herzdeformationen.

Im abschließenden dritten Teil wird die Simulation der Kontrastmittelausbreitung gezeigt. Es wird kurz dargelegt, warum aktuelle Methoden der Fluiddynamik (Gitterbasiertes Navier Stokes, Smooth Particle Hydrodynamics) derzeit noch nicht für die gezeigte Simulation geeignet sind und eine eigene Methode - der zelluläre Baum - wird vorgestellt.

5.1 Instrumentensimulation

5.1.1 Einführung

Eine ausgiebige Prüfung des bekannten Stand der Technik (siehe Kapitel 4.2) zeigt, dass derzeit keine der publizierten Methoden alle an ein Simulationssystem in der Kinderkardiologie gestellten Eigenschaften erfüllt. Diese, der Vollständigkeit halber hier noch einmal kurz wiederholt, sind:

1. **Echtzeitfähigkeit**
2. **Realitätsgrad**
3. **Flexibilität**
4. **Bedingungen an die Simulationsumgebung**

5. Zusammenspiel mehrerer Instrumente

In dieser Arbeit wird daher ein neuartiges System für die Simulation von Instrumenten entwickelt. Der grobe Ansatz baut auf das System von Alderliesten auf, es werden allerdings gänzlich andere Lösungsstrategien und eine andere geometrische Beschreibung verwendet, um den - von der Qualität der Lösungen her gesehen - sehr guten Ansatz um mehrere Größenordnungen performanter zu gestalten und somit an Echtzeitfähigkeit und Flexibilität zu gewinnen.

Zudem wird ein neuartiges Verfahren zur Simulation ineinander verschachtelter Instrumente hinzugefügt und die an die Simulationsumgebung gestellten Bedingungen so unrestrictiv wie möglich gestaltet.

Die Grundidee der vorgestellten Simulation basiert auf Energieminimierung. Dabei wirken in einem katheterförmigen Instrument¹ generell zwei Kräfte gegeneinander. Auf der einen Seite steht die Federrückstellkraft mit dem Drang ein Instrument in seiner vordefinierten Form zu halten. Diese aus dieser Kraft resultierende Energie wird im Folgenden als interne Energie (E_{int}) bezeichnet.

Auf der anderen Seite stehen die Kräfte, die durch Deformation der elastischen Gefäßwände induziert werden. Ohne Einwirkung einer äußeren Kraft kann davon ausgegangen werden, dass ein Instrument in seiner vordefinierten Form verbleibt. Liegt jedoch eine äußere Kraft an (wird der Katheter beispielsweise gegen eine Gefäßwand gedrückt), ist das Instrument gezwungen seine Form - unter Veränderung des internen Energiezustands - anzupassen. Die aus dem Anliegen einer äußeren Kraft resultierende Energie wird externe Energie (E_{ext}) genannt.

Hamilton's Prinzip der kleinsten Wirkung folgend, wird ein Instrument stets bestrebt sein, auf direktem Weg einen energetischen Extremzustand minimaler Energie einzunehmen. Die Form eines Instrumentes unter Einwirkung einer äußeren Kraft kann also durch Lösung der Gleichung

$$\min(E_{int} + E_{ext}) \tag{5.1}$$

ermittelt werden.

5.1.2 Geometrische Repräsentation

Um Gleichung 5.1 zu lösen, bedarf es zuvor einer numerisch geeigneten Beschreibung des Instruments. Da ein Instrument generell als unveränderlich in seiner Länge angenommen werden kann (die verwendeten Materialien sind nahezu nicht dehnbar), wurde hier eine neue Beschreibung über bewegliche Gelenke gewählt, die über fixe Abstände miteinander verbunden sind.

Die meisten Instrumentenformen ließen sich zwar generell über Splines mit einer geringeren (im Vergleich zu der Menge an benötigten Gelenken) Zahl an Kontrollpunkten repräsentieren, die Anforderung an die Unveränderlichkeit in der Länge bedingte dann aber zusätzliche Beschränkungen im Optimierungsverfahren, die nicht oder nur unter erheblichem rechnerischen Mehraufwand durchführbar wären.

¹Im Folgenden wird, sofern nicht anders angegeben, der Begriff Instrument stellvertretend für alle simulierten Instrumente (Katheter, Führungsdrähte...) verwendet.

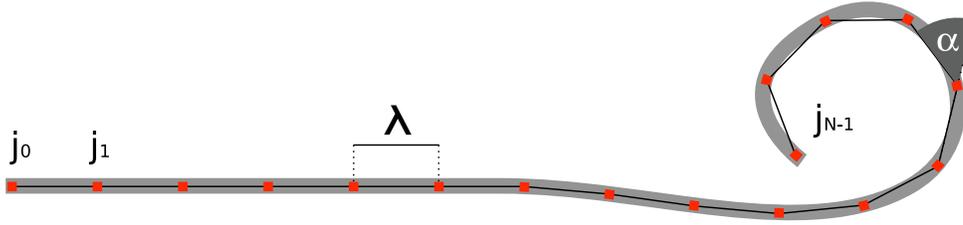


Abbildung 5.1: Katheterform mit Gelenken
Schematische Darstellung einer typischen Katheterform mit eingezeichneten Gelenken.

Eine beliebige Katheterform lässt sich theoretisch darstellen, wenn jedes Gelenk über drei Parameter beschrieben wird. Die Parameter sind λ (Abstand zum nächsten Gelenk), α (Winkeldifferenz zwischen zwei Gelenken in der Ebene) und β (Biegerichtung). Bild 5.1 zeigt eine (zweidimensionale) Darstellung eines Katheters. In Bild 5.2 ist zum besseren Verständnis die Veränderung der Winkel in einem Gelenk in einer dreidimensionalen Skizze dargestellt.

Jedes Gelenk j_i befindet sich im Ursprung eines eigenen (kartesischen) Koordinatensystems mit drei orthonormalen Achsen a_1^i , a_2^i und a_3^i . Die Position von j_i wird im Folgenden als x_i bezeichnet. Das nächste (distal gelegene) Gelenk j_{i+1} befindet sich auf der Oberfläche einer Kugel mit Mittelpunkt in x_i und Radius λ . Die Position des nächsten Gelenks x_{i+1} lässt sich dann über Formel 5.2 bestimmen.

$$x_{i+1} = x_i + \lambda * a_1^i \quad (5.2)$$

Auch wenn sich jede geometrische Form mit diesen drei Parametern pro Gelenk (λ , α & β) erzeugen lässt, ist ein weiterer Parameter nötig, um die interne energetische Konfiguration eines Instruments vollständig abzubilden. Optisch identische Formen können sowohl über Änderung der Biegerichtung als auch über Einführung einer Torsion in einem Gelenk erreicht werden. Dennoch kann die interne Energie beider Formen voneinander abweichen, weshalb ein vierter Parameter γ (Torsionswinkel) zur Beschreibung nötig wird.

Die Transformationen der Koordinatensysteme sind formell wie folgt definiert: Sei $R_v(\omega)$ die Rotationsmatrix, die eine Rotation um den Winkel ω mit einem beliebigen Einheitsvektor v als Drehachse beschreibt.

$$R_v(\omega) = \begin{pmatrix} \cos \omega + v_1^2 (1 - \cos \omega) & v_1 v_2 (1 - \cos \omega) - v_3 \sin \omega & v_1 v_3 (1 - \cos \omega) + v_2 \sin \omega \\ v_2 v_1 (1 - \cos \omega) + v_3 \sin \omega & \cos \omega + v_2^2 (1 - \cos \omega) & v_2 v_3 (1 - \cos \omega) - v_1 \sin \omega \\ v_3 v_1 (1 - \cos \omega) - v_2 \sin \omega & v_3 v_2 (1 - \cos \omega) + v_1 \sin \omega & \cos \omega + v_3^2 (1 - \cos \omega) \end{pmatrix} \quad (5.3)$$

Mit dieser Definition gilt für die Transformationen:

$$a_1^{(i+1)} = R_{(R_{a_1^i}(\beta_i) \cdot a_2^i)}(\alpha_i) \cdot a_1^i \quad (5.4)$$

$$a_2^{(i+1)} = R_{a_1^{(i+1)}}(\gamma_i) \cdot R_{R_{a_1^i}(\beta_i) \cdot a_2^i}(\alpha_i) \cdot a_2^i \quad (5.5)$$

$$a_3^{(i+1)} = a_1^{(i+1)} \otimes a_1^{(i+2)} \quad (5.6)$$

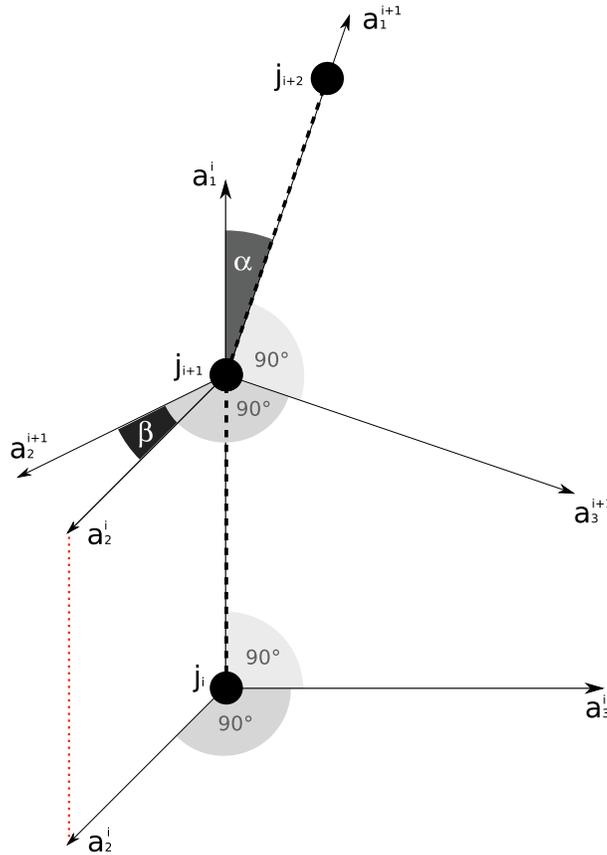


Abbildung 5.2: Transformation des Koordinatensystems am Kathetergelenk. Die beiden Richtungsvektoren a_2^i und a_2^{i+1} stehen beide senkrecht auf dem Verbindungsvektor zwischen j_i und j_{i+1} .

5.1.3 Interne Energie

Mit dieser diskreten Darstellung des Instruments berechnet sich die aus der Veränderung von α und β resultierende Winkelabweichung θ_i von der vordefinierten Form in einem einzelnen Gelenk j_i mit vordefinierten Biegungswinkeln $\tilde{\alpha}$ und $\tilde{\beta}$ durch:

$$\theta_i = \arccos(\sin \tilde{\alpha} * \sin \alpha * \sin \tilde{\beta} * \sin \beta + \cos \tilde{\beta} * \cos \beta * \sin \tilde{\alpha} * \sin \alpha + \cos \tilde{\alpha} * \cos \alpha) \quad (5.7)$$

Das Gelenk spezifizierende Indizes an den Winkeln wurden aufgrund der Übersichtlichkeit verworfen - alle Winkel beziehen sich auf das entsprechende zu berechnende Gelenk. Damit lässt sich die interne Energie E_{int}^i in dem Gelenk, auf Grundlage des Hookeschen Gesetzes, wie folgt beschreiben:

$$E_{int}^i = \left(\left(\frac{1}{2} * \theta_i^2 * S_i \right) + \left(\frac{1}{2} * \gamma_i^2 * T_i \right) \right) * \frac{1}{\lambda} \quad (5.8)$$

S_i bezeichnet hier die Federkonstante - ein Parameter für die Steifigkeit des Instruments - und T_i die Torsionskonstante im Gelenk j_i . Da Instrumente nicht über eine

vordefinierte Torsion verfügen, wurde auf Angabe eines definierten Torsionswinkels $\tilde{\gamma}$ verzichtet.

Somit lässt sich die interne Energie eines Instrumentes mit N Gelenken unter Verwendung von 5.7 und 5.8 als harmonische Approximation darstellen:

$$E_{int} = \sum_{i=0}^{N-1} E_{int}^i \quad (5.9)$$

5.1.4 Externe Energie

Die externe Energie in einem Gelenk ist im Fall einer Kathetersimulation durch die Rückstellkraft des von dem Gelenk deformierten Gewebes (Gefäße, Herzmuskel...) induziert und kann, gemäß Hookescher Approximation geschrieben werden als:

$$E_{ext}^i = \frac{1}{2} * r^2 * U * \lambda \quad (5.10)$$

wobei (siehe Abbildung 5.3) r dem Abstand der Deformation entspricht, und U eine Konstante für die Elastizität des deformierten Gewebes darstellt.

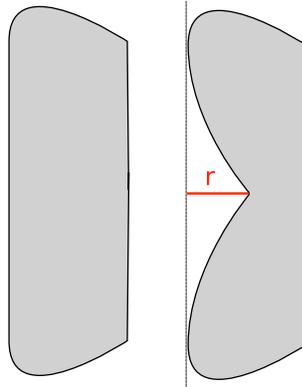


Abbildung 5.3: Darstellung der durch Deformation induzierten externen Energie.

Der Wert von r ist abhängig von der Position des Gelenkes $j_i(x_i)$ und dem Durchmesser² d_i des Gelenks. Da der Durchmesser der Gelenke oft (im Vergleich zur Länge des Gelenkabschnitts) kleine Werte annimmt, kann das Gelenk näherungsweise durch eine Kugel approximiert werden, was die Berechnung vereinfacht. Ist der Durchmesser deutlich größer als die Länge, muss die Berechnung der Deformation durch Abtastung mehrerer Punkte auf der Oberfläche des Instruments erfolgen (etwa bei aufgespannten Retentionsscheiben).

Ausgehend von der Annahme, dass es M Strukturen gibt, auf die das Gelenk j_i Einfluss nehmen kann, wird die externe Energie des Instruments unter Berufung auf die harmonische Approximation beschrieben als:

²Da Instrumente in der Kardiologie üblicherweise durch ihren Durchmesser und nicht durch ihren Radius beschrieben werden, passt sich diese Arbeit dem kardiologischen Standard weitgehend an.

$$E_{ext} = \sum_{i=0}^{N-1} \sum_{k=0}^{M-1} \begin{cases} (\frac{1}{2} * F_k(j_i)^2 * U_k * \lambda), & \text{if } F_k(j_i) < 0 \\ 0, & \text{if } F_k(j_i) \geq 0 \end{cases} \quad (5.11)$$

Die Funktion F_k sei als vorzeichenbehaftete Abstandsfunktion so definiert, dass der Funktionswert an Stelle j_i dem Abstand r zwischen der Struktur k und dem Gelenk j_i entspricht (siehe Abbildung 5.3).

Die Gesamtenergie des Instruments errechnet sich somit durch Summation aus 5.9 und 5.11:

$$E = E_{int} + E_{ext} \quad (5.12)$$

Es bleibt zu erwähnen, dass sowohl Gleichung 5.9 als auch Gleichung 5.11 aufgrund der harmonischen Approximation nur bei Verwendung kleiner Segmentlängen ($\lambda \leq 5\text{mm}$) sinnvolle Resultate liefert.

5.1.5 Numerische Approximation des Gradienten

Auch wenn sich der externe Teil des Gradienten teilweise auf analytische Weise approximieren lässt (siehe [30]), sind die benötigten Kalkulationen deutlich zu aufwendig, um eine Echtzeitanwendung auf aktuellen Rechnergenerationen durchführen zu können (In [32] werden Zeiten von teilweise $> 20s$ pro Berechnungsschritt angegeben, wobei die geometrische Repräsentation aus [30] von der hier vorgestellten abweicht).

Da in dieser Arbeit die physikalisch korrekte Berechnung eines Instruments in Echtzeit im Vordergrund stehen soll, wird die Gleichung stattdessen auf eine neue Weise numerisch effizient gelöst. Der externe Teil des Gradienten wird dabei geschickt numerisch approximiert, während nur der trivial zu berechnende interne Teil in der Regel analytisch bestimmt wird.

Für den internen Gradienten eines Gelenks gilt (Aus Gründen der Übersichtlichkeit werden keine gelenkspezifisierenden Indizes angegeben):

$$\nabla E_{int}^i(\alpha, \beta, \gamma) = \begin{pmatrix} \frac{-\psi_1}{1-\psi_2^2} \cdot \theta \cdot S \cdot \frac{1}{\lambda} \\ \frac{-\psi_3}{1-\psi_4^2} \cdot \theta \cdot S \cdot \frac{1}{\lambda} \\ (\gamma - \tilde{\gamma}) \cdot T \end{pmatrix} \quad (5.13)$$

Wobei ψ_{1-4} die folgenden Hilfwerte bezeichnen

$$\psi_1 = (\sin \tilde{\alpha} \cos \alpha \cos \tilde{\beta} \cos \beta) + (\sin \tilde{\alpha} \cos \alpha \sin \tilde{\beta} \sin \beta) - (\cos \tilde{\alpha} \sin \alpha) \quad (5.14)$$

$$\psi_2 = (\sin \tilde{\alpha} \sin \alpha \cos \tilde{\beta} \cos \beta) + (\sin \tilde{\alpha} \sin \alpha \sin \tilde{\beta} \sin \beta) + (\cos \tilde{\alpha} \cos \alpha) \quad (5.15)$$

$$\psi_3 = (\sin \tilde{\alpha} \sin \alpha \sin \tilde{\beta} \cos \beta) - (\sin \tilde{\alpha} \sin \alpha \cos \tilde{\beta} \sin \beta) \quad (5.16)$$

$$\psi_4 = (\sin \tilde{\alpha} \sin \alpha \cos \tilde{\beta} \cos \beta) + (\sin \tilde{\alpha} \sin \alpha \sin \tilde{\beta} \sin \beta) + (\cos \tilde{\alpha} \cos \alpha) \quad (5.17)$$

Zur Beschleunigung der Rotationsberechnungen werden die Rotationsmatrizen - speziell bei aufeinanderfolgenden Rotationen - intern durch Quaternionen repräsentiert.

Die eigentliche numerische Berechnung des Gradienten geschieht nach dem bekannten Verfahren des zentralen Differenzenquotienten, zieht jedoch Nutzen daraus, dass sich die Orte der nachfolgenden (distalen) Gelenke bei Veränderung der Winkel in einem vorhergehenden Gelenk direkt durch Kombination von Rotation und Translation berechnen lassen. Proximal gelegene Gelenke haben keinen Einfluss auf den externen Gradienten distaler Gelenke, da sich ihr Ort (und damit die auf sie wirkende externe Kraft) nicht verändert.

Entscheidend für den externen Beitrag zu einer Energie ist der Ort eines Gelenks (bzw. im Gradienten alle distal auf ein Gelenk folgende Gelenke). Dieser lässt sich zwar iterativ aus den bereits dargestellten Formeln berechnen, kann aber auch effizient durch eine Rotation um das distale Gelenk einer Bewegung berechnet werden.

Sei $R_v^u(\omega)$ eine (homogene 4×4) Matrix, die bei Multiplikation mit einem (homogenen 4×1) Vektor einer Rotation um ω Grad um die Rotationsachse v mit Drehzentrum in u entspricht.

Entsprechend gilt für den externen Gradienten eines Gelenks (x , der Ort eines Gelenks sei in homogenen 4×1 Koordinaten gegeben):

$$\nabla E_{ext}^i(\alpha, \beta, \gamma) = \begin{pmatrix} \sum_{k=i+1}^{N-1} (F(R_{\alpha_1}^{x_i} (k-1) (\beta) (h) \cdot x(j_k)) + F(R_{\alpha_1}^{x_i} (k-1) (\beta) (-h) \cdot x(j_k)))/(2h) \\ \sum_{k=i+1}^{N-1} (F(R_{\alpha_1}^{x_i} (-h) \cdot R_{\alpha_1}^{x_i} (i-1) (h) \cdot x(j_k)) + F(R_{\alpha_1}^{x_i} (h) \cdot R_{\alpha_1}^{x_i} (i-1) (-h) \cdot x(j_k)))/(2h) \\ \sum_{k=i+1}^{N-1} (F(R_{\alpha_1}^{x_i} (h) \cdot x(j_k)) + F(R_{\alpha_1}^{x_i} (-h) \cdot x(j_k)))/(2h) \end{pmatrix} \quad (5.18)$$

Der vollständige Gradient eines Gelenks approximiert sich abschließend nach der Summationsregel unter Anwendung von 5.13 und 5.18 aus:

$$\nabla E^i(\alpha, \beta, \gamma) = E_{int}^i(\alpha, \beta, \gamma) + E_{ext}^i(\alpha, \beta, \gamma) \quad (5.19)$$

5.1.6 Verschachtelte Instrumente

Die bekannten Verfahren zur Instrumentensimulation gehen meist davon aus, dass Drähte die Form des Katheters nicht, oder nur streckend durch eine Veränderung der Steifigkeiten beeinflussen.

Instrumente werden in der Kardiologie jedoch in verschiedenen komplexeren Kombinationen verwendet. Führungsdrähte strecken *und* biegen Katheter (siehe Abbildung 5.4), vorgeformte Katheter werden dazu genutzt, um Spulen oder Occluder in die richtige Richtung zu lenken. Die gegenseitige Beeinflussung von Instrumenten ist also ein in der Simulation nicht zu vernachlässigender Faktor.

In der vorliegenden Arbeit werden mehrere ineinander verschachtelte Instrumente als eine neuartige Form eines Instrumentenstrangs betrachtet, der vollständig in den Optimierungsprozess übernommen wird. Verschachtelte Instrumente werden also nicht eigenständig berechnet, sondern in einem Vorverarbeitungsschritt effizient zu einem virtuellen Instrument (im Folgenden auch Proxy genannt) kombiniert.

Zu diesem Zweck werden alle beschreibenden Parameter des Instruments in Paketen zusammengefasst (siehe Abbildung 5.5). In der Abbildung wird ein Katheter durch ein 60mm langes, gerades Segment (S_1) und zwei Kurvenssegmente mit verschiedenen Radien und Winkeln (C_1 und C_2) repräsentiert. Die Beschreibung über Kreissegmente wurde aus der kardiologischen Praxis übernommen.

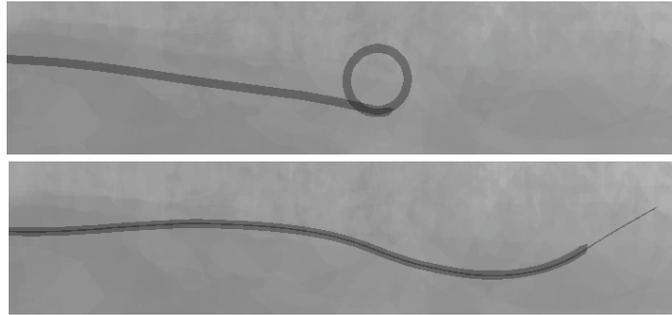


Abbildung 5.4: Darstellung verschachtelter Instrumente. Ein steifer Führungsdraht wird durch einen Katheter geschoben und bewirkt eine Deformation.

Ähnliche Pakete werden für die Beschreibung der Torsions- und Biegesteifigkeiten und der Instrumentendurchmesser verwendet. Überdecken sich zwei ineinander verschachtelte Instrumente, werden die Pakete so aufgesplittet, dass die Anfänge und Enden der Pakete auf identische Punkte fallen (siehe schematische Abbildung 5.6). Aus den gesplitteten Paketen können dann kombinierte Pakete berechnet werden.

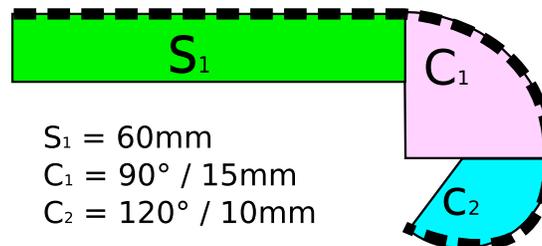


Abbildung 5.5: Einteilung eines Instruments in Formpakete. S_1 beschreibt ein gerades, C_1 und C_2 gebogene Segmente verschiedener Krümmung.

Die Werte der resultierenden Pakete werden berechnet, indem die Werte der Originalpakete zur Beginn- bzw. Endposition des neuen Paketes ermittelt und, ggf. unter Zuhilfenahme weiterer Pakete (Formen benötigen Informationen über die Steifigkeit), nach der untenstehenden Tabelle (5.1) verknüpft werden. In der Tabelle bezeichnet I_i das i -te von N Instrumenten, $L(I)$ die Länge des Pakets $S(I)$ die Steifigkeit, $D(I)$ den Durchmesser, $T(I)$ den Torsionsparameter. Bei Formparametern bezeichnet $W(I)$ den Winkel des Kreises und $R(I)$ den zugehörigen Radius.

Weitere spezialisierte Pakete können für die Beschreibung komplexerer oder multifunktionaler Instrumente (Occluder, Stent, Filterwire) verwendet werden. Der interne, beschreibende Aufbau der Instrumente lässt prinzipiell eine beliebige Anzahl Pakete zu.

Als Beispiel sei hier ein sich öffnender Occluderschirm angeführt, der positionabhängig seine Form ändert und daher den Radius seiner Retentionsscheibe über Linearkombinationen von Splines mit variablen Kontrollpunkten bestimmt. Die Kontroll-

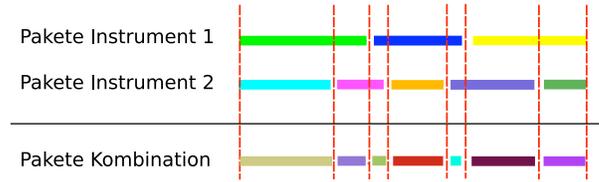


Abbildung 5.6: Schematische Darstellung der Aufsplittung der Instrumentenpakete.

PARAMETER	REGEL
$D(I_r)$	$\max(D(I_0), \dots, D(I_{N-1}))$
$S(I_r)$	$\sum_{i=0}^{N-1} S(I_i)$
$T(I_r)$	$\sum_{i=0}^{N-1} T(I_i)$
$\begin{pmatrix} W(I_r) \\ R(I_r) \end{pmatrix}$	$\begin{pmatrix} \frac{L(I_r)}{R(I_r)*0.5} \\ \frac{\sum_{i=0}^{N-1} R(I_i)*S(I_i)}{\sum_{i=0}^{N-1} S(I_i)} \end{pmatrix}$

Tabelle 5.1: Regeln für die Berechnung von Instrumentenparametern

punkte sind Teil einer weiteren Paketklasse.

5.1.7 Erste Anpassung des Funktionsraums

Würde man versuchen, auf der Zielfunktion stets das globale Optimum zu finden, wäre es prinzipiell nicht auszuschließen, dass dabei Energiebarrieren (wie sie beispielsweise zwischen parallel laufenden Gefäßen entstehen) überwunden würden, die in der Natur eine nicht durchdringbare Grenze darstellten. Ein globales Optimum könnte zudem, bedingt durch die Diskretisierung des Instrumentes, möglicherweise Gefäßwände durchstoßen und eine auf diese Weise physikalisch nicht motivierte Lösung repräsentieren.

Vergleicht man das Verfahren mit natürlichen Prozessen, so ist zu sehen, dass die Natur in der Regel nach ideal geeigneten lokalen Tiefpunkten sucht. Eine Kugel, die in einem hügeligen Gebiet einen Abhang hinunterrollt, kann nur dann eine lokale Tiefstelle (Kessel) wieder verlassen, wenn ihre verbleibende kinetische Energie dazu ausreicht, die nächste Barriere zu überwinden.

Das Problem soll im Kontext der Instrumentensimulation durch Abbildung 5.7 deutlich gemacht werden, wo der α und der β Parameter eines Gelenks im Bereich $[-\pi, \pi]$ verändert wurde, und die resultierende Gesamtenergie des Instruments auf ein Diagramm aufgetragen wurde³. Deutlich zu sehen sind die verteilten lokalen Optima, und Energiebarrieren, die beim Übertreten der Gefäßgrenzen auftreten.

Die einfachste Variante, solche "unrealistischen" Optima zu verhindern, wäre es, die Schrittweite des Lösungsverfahrens stark zu begrenzen und damit das Verweilen

³Im Weiteren wird der zu lösende Funktionsraum einer kurzen Untersuchung unterzogen. In späteren Abschnitten folgt eine ausführlichere Betrachtung, die sich aber bereits auf den in dieser Sektion angepassten Funktionsraum beziehen soll.

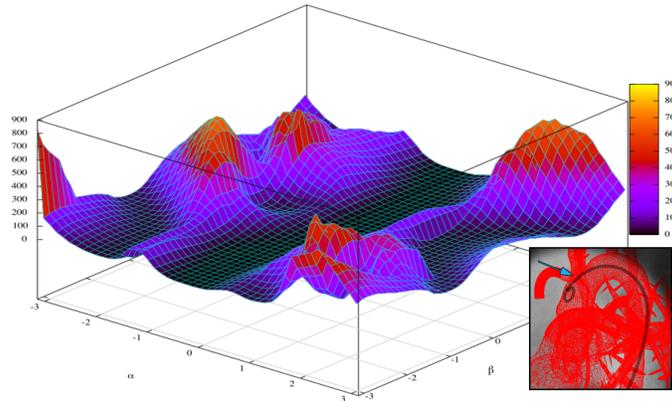


Abbildung 5.7: Entstehende Energiebarriere bei der Überquerung mehrerer Gefäße. Mehrere Täler deuten auf verteilte Optima hin, die entsprechend eingegrenzt werden müssen.

an lokalen Optimalstellen zu forcieren. Dieses (eher heuristische) Verfahren bringt aber weitere Probleme mit sich. Die Anzahl der Optimierungsschritte - und damit die benötigte Rechenzeit - erhöht sich, bei zu geringer Schrittweite bleibt das Verfahren möglicherweise bereits in sehr kleinen lokalen Optima hängen.

Eine weitere, oft für derartige Probleme genutzte Methode wäre die Einführung einer schnellen Kollisionsdetektion und adäquater Strafterme (Penalties) in der Zielfunktion [75]. Auch hier gilt aber, dass die Konsequenzen für die Rechenzeit groß wären - eine Auflösung der Kollisionsdetektion über einen Vorzeichenwechseltest auf den (ohnehin vorhandenen) Distanzfeldern ist nicht praktikabel, da Strukturen teilweise im Submillimeterabstand zueinander verlaufen und damit für Gelenke mit fixem Abstand nicht disjunkt erschienen. Zudem würde durch den Strafterm die eigentliche Zielfunktion auf nicht physikalischer Basis verändert - falsche bzw. physikalisch unmotivierte Ergebnisse wären die Folge.

Aus diesen Gründen wurde in dieser Arbeit ein anderer Weg gewählt, um ein Springen zwischen den Strukturen zu vermeiden: Anstatt den kompletten Gefäßbaum als eine große anatomische Struktur zu betrachten, werden die Gefäße (oder Gewebeteile) in Segmente aufgesplittet und als Knoten in einem nicht gerichteten (ggf. zyklischen) Graphen angesehen (siehe Abbildung 5.8).

Auf diese Weise ist es möglich, jedes Gelenk einem Knoten des Graphen (bzw. einer Gefäßstruktur) zuzuordnen und den Suchraum (Strukturen mit denen das Gelenk kollidieren kann) auf dieses Gefäß und seine direkten Nachbarn zu beschränken. So wird ein Springen zu anderen, nicht direkt benachbarten Gefäßen effektiv verhindert. Zugleich werden bei geschickter Wahl der Segmentabstände auch andere Probleme wie ein Umknicken des Instruments effektiv verhindert und der Funktionsraum, abgesehen von kleinen lokalen Tiefstellen, verbessert.

Die Zuordnung zu den Segmenten erfolgt über die Anwendung des Regelsatzes 5.1.7 in Folge auf den Optimierungsprozess. Für die Tabelle sei N_i der Knoten bzw. die Struktur, die mit dem Gelenk j_i verknüpft ist. $m(N)$ bezeichnet die Menge der Knoten, die direkt mit N verlinkt sind (einschließlich N). Die Funktion $\xi(S)$ schließlich

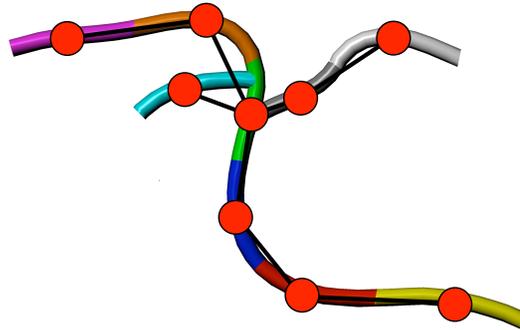


Abbildung 5.8: Graph aus Gefäßstrukturen

PARAMETER	REGEL
$j_i = \text{Distales Gelenk}$	$\xi(m(N_i) \cup m(N_{i-1}))$
$j_i = \text{Mittleres Gelenk}$	$\xi(m(N_i) \cup m(N_i + 1) \cup m(N_{i-1}))$
$j_i = \text{Proximales Gelenk}$	Gelenk wird nicht bewegt - Struktur unverändert.

Tabelle 5.2: Zuweisungsregeln für Knoten

bezeichnet den Knoten N aus einer Summe von Knoten S , dessen vorzeichenbehaftete Distanzfunktion F_k den größten Wert liefert.

Der Plot, der entsteht, wenn die Gelenke vor der Messung auf diese Weise fixiert werden, ist in Abbildung 5.9 dargestellt. Die lokalen Optima sind größtenteils verschwunden und es zeigt sich eine glatte, augenscheinlich konvexe Funktion⁴.

5.1.8 Lösung des Optimierungsproblems

Mit 5.12 und 5.19 kann Gleichung 5.1 unter Verwendung der bekannten numerischen, gradientengestützten Optimierungsverfahren [162] prinzipiell gelöst werden. Dabei spielt die Wahl des konkreten Verfahrens eine große Rolle für die umsetzbare Geschwindigkeit.

Charakter der Zielfunktion

Um eine Entscheidungsgrundlage für ein geeignetes Verfahren aufzustellen, ist es nötig, die zu optimierende Zielfunktion 5.12 im Vorfeld weiter zu untersuchen. Die Dimensionalität D des Problems errechnet sich aus:

$$D = k \cdot N \quad (5.20)$$

⁴Die Gesamtfunktion in N Dimensionen ist natürlich weiterhin nicht vollständig konvex, wie im Weiteren gezeigt werden wird

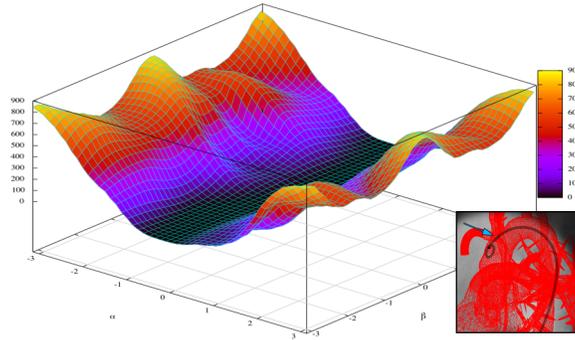


Abbildung 5.9: Entstehende Energiebarriere bei der Überquerung mehrerer Gefäße mit Fixierung auf Baumstrukturen. Die mehrfachen Täler sind verschwunden und es zeigt sich ein glatter Funktionenverlauf.

wobei N die Anzahl der freien Gelenke bezeichnet, und k der Anzahl der freien Parameter (in der Regel $\alpha, \beta, \gamma \Rightarrow 3$) pro Gelenk entspricht.

Da sich die Untersuchung einer solchen Funktion oft nicht intuitiv gestaltet, wurde für die folgenden Diagramme die Dimensionalität (wie bereits in Abschnitt 5.1.7) abermals auf 2 begrenzt. Tabelle 5.1.8 zeigt die Energieverteilung bei Variation der α und β Parameter eines einzelnen Gelenks im Bereich $[-\pi, \pi]$, während in Abbildung 5.1.8 die α Parameter zweier aufeinanderfolgender Gelenke im selben Bereich als Funktionswerte dienen.

Variables Gelenk	Interne Energie	Externe Energie	Gesamtenergie

Tabelle 5.3: Darstellung der Energieaufteilung bei Veränderung der α/β Werte in einem Gelenk.

Es zeigt sich, dass schon bei solch einer drastischen Einschränkung des Suchraums keine unimodale⁵ Funktion entsteht. Bei Erweiterung auf die volle Dimensionalität D zeigt sich in numerischer Auswertung eine multimodale (nicht konvexe), stetige, nach unten beschränkte (die Gesamtenergie kann nicht < 0 sein) und periodische (mit 2π in α und β) Funktion.

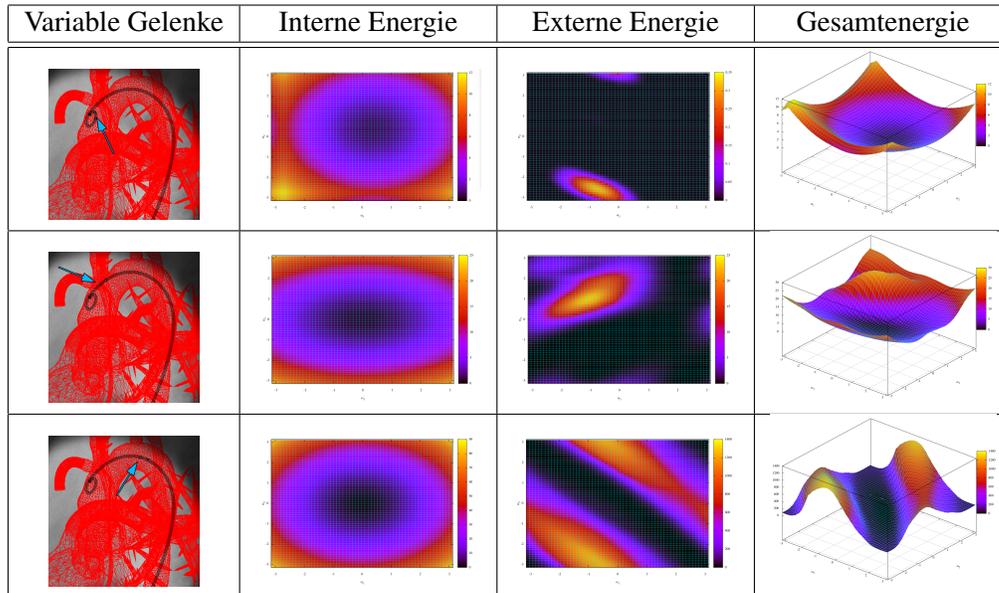


Tabelle 5.4: Darstellung der Energieaufteilung bei Veränderung der α Werte zweier aufeinanderfolgender Gelenke.

Um den multimodalen Charakter weiter zu demonstrieren, zeigt Tabelle 5.10 die Energieverteilung zweier direkt aufeinanderfolgender, distaler Gelenke mit sehr geringer Steifigkeit, deren α Werte im Bereich $[-\pi, \pi]$ verändert wurden. Es zeigt sich eine stark multimodale Funktion, deren lokale Tiefpunkte symmetrisch um einen globalen Tiefpunkt verteilt sind.

Ein anschaulicher Grund für die Entstehung einer solchen Energieverteilung ist die Redundanz der Winkeleinstellung aufeinanderfolgender Gelenke. Knicken zwei Gelenke in entgegengesetzte Richtungen, kann sich die externe Energie, die entsteht, wenn nur ein Gelenk knickte, wieder aufheben (siehe Abbildung 5.11).

Das Lösungsverfahren muss also in der Lage sein, schnell ein möglichst globales Optimum in Bezug auf den in Abschnitt 5.1.7 angepassten Funktionsraum zu finden (kleine lokale Minima müssen dabei so gut wie möglich übergangen werden).

Kosten der Zielfunktion

Die Kosten für die Auswertung der Zielfunktion sind bei N Gelenken im Bereich $O(N)$, eine Gradientenauswertung kann bei direkter Auswertung nach dem oben angegebenen Verfahren in $O(\frac{N^2}{2}) \Rightarrow O(N^2)$ erfolgen.

⁵Funktion ohne lokale Tiefpunkte

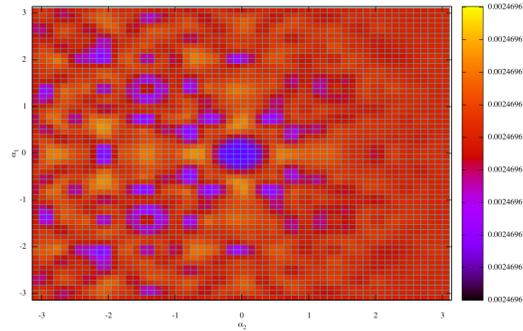


Abbildung 5.10: Repräsentative Darstellung der Multimodalität der Zielfunktion

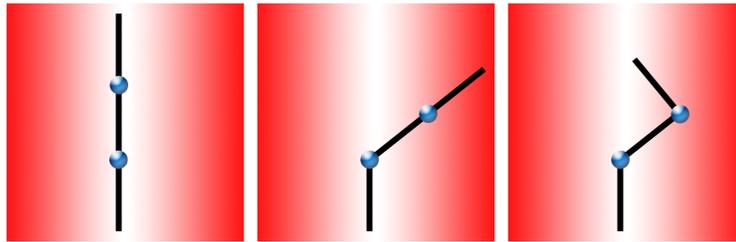


Abbildung 5.11: Zwei aufeinanderfolgende Gelenke beeinflussen ihren externen Energieeinfluss durch gegenläufig redundante Ausrichtung

Da die Auswertung der Gradienten durch lokale Linearisierung (siehe Abschnitt 5.1.9) weiter beschleunigt (allerdings nicht in der Komplexität verändert) werden kann, bietet es sich an, für die Lösung des Optimierungsproblems ein gradientengestütztes Verfahren zu verwenden.

Im Rahmen dieser Arbeit wurden mehrere numerische Lösungsverfahren untersucht, implementiert und miteinander verglichen. Verfahren, die ohne Gradientenauswertung funktionieren (Nelder Mead Simplex, Dynamic Hill Climbing, Powell) wurden zwar implementiert - werden aber aufgrund der vergleichsweise langsamen Performance an dieser Stelle nicht weiter untersucht. Betrachtet werden sollen im Folgenden daher nur triviale Gradientenverfahren (Steepest Decent), konjugierte Gradienten (mit und ohne Vorkonditionierung) und Quasi Newton Methoden.

Steepest Decent

Als einfachstes, primitives Verfahren kann das Verfahren des steilsten Abstiegs herangezogen werden. Mit einer definierten Schrittweite h , der Energiefunktion $f(x)$ und dem zugehörigen Gradienten $\nabla f(x)$, wird ein lokales Optimum durch iterative Ausführung von

$$x_{i+1} = x_i - h * \nabla f(x) \quad (5.21)$$

gefunden. Wird für h ein hinreichend kleiner Wert gewählt und handelt es sich bei $f(x)$ um eine konvexe Funktion, wird ein Punkt nahe des Minimums auf diese Weise sicher gefunden. Allerdings ist das Verfahren, bedingt durch die meist recht kleine benötigte Schrittweite h , sehr langsam und ungeeignet für die meisten komplexeren Funktionen.

Da die Zielfunktion wie bereits gezeigt nicht konvex ist, kann nicht davon ausgegangen werden, dass das Verfahren zu einem stabilen Ergebnis führt. Das Absteigen in lokale Minima ist bei direkter Auswertung der Gradienten wahrscheinlich, die Performance nicht ausreichend.

Nonlineares Konjugierte Gradienten Verfahren

Ein häufig verwendetes Verfahren zur Optimierung stellen konjugierte Gradienten dar. Das Verfahren nutzt die Eigenschaft linearer Funktionen, dass die Optimierung von

$$f(x) := b \cdot x - \frac{1}{2}xAx \quad (5.22)$$

äquivalent zum Lösen der Gleichung $Ax = b$ ist. Die Lösung wird berechnet, indem sukzessive in Richtung konjugierter (sich gegenseitig nicht beeinflussender) Gradienten optimiert wird. Bei *linearen* Funktionen in N Dimensionen, kann mit dem Verfahren das Erreichen eines Optimums innerhalb von N Schritten garantiert werden, da die jeweils i -te Iteration sicherstellt, den (lokalen) Tiefpunkt einer Funktion in dem durch i konjugierte Gradienten $\nabla f(x)^{(1..i)}$ aufgespannten Vektorraum V_k zu finden.

$$V_k := \text{span} \left(\{ \nabla f(x)^{(1)}, \dots, \nabla f(x)^{(i)} \} \right) \quad (5.23)$$

Handelt es sich bei $f(x)$ um eine nicht lineare Funktion (wie im gegebenen Fall), wird die Funktion in jedem Iterationsschritt durch eine Taylor Approximation zweiten Grades angenähert. In der oben dargestellten Formel (5.22) stellt $b \cdot x$ dabei die Komponente erster und $\frac{1}{2}xAx$ die Komponente zweiter Ordnung der Taylor Approximation dar.

Bei den Versuchen im Rahmen dieser Arbeit kamen sowohl das Verfahren von Polak & Ribiere als auch das Verfahren von Fletcher & Reeves zum Einsatz. Beide Verfahren haben gemein, dass die normalerweise für konjugierte Gradienten Verfahren benötigte Hesse Matrix (A in der oben gewählten Form) nicht berechnet wird, sondern die konjugierten Gradienten direkt unter Zuhilfenahme einer Gradientenauswertung auf der *nicht approximierten* Funktion $f(x)$ errechnet werden [162], was beweisbar⁶ zu einem identischen Ergebnis führt.

Polak-Ribiere gilt in der Literatur allgemein als das schnellere Verfahren, kann allerdings in seiner ursprünglichen Form zu einer "nicht Konvergenz" der Methode führen. Insofern wird in dieser Arbeit der modifizierte Ansatz aus [182] verwendet, der in der Lage ist, diesen Fall abzufangen.

⁶Der Beweis findet sich in [162], spielt aber im Kontext dieser Arbeit keine Rolle, weshalb er hier nicht vollständig zitiert werden soll

Fletcher-Reeves wurde zu Vergleichszwecken implementiert - jedoch waren im vorliegenden Fall keinen nennenswerten Unterschiede feststellbar, weshalb im weiteren oBdA von der Verwendung des Verfahren von Polak & Ribiere ausgegangen wird.

Im Vorfeld zu bemerken ist, dass die maximale Anzahl der Iterationen bei der nonlinearen KG Methode im Gegensatz zum linearen Fall nicht beschränkt ist (siehe auch [182]). Daher gilt: Ist nach N Iterationsschritten kein hinreichend genaues Optimum gefunden (die konjugierten Gradienten spannen zu diesem Zeitpunkt bereits den gesamten Vektorraum der Funktion auf und es können keine weiteren Gradienten gefunden werden, die den Konjugationsbedingungen genügen), wird das Verfahren in der Regel vom bisher erreichten Punkt aus erneut gestartet.

Von großer Bedeutung für die Performanz des Verfahren ist auch die Wahl des Verfahrens zur Optimierung in Gradientenrichtung (im Folgenden auch als Liniensuche bezeichnet). Dabei handelt es sich um eindimensionale Minimierungsverfahren, die versuchen den Tiefpunkt in Richtung des Gradienten möglichst exakt zu bestimmen.

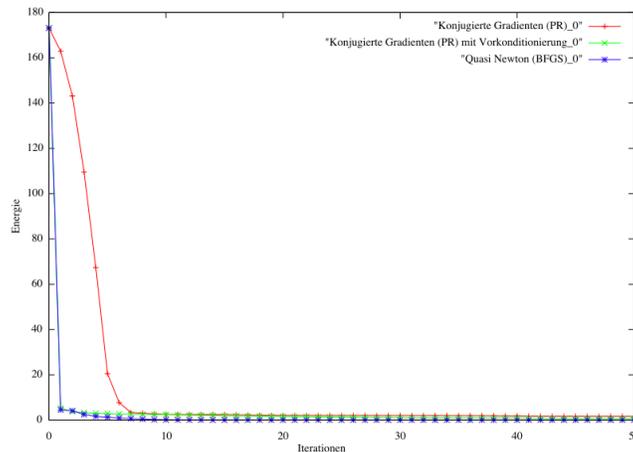


Abbildung 5.12: Verlauf der verschiedenen Lösungsverfahren im Verhältnis zu den Iterationen

Die Abbildungen 5.12 und 5.13 zeigen einen durchschnittlichen Lösungsverlauf (150 freie Variablen), der bei Verwendung des unmodifizierten konjugierte Gradientenverfahren nach Polak Ribiere mit einer exakten Liniensuche nach Brent erreicht wird im Vergleich zu den beiden noch folgenden Verfahren (Konjugierte Gradienten mit Prädiktionierung und Quasi Newton).

Betrachtet man das herkömmliche konjugierte Gradienten Verfahren, so fällt im Vergleich zu den anderen Verfahren die anfänglich langsame Konvergenz sowie das final erreichte, nicht ideale⁷ Optimum auf. Diese Verhalten soll im Folgenden und in Abgrenzung zu den anderen Lösungsverfahren näher untersucht und erklärt werden.

⁷Der Begriff des nicht idealen Optimums bezeichne hier eine lokale Tiefstelle mit Funktionswert f , in deren Umgebung sich weitere Tiefstellen mit einem Funktionswert g befinden wobei $g < f$.

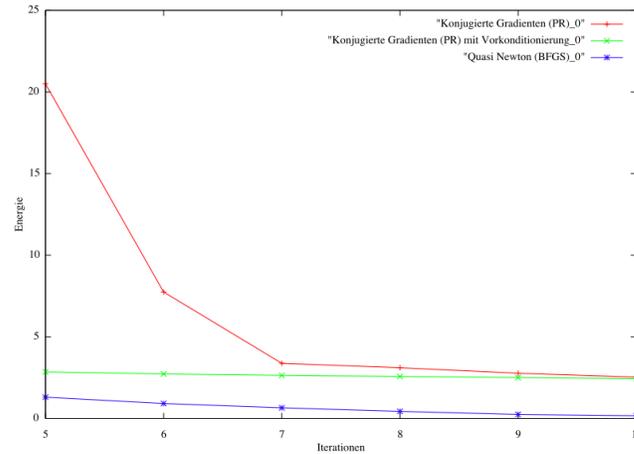


Abbildung 5.13: Verlauf der verschiedenen Lösungsverfahren im Verhältnis zu den Iterationen (vergrößerter Ausschnitt aus Abbildung 5.12)

Nonlineares Konjugierte Gradienten Verfahren mit Präkonditionierung

Ein häufig vorkommendes Problem beim KG Verfahren stellt die Kondition der (implizit verwendeten) Hesse Matrix dar. Ist die Matrix schlecht konditioniert, bedeutet das anschaulich ein stark ungleiches Verhältnis der Steigungen in den verschiedenen Gradientenrichtungen. Das wiederum führt aufgrund numerischer Probleme ggf. zu langen Iterationszeiten und schlechter Konvergenz und sollte daher möglichst vermieden werden.

Um schlecht konditionierte Matrizen zu verbessern, kann eine Präkonditionierungsmatrix P , die die Anforderungen $P^{-1} \approx A^{-1}$ erfüllt verwendet werden und Alternativ die Gleichung $P^{-1}Ax = P^{-1}b$ gelöst werden.

Problematisch stellt sich allerdings die Berechnung der Matrix P dar. Eine echte Berechnung und Invertierung der Hesse Matrix wäre zu aufwendig und würde das gegebene Verfahren eher verlangsamen als tatsächlich beschleunigen. Daher wurde in dieser Arbeit als einfachster Präkonditionierer eine Jacobi Präkonditionierungsmatrix J gewählt.

$$J_{ij} = \begin{cases} A_{ii}, & \text{wenn } i = j \\ 0, & \text{sonst} \end{cases} \quad (5.24)$$

Da es sich bei J um eine Diagonalmatrix handelt, fällt die Invertierung entsprechend leicht. Die Präkonditionierung selbst wird in jedem Schritt durchgeführt. Zwar könnte die Berechnung der Konditionszahl als Maß für die Notwendigkeit einer Präkonditionierung herangezogen werden, die dafür notwendige Berechnung des Verhältnisses zwischen dem jeweils größten und kleinsten Singulärwert der Hesse Matrix A ließe die Rechenzeiten allerdings in einem höheren Maß anwachsen als das bei einer wiederholten Präkonditionierung der Fall ist.

Da die Elemente von J jedoch numerisch berechnet werden müssen (beim nicht linearen konjugierte Gradientenverfahren steht die Hesse Matrix nicht direkt zur Ver-

fügung), und eine numerische Ableitung zweiter Ordnung jeweils 2 Funktionsaufrufe benötigt, entsteht für die numerisch bessere Stabilität ein zusätzlicher Bedarf an $2 \cdot N$ Funktionsaufrufen pro Iteration, was das Verfahren bei einer teuren Funktionsauswertung - wie es im vorliegenden der Fall ist - leider nicht brauchbar macht.

Die Abbildungen 5.12 und 5.13 zeigen ein anfänglich schnelleres Konvergenzverhalten des Verfahrens in Bezug auf die Anzahl der Iterationen. Die Funktion bleibt jedoch - wie bereits das nicht vorkonditionierte Verfahren - an einem früheren Punkt in der Optimierung stecken.

Quasi Newton

Die Lösung von Gleichung 5.1 erfolgt in der vorliegenden Arbeit durch Anwendung einer Quasi Newton⁸ Methode. Bei dieser Klasse von Verfahren wird die zur Lösung des Systems benötigte inverse Hesse Matrix abermals nicht direkt berechnet, sondern schrittweise im Verlauf der Iterationen approximiert.

Der Vorteil beim Quasi Newton - im Vergleich zum reinen Newton Verfahren - ist, dass durch die Iterationsvorschriften (siehe [162] für Details) gewährleistet wird, dass die Approximation der inversen Hesse Matrix zu jedem Schritt positiv definit und symmetrisch ist. Es kann also garantiert werden, dass die Lösung (auch in weiterer Entfernung zum Minimum) immer in Richtung des Minimums geht und, je näher die Lösung der Minimalstelle kommt, die inverse Hesse Matrix besser und besser approximiert wird.

Nachteile wie ein erhöhter Speicherplatzverbrauch (die gesamte Hesse Matrix muss während des Verfahrens gesichert werden, was den Platzverbrauch um N^2 erhöht) können bei der zu erwartenden Größe des Gleichungssystems und den zur Verfügung stehenden Rechnern vernachlässigt werden.

Vergleicht man das Quasi Newton Verfahren (implementiert wurde die Methode von Broyden, Fletcher, Goldfarb & Shanno (BFGS)) mit den beiden bisherigen Verfahren (Abb. 5.12 & 5.13), fällt auf, dass die Konvergenz dem konjugierten Gradientenverfahren mit Präkonditionierung stark ähnelt während aber final ein besseres Optimum gefunden wird. Ohne weiteres Wissen über das Verfahren lässt sich dieses (wünschenswerte) Verhalten nicht allein aus dem Quasi Newton Algorithmus heraus begründen, weshalb im Folgenden dieses Verhalten weiter untersucht werden soll.

Glättung des Funktionsraums

Das Verhalten des verwendeten BFGS Algorithmus, ein besseres Optimum als vergleichbare KG Verfahren zu finden, konnte in einer Vielzahl Messungen bestätigt werden (siehe auch Abbildung 5.14). In der Endphase konvergiert das KG Verfahren bevor ein ideales Minimum gefunden werden kann. Problematisch ist das vor allem, weil lokale Tiefstellen physikalisch nicht motivierte und entsprechend deformierte Instrumentenpositionen darstellen können (siehe Abschnitt 5.1.8).

Die Vermutung liegt daher nahe, dass die Verfahren der konjugierten Gradienten vorzeitig in einem lokalen Minimum "stecken bleiben" und daher nicht mehr in der Lage sind, ein globaleres Minimum zu finden. Um diese Vermutung zu bestätigen, wurden mehrere Versuchsreihen aufgenommen, in denen die erreichten Lösungen eines

⁸oder auch Methode variabler Metrik

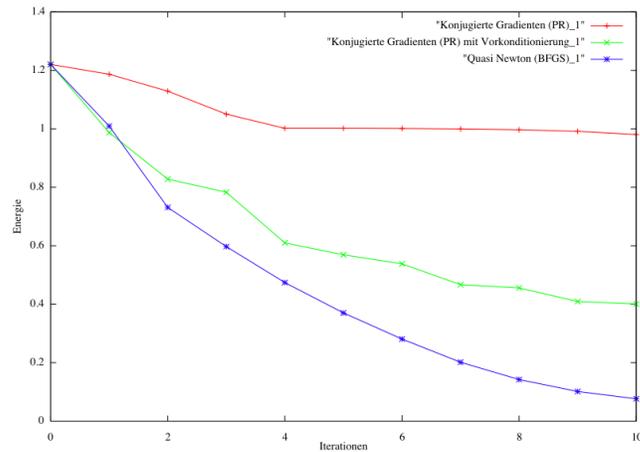


Abbildung 5.14: Verlauf der verschiedenen Lösungsverfahren in Nähe zum Optimum

KG Verfahrens nachträglich mit Zufallswerten perturbiert wurden und das Verfahren erneut gestartet wurde. Die Ergebnisse sind in Abbildung 5.15 zu sehen.

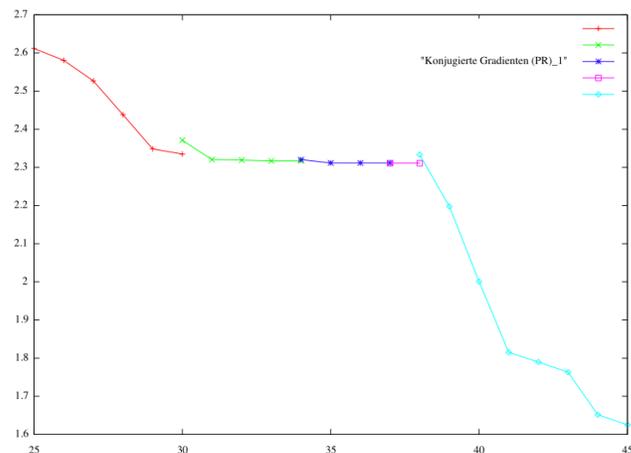


Abbildung 5.15: Neustart des KG Verfahrens nach Perturbation der gefundenen Lösung

Es zeigt sich, dass das KG Verfahren mehrfach konvergiert bevor mit einer Perturbation der Lösung schließlich eine (offensichtlich vorhandene) lokale Tiefstelle verlassen werden kann. Um das Verfahren stabiler werden zu lassen, ist es also nötig zum Einen zu untersuchen, warum die Implementierung des Quasi Newton Verfahrens nicht unter diesen Problemen leidet und zum anderen, wie die Zielfunktion entsprechend angepasst werden kann um das Auftreten dieser lokalen Tiefstellen möglichst zu vermeiden.

An erster Stelle soll die Glättung der Zielfunktion stehen. Stellt man sich anschaulich die Funktion im zweidimensionalen als Gelände mit globaler Tiefstelle und eini-

gen flachen lokalen Minima (Kuhlen) vor, so muss das Lösungsverfahren daran gehindert werden, diese Kuhlen zu erreichen. Ein mögliches Verfahren dazu ist es, die Auswertung der Gradienten im Funktionsraum einer Trägheit zu unterwerfen. Vorstellbar ist dieser Effekt - um wieder ein Kugelbeispiel anzuführen - mit einem schweren Ball, der Aufgrund einer Trägheit nicht direkt in eine kleine Vertiefung rollt sondern der Spur des globalen Tiefpunkts folgt. Formell ausgedrückt bedeutet das, dass Gradienten nicht direkt ausgewertet, sondern gleitend nach der Formel

$$g_{i+1} = \varphi \cdot g_i + (1 - \varphi) \cdot g \quad (5.25)$$

berechnet werden, wobei φ einem Interpolationsparameter im Bereich $[0, 1]$ darstellt und g den aktuell ausgewerteten Gradienten bezeichnet. Über Anpassung des Parameters φ kann der Einfluss des aktuellen Gradienten kontrolliert werden. Die Abbildungen 5.16 und 5.17 zeigen das Verhalten des konjugierte Gradienten Verfahren mit verschiedenen Parametern θ .

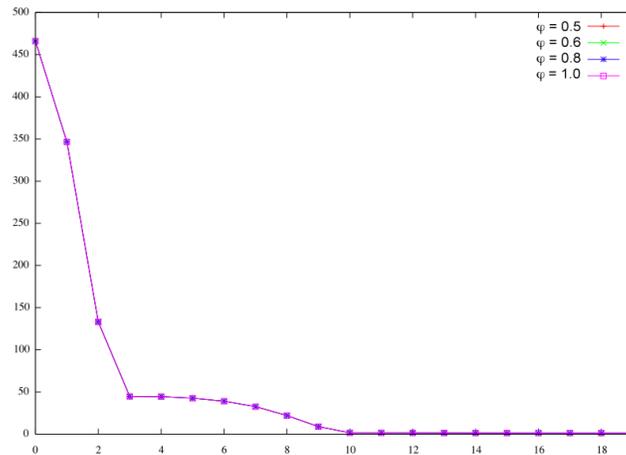


Abbildung 5.16: Verhalten des Konjugierte Gradienten Verfahren mit gleitender Gradientenberechnung und verschiedenen Parametern θ .

Auf 5.16 zeigt sich klar, dass das globale Verhalten des Lösungsverfahrens unverändert bleibt. In der Vergrößerung (5.17) wird jedoch deutlich, dass lokal bessere Tiefstellen erreicht werden - kleine örtliche Minima also umgangen werden. Auch wenn es sich bei diesem Verfahren natürlich um eine Heuristik zur Glättung der Funktion handelt, so kann man doch von einer Annäherung an physikalische Gegebenheiten ausgehen. Da die zu berechnende Originalfunktion die Trägheit der Gleichungen, die einer solchen physikalischen Berechnung innewohnt nicht implementiert, wird sie auf diese Weise nachträglich im Lösungsverfahren angenähert.

Offen bleibt die Frage, warum das verwendete Quasi Newton Verfahren bereits ohne eine entsprechende Anpassung der Gradienten gute Ergebnisse liefern konnte. Bei näherer Betrachtung der verwendeten Algorithmen zeigt sich, dass auch Quasi Newton Algorithmen Liniensuchen in Richtung der errechneten Gradienten verwenden. Der Unterschied (in der vorliegenden Implementierung!) liegt aber in der Form der Liniensuche: Während die Verfahren der konjugierten Gradienten in der Liniensuche eine

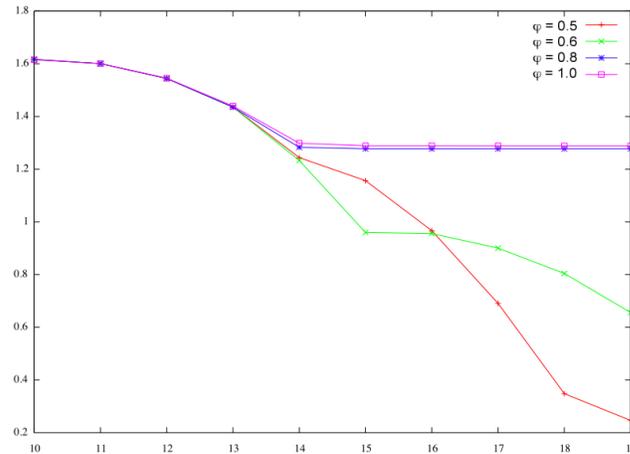


Abbildung 5.17: Verhalten des Konjugierte Gradienten Verfahren mit gleitender Gradientenberechnung und verschiedenen Parametern θ (Vergrößerung von 5.16).

möglichst exakte Berechnung des Minimums entlang des Gradienten (als Abbruchkriterium dient hier der Funktionsgradient, der möglichst orthogonal zur Richtung der Minimierung werden soll) verwenden, versucht das hier verwendete BFGS Verfahren nur eine prozentuale Verbesserung des Funktionswertes in der Minimierungsrichtung zu erwirken - das eigentlich Minimum wird nicht in jeder Iteration zwingen erreicht.

Dadurch entsteht ein in erster Annäherung ähnliches Verhalten wie durch die Trägheit der Gradienten. Kleine, flache lokale Tiefstellen werden nicht bis zum Optimum hin optimiert was zu einer "Nicht Konvergenz" des Verfahrens in lokalen Tiefstellen führt. Kombiniert man die Trägheiten der Gradienten mit einer bewusst approximativen Liniensuche, kann eine Konvergenz des Verfahrens in einem möglichst idealen Tiefpunkt in den meisten Fällen sichergestellt werden.

Schlägt das Verfahren dennoch fehl, so kann durch eine leichte Perturbation der Ergebnisse zu Beginn jedes Lösungsprozesses erreicht werden, dass Verfahren spätestens nach einigen (wenigen) Lösungsprozessen in einem idealen Tiefpunkt konvergiert, was bei einer interaktiven Rate von ca. 20 Lösungen pro Sekunde gut ausreicht um ein stabiles Verfahren zu generieren.

5.1.9 Geschwindigkeitsoptimierung

Ein weiteres Problem ist die Rechengeschwindigkeit - selbst bei geschickter Implementierung der oben angegebenen Formeln, ist die angegebene Vorgehensweise nicht ausreichend interaktionsfähig. Gründe dafür sind vor allem kostenintensive Operationen, die während der Lösung des Optimierungsproblems wiederholt ausgeführt werden müssen.

Ein Großteil der Rechenleistung wird während der Optimierung für die Bestimmung der Distanzfunktion (F_k) aufgewandt. Da Gefäßstrukturen in der Simulation als 3D Dreiecksnetze vorliegen (siehe Kapitel 5.3), stellt die Bestimmung des korrekten Abstands zur undeformierten Wand ein nicht triviales Problem dar.

Die offensichtliche Lösung wäre die Bestimmung des vorzeichenbehafteten Ab-

stands zu jedem Dreieck einer Struktur nach einem schnellen Verfahren (bsp. [103]) und anschließende Auswertung des geringsten Abstands. Dieses Verfahren erweist sich aber bei zunehmender Anzahl an Dreiecken (> 1.000 pro Struktur, > 250.000 im Mesh des gesamten Gefäßsystems) als äußerst zeitaufwendig. Diverse Techniken (Octrees, Bereichsabschätzungen, multiple Detaillevels...) wie im Abschnitt 4.4.2 vorgestellt, können die Laufzeit optimieren - aber nicht auf ein echtzeitfähiges Niveau anheben.

Distanzfelder

Eine Möglichkeit, das Optimierungsproblem in deutlich kürzer Zeit zu lösen, bieten Cache optimiert implementierte adaptive Distanzfelder (für Informationen zur Generierung - siehe Abschnitt 5.3). Die Rechenzeiten sind deutlich kürzer und vor allem nicht mehr von der Anzahl der Dreiecke abhängig, da die Distanzfelder bereits im Vorfeld berechnet werden können.

Ein weiterer großer Vorteil der Distanzfelder ist ihre Datenunabhängigkeit. Die Existenz einer globalen Abstandsfunktion vorausgesetzt, kann für jeden Datentyp (bsp. Dreiecksnetze, Volumendaten, Zylinder-Kugel Modelle [118]) ein Distanzfeld erstellt und somit ohne weiteres in der Simulation genutzt werden. Die drei oben genannten Typen wurden im Rahmen dieser Arbeit umgesetzt und ausführlich getestet.

Ein bestehendes Problem bei Distanzfeldern, ist die Auflösung und der damit verbundene Speicherverbrauch. Durch die adaptive (Octree basierte) Speicherung wird zwar effektiv eine Kompression ermöglicht, eine Verdoppelung der Auflösung (in jeder Richtung) bringt aber immer noch (grob) eine Verachtfachung des Speicherbedarfs mit sich.

Aus diesem Grund wird die Dimension der Distanzfelder bewusst klein gehalten (max metrische Auflösung $1mm$, Feldgröße von ca. $1.4\times$ der euklidischen Größe der Struktur in jeder Richtung). Abbildung 5.18 zeigt exemplarisch ein großes generiertes Distanzfeld.

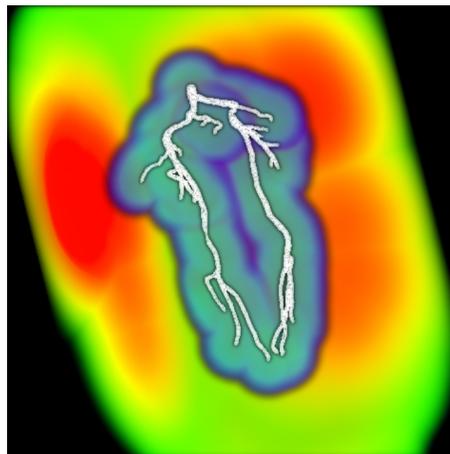


Abbildung 5.18: Großes Distanzfeld der Beinarterien

Außerhalb des definierten Wertebereichs eines Feldes kann der Distanzwert eines

Ortes x mit hoher Genauigkeit geschätzt werden, indem der Distanzwert im Schnittpunkt aus der Geraden, die den Mittelpunkt des Distanzfeldes mit x verbindet und der das Feld begrenzenden Fläche zur euklidischen Distanz zwischen dem Schnittpunkt und x hinzuaddiert wird.

Trotzdem wird auch bei effektiver Implementierung der Distanzfelder, noch viel Rechenzeit auf den Lookup und die darauf folgenden Werteinterpolierung verbraucht. Da nur eine beschränkte Genauigkeit benötigt wird, bietet es sich also an, ein Verfahren zur Schätzung der Werte zu implementieren.

Linearisierung

Ein weiteres Mittel zur Beschleunigung wird daher durch eine grundlegende Eigenschaft der Gefäßstrukturen geboten. Gefäße sind in der Regel glatte Strukturen. Graten und spitze Kanten kommen äußerst selten vor und können bei Bedarf ohne Einfluss auf die Gesamtsimulation geglättet werden. Die daraus resultierenden Distanzfelder zeichnen sich durch ebenso glatte Verläufe aus, was die Approximation nahe gelegener Werte durch bekannte Verfahren vereinfacht.

Rechnerisch werden bei jeder Auswertung des Instrumentengradienten durch zentralen Differenzenquotienten bei N Gelenken und drei freien Winkelparametern $2 * 3 * \frac{N^2}{2}$ Auswertungen der Distanzfunktion benötigt (siehe Formel 5.18). Bei geschickter Schätzung lassen sich diese auf N Auswertungen der Distanzfunktion auf den Distanzfeldern reduzieren

Euklidische Gradientenfelder auf den Distanzen können im Vorfeld berechnet und gespeichert werden. Da die Felder wie erwähnt in der Regel als glatt angenommen werden können, ist ausgehend von der Existenz eines Wertes d_i an einem Punkt x_i und dem dazugehörigen Gradienten g_i eine Abschätzung des Wertes am Punkt \tilde{x}_i durch Anwendung einer einfachen Taylorreihe 1. Ordnung (Linearisierung der externen Gradientenfunktion an Stelle x_i) möglich:

$$E_{ext}^i = d_i + (g_i * |x_i - \tilde{x}_i|) \quad (5.26)$$

Die Abschätzung mit einer Taylorreihe 1. Ordnung ist insbesondere deshalb hinreichend genau, weil innerhalb eines Würfels des adaptiven Distanzfeldes die Werte der Ecken linear interpoliert werden. Durch die Linearisierung der Funktion geht also, zumindest innerhalb eines Knotens keine Genauigkeit verloren.

5.1.10 Algorithmus

Der eigentliche Algorithmus zur Katheterberechnung läuft in mehreren Schritten ab: Translation und Rotation werden auf die Instrumente übertragen und ihre Pakete entsprechend gegeneinander verschoben. Anschließend werden die Pakete kombiniert und das Proxy Instrument erstellt. Das Proxy Instrument hat entsprechend die Länge des längsten vorhandenen Instruments. Ist das Proxy Instrument länger (oder kürzer) als es die derzeitige Konfiguration der Gelenke zulässt, werden Gelenke (an der proximalen Basis) hinzugefügt bzw. entfernt.

Anschließend werden die Pakete des Proxies an den Gelenkpositionen abgetastet und die Werte entsprechend auf die Gelenke übertragen. Kurvenssegmente werden nach

den Regeln der Trigonometrie auf α/β Winkel umgerechnet, Rotationen als Torsion an der Basis (bzw. an Übergängen der Instrumente) eingefügt.

Die so entstandene Gelenksrepräsentation wird gelöst und - da zur geglätteten Darstellung der Katheterformen im Simulator Splines (lokal interpolierende Kardinal Splines mit $t = 0$ (Catmull-Rom [50])) verwendet werden - die resultierende Form als Spline auf die Ursprungsinstrumente übertragen. Bild 5.19 zeigt eine schematische Demonstration des Verfahrens. Catmull Rom Splines werden verwendet, da sie als Unterklasse der Hermite Splines schnell zu berechnen sind, eine kontinuierliche (stetig ableitbare) Form bilden und (im Gegensatz zu beispielsweise B-Splines) jeden vorgegebenen Punkt des Instrumentes exakt interpoliert.

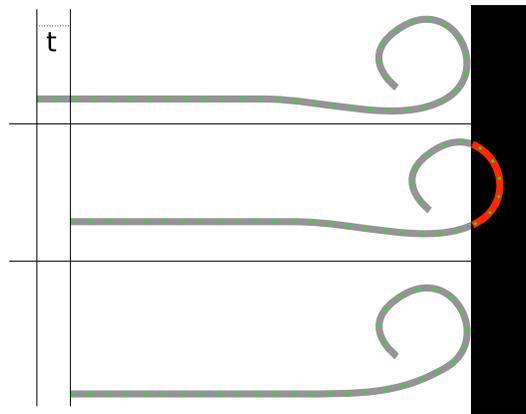


Abbildung 5.19: Schematische Darstellung der Instrumentenberechnung

5.2 Resultate der Instrumentensimulation

Die Instrumentensimulation muss, will man sie objektiv bewerten, auf zwei Gesichtspunkte hin untersucht werden. Die Geschwindigkeit, wobei die Frage im Vordergrund steht ob eine Berechnung der Instrumente in interaktiver Echtzeit möglich ist. Und der Grad an Realismus - verhält sich ein Instrument in der Simulation so, wie man es in der Realität erwarten würde. Beide Punkte sollen im Folgenden näher untersucht werden.

5.2.1 Realitätsnähe

Da die Simulation auf deskriptiven Parametern (Steifigkeit und Wandelastizität sind nur schwer messbar und werden von den Herstellern der Instrumente nicht direkt angegeben) beruht, kann die Betrachtung einer Realitätsnähe nur unter eingeschränkten Gesichtspunkten erfolgen. Über die Einstellung der Parameter, kann die Simulation an äußere Bedingungen angepasst werden (siehe Abbildung 5.20), was zu der Anforderung führt, dass die Simulation unter angepassten Bedingungen realistische Resultate liefern muss.

Um dennoch einen Vergleich mit der Realität herstellen zu können, wurde die Simulation mit einer kontrollierten Umgebung verglichen. Ein Katheter wurde in ein

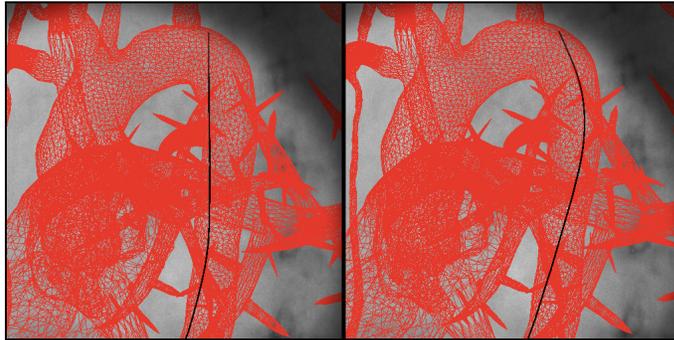


Abbildung 5.20: Veränderung des Instrumentenverhaltens über Anpassung der Steifigkeit

Phantom aus (geölten) Kunststoffschläuchen eingebracht und fotografisch festgehalten. Anschließend wurde das Phantom vermessen, identisch in einer 3D Umgebung (Cinema 4D) nachgebaut und in der Simulationsumgebung als Gefäßmodell verwendet. Das Verhalten des virtuellen Instrumentes wurde durch Justage der Parameter auf die Deformationen des realen Katheters angepasst (siehe Abbildung 5.21 und 5.22) und die resultierenden Werte als Grundlage für den Aufbau eigener Instrumente verwendet.

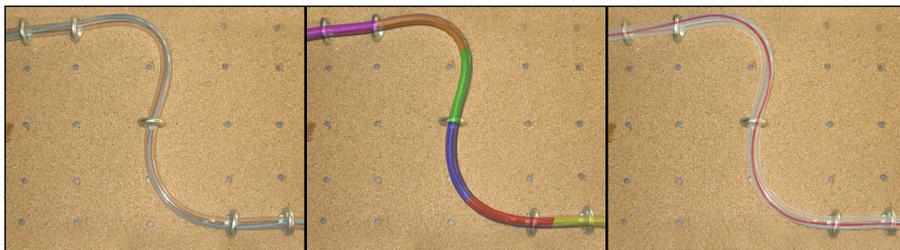


Abbildung 5.21: Aufbau des Phantoms (Ausschnitt), resultierendes Röhrenmodell und überlagert dargestellter Katheter.

Die resultierenden virtuellen Instrumente wurden mehreren erfahrenen Kardiologen zur Evaluation übergeben. Die Reaktionen bescheinigen der Simulation einstimmig eine hohe Korrelation mit der Realität. Teilweise waren keine weiteren, bei einigen wenigen Instrumenten geringe (oft sehr spezifische) Anpassungen nötig, um den subjektiven Ansprüchen der Versuchspersonen gerecht zu werden. Kardiologen, die bereits andere Simulationssysteme getestet hatten, kamen vermehrt zu dem Schluss dass die vorgestellte Simulation anderen Verfahren subjektiv überlegen ist und sich realistisch verhält.

Es kann daher davon ausgegangen werden, dass die Instrumentensimulation ihrem Einsatzzweck in Bezug auf die Abbildung der Realität genügt und der zweite Punkt - die Performanz untersucht werden.



Abbildung 5.22: Vergleich realer Katheter (blau) und virtuelles Instrument (rot).

5.2.2 Performanz

Die Akzeptanz der Instrumentensimulation ist stark von der Updategeschwindigkeit abhängig. Liefert der Algorithmus zu wenig Bilder pro Sekunde, lehnt der Benutzer das System ab, da die Korrelation zwischen seinen Handlungen (Manipulation des Instruments) und dem visuellen Feedback zu stark abnimmt.

Das Verfahren wurde daher stark auf Geschwindigkeit optimiert. Dabei hängt die erreichbare Updaterate neben der Länge des Instruments (Anzahl der Gelenke) auch von der Topologie und Beschaffenheit der umliegenden Strukturen ab, die Einfluss auf die Komplexität des Lösungsraums nehmen.

Ab einer Bildrate von ca. zehn Bildern pro Sekunde beginnt der Benutzer in der Regel das System zu akzeptieren. 20-25 Bilder pro Sekunde sind für anspruchsvollere Nutzer üblich. 50 Bilder pro Sekunde werden in der Kinderkardiologie gelegentlich bei sehr schnell schlagenden Herzen verwendet - allerdings ist es nicht nötig, den Katheter bei jedem Bild neu zu berechnen. Die Berechnung kann asynchron mit 20 Bildern pro Sekunde erfolgen, ohne eine Beeinträchtigung des Nutzerverhaltens feststellen zu können.

Diagramm 5.23 zeigt die Rechenzeit pro Frame für ein Instrument mit wachsender Anzahl an Gelenken in einer topologisch realistischen Umgebung (Standard Kinderherz Datensatz) mit allen Abzweigungen. Dabei wurde bei dieser ersten Messung auf die Linearisierung als Beschleunigungsfaktor verzichtet.

Die Messungen wurden auf einem Macbook Pro (Laptop) mit 2.4 Ghz Intel Core2 Duo Rechner mit 2GB Hauptspeicher vorgenommen. Bei allen Werten ist zu erwähnen, dass sie während der laufenden Simulation (Berechnung von Grafik, Deformation etc. wurde nicht abgeschaltet) aufgenommen wurden und daher auch im fertigen Simulator erreicht werden. Zudem wurde das Instrument während der Messungen ständig in Bewegung gehalten (Rotation und Translation) um nicht in lokalen Tiefpunkten des Lösungsraums verweilend Cacheeffekte beim Distanzlookup und schnelle Rechenzei-

ten durch frühere Konvergenzen auszunutzen.

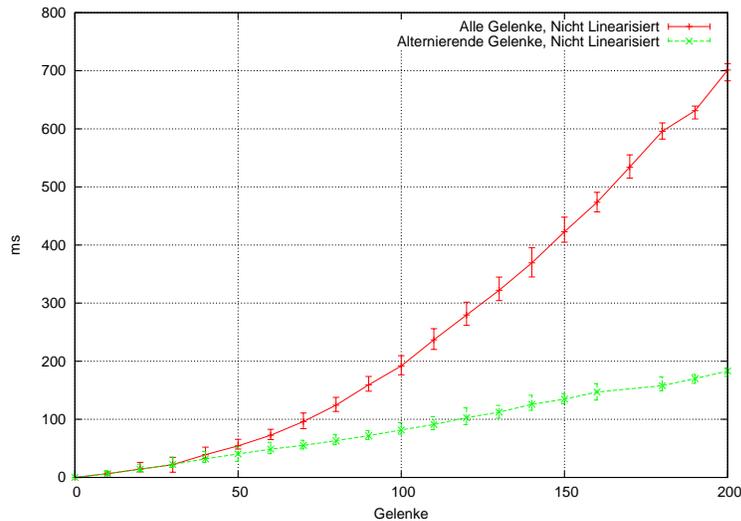


Abbildung 5.23: Rechenzeit ohne Linearisierung in Relation zu den Gelenken.

Die rote Linie zeigt die Zeit pro Bild in Millisekunden, wenn jedes Gelenk voll berechnet wird. Geht man von einer (ausreichend realistischen) Gelenklänge von ca. 3 mm aus, sieht man, dass der kritische Punkt von zehn Bildern pro Sekunde bei etwa 70 Gelenken (entspricht 21 cm Instrument) unterschritten wird. Damit ist der Anspruch an die interaktive Performance noch nicht erreicht.

Die grüne (gestrichelte) Linie zeigt im Vergleich dazu eine Technik, die nicht jedes Gelenk in jedem Rechenschritt berechnet sondern proximal gelegene Gelenke (die in der Regel weniger Fluktuation unterworfen sind) alternierend berechnet. Das Verhalten des Instruments ändert sich subjektiv nicht, es werden jedoch bereits ca. 120 Gelenke (entspricht 36 cm) in ausreichender Zeit nutzbar.

Erschwerend kommt jedoch hinzu, dass in der Kinderkardiologie häufig zwei oder drei Instrumente parallel verwendet werden. Daher sind weitere Beschleunigungen nötig. Diagramm 5.24 zeigt die gleiche Messung unter Verwendung der oben beschriebenen Linearisierungsabschätzungen.

Auch hier zeigt die rote Linie die Zeit pro Rechenschritt in Millisekunden, wenn jedes Gelenk voll berechnet wird. Allerdings ist eine Geschwindigkeitssteigerung um etwa einen Faktor von fünf feststellbar. Diesmal wird der kritische Punkt entsprechend, trotz voller Berechnung, erst bei etwa 160 Gelenken (entspricht 48 cm Instrument) durchbrochen.

Zieht man noch das Alternierungsverfahren hinzu, kann ein performantes Verhalten des Instruments noch mit über 300 Gelenken (entspricht 90 cm) erzielt werden.

In der Kinderkardiologie werden in der Regel Instrumente mit einer Länge von maximal ca. 50 cm verwendet. Da der Einsatzpunkt in der Simulation etwas später (ca. 10

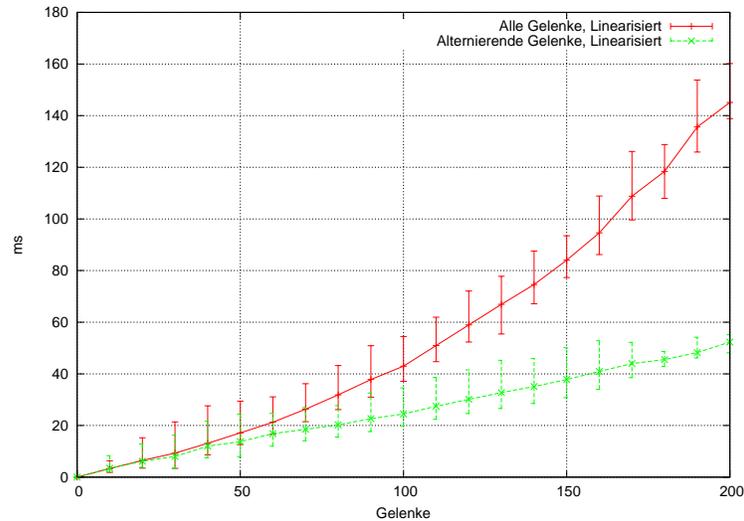


Abbildung 5.24: Rechenzeit mit Linearisierung in Relation zu den Gelenken.

cm vor dem Herzen) erfolgt reichen 30 cm - 40 cm pro Instrument für die abzubilden Situationen vollständig aus. Mit der alternierenden Technik wird bei entsprechen 100-120 Gelenken eine Zeit von ca. 25 ms pro Bild erzielt was einer Framerate von ca. 40 Bildern pro Sekunde entspricht - den oben gestellten Anforderungen also vollständig genügt.

Weiter Techniken, wie etwa die Abschaltung der Berechnung, wenn über längere Zeit keine Bewegung und Optimierungsmöglichkeit mehr festgestellt werden kann, tragen weiter dazu bei dass auch mehrere unabhängige Instrumente zeitgleich genutzt werden können.

5.3 Datengewinnung

Um die Simulation in einer möglichst realistischen Umgebung gestalten zu können, ist eine Verwendung von Originaldaten aus dem klinischen Alltag notwendig. Auch sollen damit Möglichkeiten geschaffen werden, zu prüfen, ob das Langziel einer patientenspezifischen Trainingsumgebung einen positiven Einfluss auf Therapiemöglichkeiten haben kann.

5.3.1 Rekonstruktion

Die in der vorliegenden Simulation verwendeten Daten stammen aus der halbautomatischen Rekonstruktion von CT- & MRT Daten (siehe Abbildung 5.25).

Die unter Kontrastmittelgabe aufgenommenen Volumen eines Kinderherzens wurden mittels 3D Live Wire Algorithmus ([115]) segmentiert und anschließend unter

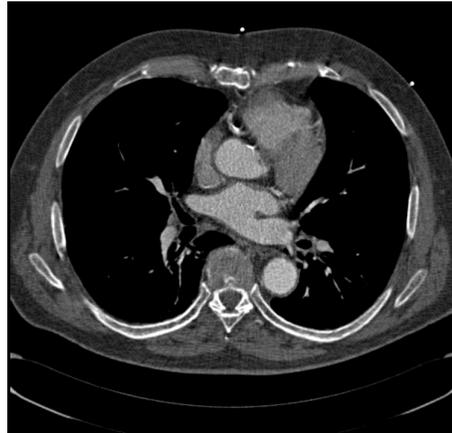


Abbildung 5.25: CT Daten als Grundlage der Herzrekonstruktion. Die klare Abgrenzung der Kompartimente voneinander ist eine wichtige Voraussetzung für die Segmentierung und Rekonstruktion.

Verwendung des bekannten Marching Cubes Verfahrens ([136] in Polygonale Daten überführt (siehe Abbildung 5.26).

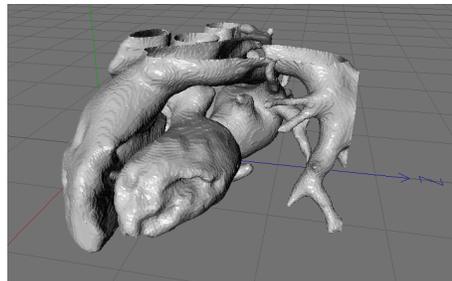


Abbildung 5.26: Rekonstruierte Polygonale Daten. Gut zu erkennen sind die durch den Marching Cubes Algorithmus entstandenen Fragmente (Treppenbildung) auf der Oberfläche des nicht geglätteten Modells.

Die polygonalen Daten werden für die spätere Visualisierung des animierten Herzens verwendet und dienen auch der Kollisionsdetektion bei der Berechnung des Kontrastmittelflusses. Weitere Standardverfahren aus der bekannten Literatur wie Oberflächenglättung und Dreiecksdezimierung wurden angewandt, um das endgültige Herzmodell (siehe Abbildung 5.27) weiter zu verbessern.

5.3.2 Baumstrukturen

Die in Abschnitt 5.1.7 vorgestellte Unterteilung in Baumstrukturen geschieht halbmanuell. Geeignete Schnittpunkte werden manuell selektiert und an ein im Rahmen dieser Arbeit geschriebenes Plugin der Software Cinema4D [10] übergeben, das mittels Greedy Algorithmus einen möglichst optimalen Schnittpfad erstellt ohne Polygone zu

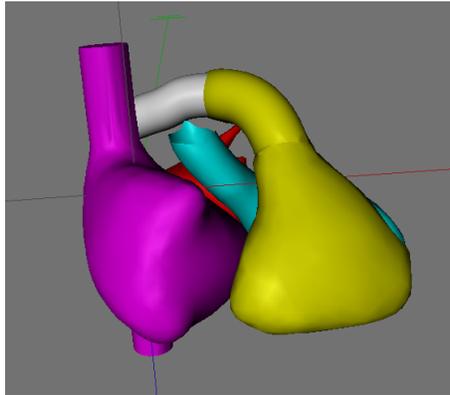


Abbildung 5.27: Geglättete Polygonale Daten. Die Glättung und Dreiecksdezimierung geschieht automatisch, ebenso die Unterteilung in *nicht zusammenhängende* Kompartimente.

teilen.

Da der Simulator mit triangulierten Daten arbeitet, dürfen Dreiecke nicht neu geschnitten werden. Eine Neutriangulierung der einzelnen Segmente wäre die notwendige Folge, was unter Umständen neue, trotz zusammengehöriger Segmente, disjunkte Vertices auf der Schnittkante erzeugen würde. Ein erneutes Zusammenfügen, wie es bei der Berechnung der Distanzfelder notwendig ist, würde dadurch erschwert werden.

Die in Bild 5.28 dargestellte Substrukturierung wurde um einen Gefäßdatensatz erweitert, der während der Entwicklung des Simulators freundlicherweise von der Firma CATHI GmbH zur Verfügung gestellt wurde.

5.3.3 Wandlung in Distanzfelder

Im Simulator werden zwei Typen von adaptiven Distanzfeldern verwendet. Niedrig (ca. 3 mm) aufgelöste Distanzfelder der einzelnen Knoten (Segmente), die während der Zuordnung der Gelenke zu den Gefäßstrukturen verwendet werden und höher aufgelöste (ca. 0.5 mm), die das betroffenen Segment und die Nachbarknoten eines Segmentes in einem Distanzfeld vereinen und während der eigentlichen Optimierung genutzt werden. Das ist nötig, da offene Segmente, die bei der Unterteilung entstehen während der Generierung der Distanzfelder geschlossen werden und somit Barrieren für die Optimierung der Instrumente entstehen könnten.

Die Generierung der niedrig aufgelösten Distanzfelder geschieht über vozeichenbehaftete Distanzberechnung mit mehrgängigen Scan Line Algorithmen (siehe [70]). Andere, schnellere Verfahren, wie die Ermittlung des Vorzeichens über Skalarprodukt zu Pseudonormalen (siehe auch [25]), zeigten sich als zu fehleranfällig in Bezug auf die Polygondaten im Zusammenspiel mit den verwendeten kommerziellen Modellierungsapplikationen (Maya 6.5 und Cinema 4D R10). Schon kleine Löcher oder leicht degenerierte Normalen, wie sie während der Oberflächenglättung und/oder Dreiecksdezimierung entstehen können (der zu Beginn stehende Marching Cubes Algorithmus führt zu vollständig geschlossenen Netzen, die Weiterverarbeitung in den kommerzi-

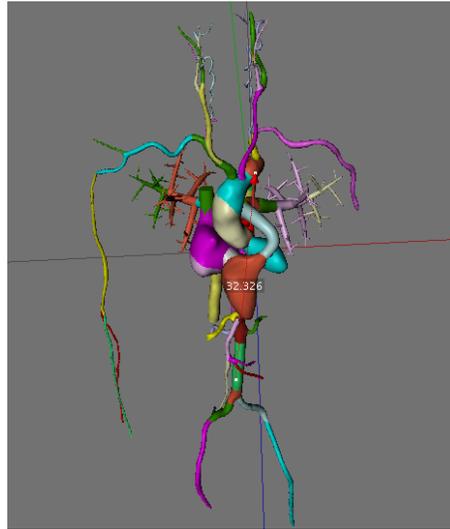


Abbildung 5.28: Komplett bearbeiteter und unterteilter Datensatz. Die Unterteilung in einzelne Segmente geschieht halbautomatisch durch Selektion geeigneter Schnittpunkte und anschließende Suche eines optimalen Schnittpfades.

ellen Programmen liegt aber außerhalb des Einflussbereichs dieser Arbeit), führen zu inkorrekt er Vorzeichenbestimmung. Bei idealen Polygondaten stellen die Pseudonormalen aber eine schnelle und exakte Alternative dar.

Um Löcher während der Berechnung zu vermeiden, wird bei jedem Segment vor der Generierung des Distanzfeldes eine Detektion und anschließender Verschluss der Löcher (bsp. Schnittstellen) über Ear Cut Triangulierung [184] durchgeführt.

Die hochaufgelösten Distanzfelder (im Folgenden auch Nachbarschaftsfelder genannt) werden ebenso über Scan Line Verfahren berechnet. Allerdings werden benachbarte Segmente vor dem Verschluss der Löcher und der anschließenden Berechnung des Feldes zu einer einzelnen Struktur kombiniert. An diesem Punkt ist es wichtig, dass zusammengehörige Vertices aufeinanderfallen, da sonst keine erfolgreiche Kombination erfolgen kann und Schnittstellen als Löcher detektiert werden können, was zum Verschluss der Schnittstellen und Einbringung von Barrieren führt.

Die eigentlichen Distanzfelder werden in der Form von Octrees gespeichert (adaptive Distanzfelder). Dadurch wird der Fußabdruck der Daten im Speicher klein gehalten und, durch die cacheeffiziente Implementierung, ein schneller Lookup der benötigten Daten ermöglicht.

5.4 Animation

Die Animation des schlagenden Herzens stellt eine wichtige Voraussetzung für die Akzeptanz der Simulation dar. Laut Aussagen von Ärzten spielt die Bewegung für die Führung der Instrumente zwar nur eine stark untergeordnete Rolle, die Simulation auf unbewegten Datensätzen führt aber schnell zur Ablehnung des Systems.

Animationen werden in der Simulation über FFD⁹ Volumina berechnet. Dabei handelt es sich um Splinevolumina, die durch eine beliebige Anzahl an in einem (regulären) Gitter angeordneten Stützpunkten definiert, durch Translation der Stützpunkte deformiert und in den Zwischenräumen mit Hilfe trivariater Bernsteinpolynome glatt interpoliert werden können.

$$FFD(u, v, w) = \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n P_{ijk} \theta_i^l(u) \theta_j^m(v) \theta_k^n(w) \quad (5.27)$$

Dabei sind l , m und n die Grade der Bernstein Polynome in der U , V und W Achse, die gemeinsam eine orthonormale Basis bilden. u , v und w sind entsprechende Repräsentanten dieser Basis. $P_{i,j,k}$ sind die Kontrollpunkte und θ_i^l bezeichnet die i -te Basisfunktion vom Grad l .

Bild 5.29 zeigt die Auswirkung der FFD auf das innenliegende Herzmodell. Das FFD Volumen wurde in diesem Fall manuell erstellt und visuell mit CT Daten abgeglichen. In der Simulation wird die Translation der Stützpunkte durch ein Keyframe basiertes System beschrieben. Die Deformation der das Herzmodell definierenden Vertices wird online aus der FFD berechnet.

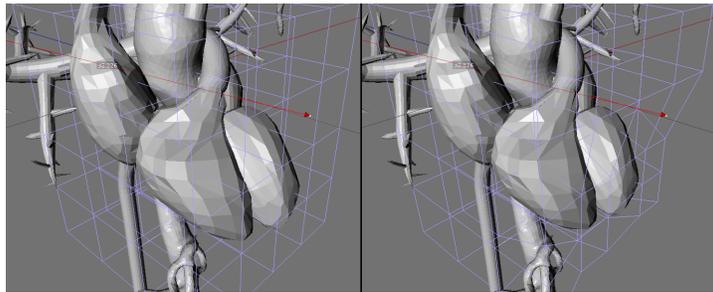


Abbildung 5.29: Auswirkung einer FFD auf das Herzmodell

Punkte, die nicht im (undeformierten) Ursprungsvolumen liegen, sind nicht von der Deformation betroffen. Ein Effekt, der an Übergangsstellen zu starken Störungen führen kann, da zwei benachbarte Vertices sich ggf. abrupt voneinander entfernen. Es gäbe zwar die Möglichkeit, das FFD Volumen auf die gesamte Geometrie (inklusive statischer Geometrien wie distalen Gefäßen) auszudehnen - dann wären aber für die Beschreibung selbst einfacher Deformationen eine größere Menge von Stützpunkten nötig, die bei direkter Anwendung von Formel 5.27 zu einer stark erhöhten Rechenzeit führten. Alternative Interpolationsmethoden (bsp. Kubische Hermitesplines, Cardinalsplines, Lagrangeinterpolation) können diesen Mangel zwar bedingt ausgleichen, dennoch ist für jeden Vertex innerhalb des Ursprungsvolumens eine Berechnung nötig, was eine Verkleinerung des Volumens auf den unbedingt benötigten (Herzmuskel umschließenden) Raum nahelegt.

Um trotzdem Sprungstellen zu vermeiden, wurde Gleichung 5.27 in dieser Arbeit ein Abschwächungsfaktor hinzugefügt, der, abhängig von der minimalen Entfernung zum Rand des Volumens, die durch die FFD induzierte Bewegung linear dämpft.

⁹Free Form Deformation

Durch das vorgestellte Verfahren lässt sich die vorhandene Geometrie zwar visuell überzeugend animieren, Instrumente bleiben von dieser Art der Animation aber vorerst unbetroffen.

Um Abhilfe zu schaffen und die Animation des Herzmuskels im Zusammenspiel mit der Instrumentensimulation zu simulieren, wurden in dieser Arbeit zwei Wege implementiert und in Bezug auf Rechenleistung und Qualität miteinander verglichen.

Als trivialer Ansatz wurde das nach außen hin gedämpfte FFD Volumen direkt auf die simulierten Instrumente angewandt. Die Ergebnisse sind visuell laut Aussage diverser Kardiologen zwar überzeugend, im Endeffekt aber nicht physikalisch motiviert und somit nicht in der Lage, mögliche Seiteneffekte (bsp. könnten Instrumente durch die Bewegung des Herzmuskels abgelenkt werden) der Deformation abzubilden. Ein auf diese Weise animierter Katheter (mit eingeblendeten Herzdaten) ist in Abbildung 5.30 dargestellt.



Abbildung 5.30: Durch FFD Volumen direkt deformierter Katheter

Ein anderer, rechenaufwendigerer Ansatz ist die physikalische Simulation des deformierten Gewebes. Ausgehend von der durch das FFD Volumen vorgegebenen Transformation wird das umliegende Gewebe deformiert und die Ergebnisse anschließend für das in der Instrumentenberechnung benötigte Distanzlookup verwendet. Da die Abstandsinformationen der Segmente aber in Form von Adaptiven Distanzfeldern (Volumendaten) vorliegen ist es nicht möglich, eine vollständige Deformation der Abstandsdaten in interaktiven Zeitabständen zu berechnen. Das in dieser Arbeit verwendete Verfahren sieht daher eine inverse Deformation der Instrumente vor.

Anstatt das umliegende Gewebe zu deformieren wird die FFD Transformation invertiert und auf die Gelenke des Katheters angewandt - die so invertierten Punkte dienen als Koordinaten für die aus den Distanzfeldern interpolierten Distanzwerte.

Als Problem stellt sich dabei dar, dass es sich bei einer FFD im klassischen Sinne nicht um einen Diffeomorphismus - also nicht um eine umkehrbare Funktion handelt. Der Grund dafür ist trivial herzuleiten, wenn die Transformation der Kontrollvertices dazu führt, dass sich das Volumen der FFD mit sich selbst überschneidet (siehe Abbildung 5.31). Die invertierte Abbildung ist dann offensichtlich nicht mehr eindeutig und nicht mehr mit rein analytischen Mitteln lösbar.

Auch wenn man die Definition der FFD dahingehend beschränkt, dass eine Selbstüberschneidung nicht mehr möglich ist, so ist die Invertierung eines Bernstein Splinevolumen eine nicht triviale Aufgabe, die - gemessen daran, dass die Invertierung für jede Iteration des Optimierungsverfahrens über 100x angewandt werden muss - den Anforderungen der Interaktivität auf heutigen Rechnern nicht gerecht werden kann.

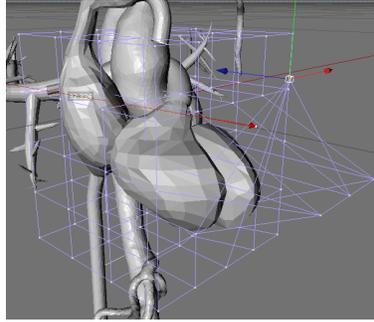


Abbildung 5.31: Sich selbst schneidendes FFD Volumen

Um die Berechnung dennoch in akzeptabler Zeit durchführen zu können, bietet sich eine Approximation der inversen Deformation an. Ein verhältnismäßig schnell berechenbares (selbst die Invertierung einer linearen Deformation wäre ein nicht triviales Problem) Verfahren kann auf Grundlage baryzentrischer Tetraederkoordinaten implementiert werden. Ähnlich dem invertierten Texturlookup, das häufig in der Computergraphik genutzt wird (in der Regel mit baryzentrischen Dreieckskoordinaten), werden die deformierten Subwürfel des FFD Volumens in fünf Tetraeder zerlegt, die das Volumen eines einzelnen Würfels vollständig ausfüllen. (siehe Abbildung 5.32)

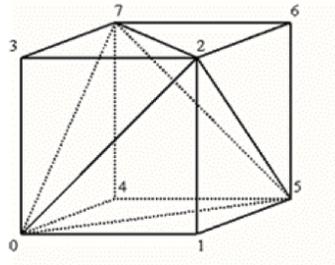


Abbildung 5.32: Zerlegung eines Würfels in fünf Tetraeder.

Bei der Invertierung eines Punktes x , der sich im FFD Volumen befindet, wird durch octreebeschleunigte Kollisionserkennung der (deformierte) Tetraeder \tilde{T} bestimmt, der x umgibt und anschließend x nach Formel 5.28 in baryzentrische Koordinaten ($b_i (i = 0..3)$) in Bezug auf \tilde{T} konvertiert.

$$b_i = \frac{\frac{1}{6} * \det M_i}{V} \quad (5.28)$$

Dabei bezeichnet b_i ($i = 0..3$) die i -te baryzentrische Koordinate, M_i die 4×4 Matrix, die in allen Spalten $\neq i$ die entsprechende homogene Eckkoordinate des Tetraeders und in der i -ten Spalte die homogene Koordinate des Punktes x enthält. V bezeichnet das Gesamtvolumen des Tetraeders \tilde{T} .

Mittels Gleichung 5.29, die in Bezug auf den undeformierten Tetraeder T angewandt wird, werden die baryzentrischen Koordinaten b_i wieder in euklidische Koor-

dinaten rückkonvertiert. Dadurch lässt sich (im Vergleich zu exakten Verfahren) eine relativ performante Berechnung des invertierten Punktes durchführen.

$$x = \sum_{i=0}^3 b_i * t_i \quad (5.29)$$

t_i bezeichnet in dieser Gleichung die Eckkoordinaten des Tetraeders. Bild 5.33 zeigt von links nach rechts ein Herz im undeformierten (Original), im vollständig deformierten und im (ausgehend vom deformierten) invertierten Zustand.

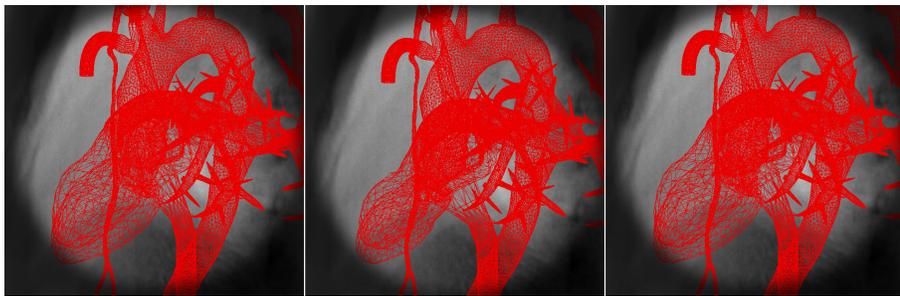


Abbildung 5.33: Deformation und inverse Deformation am Beispiel eines Herzmodells. Das Modell wurde erst mittels FFD deformiert und anschließend mittels approximierter inverser Deformation auf Grundlage baryzentrischer Tetraederkoordinaten zurückdeformiert.

Verwendet man derart deformierte Koordinaten in der Berechnung der Instrumente ergibt sich ein realistisches Bild der Deformation. Da aber auf diese Weise erheblich mehr Rechenzeit verbraucht wird als mit der oben beschriebenen - trivialen - Variante und die realen Auswirkungen (nach Aussagen von Ärzten) verschwindend gering sind, wird vorgeschlagen, in der Simulation vorerst die einfache Bewegungssimulation zu verwenden.

5.5 Kontrastmittel

Ebenso wie bei der Simulation des bewegten Herzens, stellt sich bei der Nachbildung des Kontrastmittelflusses die Frage, nach der Notwendigkeit für die exakte Simulation auf der einen und dem Anspruch an eine überzeugende, interaktionsfähige Geschwindigkeit auf der anderen Seite.

Generell zeigt sich beim Studium des Stand der Technik in Bezug auf Fluidsimulation, dass gegenwärtig immer ein starker Kompromiss zwischen physikalischer Genauigkeit und Rechenzeit getroffen werden muss. Keines der untersuchten und re-implementierten Verfahren ist derzeit in der Lage große Mengen an schnell beweglichen Fluids mit voller Kollisionserkennung in Echtzeit abzubilden. Als weiterer Punkt kommt hinzu, dass die Fluidsimulation parallel zur graphischen Darstellung und vor allem zur Berechnung der Instrumente auf einem einzelnen Rechner funktionieren muss.

Laut den Angaben der hierzu befragten Kardiologen spielt der exakte Ausbreitungsverlauf des Kontrastmittels (Verwirbelungen, Turbulenzen) für diagnostische Zwecke keine oder nur eine stark untergeordnete Rolle. Der Fluss des Kontrastmittels zwischen Kompartimenten (Gefäßen) unter unterschiedlichen Druckbedingungen muss allerdings beachtet werden. Blut fließt vom Hochdruckbereich der Aorta über einen eventuell vorhandenen Ductus in das (im Vergleich) Niederdruckgebiet der Lungenarterie - das schlagende Herz ändert sein Druckverhältnis.

Ebenso von Bedeutung ist die Einhaltung einer Durchleuchtungslänge. Durchqueren Röntgenstrahlen eine längere Strecke eines röntgenabsorbierenden Bereiches ist eine stärkere Schwärzung als bei feinen Adern wahrzunehmen, die nur von wenig Kontrastmittel durchflutet werden können.

Um diesen Ansprüchen zu genügen, wurde zu Beginn der Arbeit ein System entwickelt, das auf den Prinzipien der Smooth Particle Hydrodynamics (siehe Kapitel 4.5.1) aufbaut. Selbst mit der Unterstützung der Grafikkarte war es allerdings zu diesem Zeitpunkt nicht möglich, überzeugende Resultate in ansprechender Geschwindigkeit zu erzielen. Für eine ansprechende Darstellung und das Funktionieren der SPH Technik im Allgemeinen werden mehrere Tausend Partikel benötigt, die im kleinen Umkreis miteinander wechselwirken. Die Rechenzeiten lagen 2005 selbst bei effizienter Implementierung und fehlender (bzw. trivialer) Kollisionserkennung bei ca. 100 ms pro Rechenschritt¹⁰

Auch wenn die Ergebnisse visuell beeindrucken konnten, war bei steigender Partikelzahl kein interaktiver Betrieb mehr möglich. Weitere Schwierigkeiten stellen sich durch die meist hohe Geschwindigkeit des Kontrastmittels und die Dämpfung durch das Trägermaterial (Blut), die zu starken Verwirbelungen führt.

Aktuelle Forschungsergebnisse zeigen, dass eine interaktive Verwendung von SPH auf der GPU in Laufe der kommenden Jahre wahrscheinlich wird [59]. Es existieren heute bereits echtzeitfähige Implementierungen, die allerdings, bedingt durch die Limitierungen des GPU Processings, noch von primitiven Kollisionserkennungen (Bounding Box / Bounding Sphere) ausgehen.

Eine weitere Implementierung, basierend auf der 1999 von Stam vorgeschlagenen, gridbasierten Navier Stokes Methode [188], konnte in 3D nicht annähernd echtzeitfähige (interaktive) Geschwindigkeiten erreichen (ca. 500 - 1000 ms pro Frame).

Ausgehend von diesen - ernüchternden - Ergebnissen wurde in dieser Arbeit - unter dem Aspekt, dass die Software auf einem Standard PC lauffähig und der Großteil der Rechenzeit für die Instrumentensimulation aufgewendet werden soll, ein anderer, heuristischer, Weg gewählt, um hohe Performance mit einer ausreichend überzeugenden Simulation zu vereinen.

Entstanden ist ein Verfahren, das bei oberflächlicher (rein visueller) Betrachtung den Ansprüchen der Realität genügt, jedoch nur einen kleinen Bruchteil der Rechenzeit benötigt, die für eine wirklich physikalisch korrekte Berechnung nötig wäre. Die Grundidee zellulärer Automaten wird mit einer pseudophysikalischen Simulation von Partikeln verbunden.

¹⁰Aktuelle (2007) Implementierungen sind in der Lage ein vergleichbares SPH System bei *vollständig fehlender Kollisionserkennung/-behandlung* auf der GPU in 10-20 ms pro Rechenschritt zu lösen. Allerdings werden diese Werte nur bei Vollauslastung des Rechners erreicht, was für die vorgestellte Simulation keine Option darstellt. [139]

5.5.1 Zelluläre Baumethode

Die Grundidee bei zellulären Automaten besteht darin, die zu simulierende Umgebung in (in der Regel) gleichförmige Zellen zu unterteilen und die Ausbreitung der Dichte über einen Satz von Regeln zu definieren. Das Prinzip folgt Conways 1970 von Gardner vorgestellten Game of Life [80]. Bei der Umsetzung in der Simulation werden Gefäßsegmente (der Einfachheit halber werden die gleichen Segmente wie in der Instrumentensimulation verwendet) als Zellen in einem Netzwerk betrachtet. Die Begriffe Zelle und Gefäßsegmente werden im folgenden wechselhaft mit identischer Bedeutung verwendet.

Jede Zelle verfügt über eine Anzahl von Eigenschaften (Länge $l(S, t)$, durchschnittlicher Durchmesser $d(S, t)$, Volumen $V(S, t)$, Druck $\rho(S, t)$), die als Funktion in Abhängigkeit von einem Zeitpunkt t gegeben sind.

Über diese Zellen wird Kontrastmittel in Form von Partikeln verteilt. Seien i Partikel im System vorhanden - dann verfügt jedes Partikel p_i über eine Geschwindigkeit $v_i(t)$, eine Beschleunigung $a_i(t)$ und eine Energie $E_i(t)$. Die Energie wird als streng monoton sinkende Funktion allein über das Alter eines Partikels bestimmt und kontrolliert während der graphischen Darstellung die Dichte des Partikels (siehe Kapitel 6.4). Sinkt die Dichte auf einen Wert < 0 , wird das Partikel gelöscht und nicht weiter simuliert.

Partikel bewegen sich zwischen den Zellen, indem die Druckunterschiede zwischen benachbarten Gefäßsegmenten als treibende Kraft auf die gerichtete Beschleunigung eines Partikels wirken. Der zu Grunde liegende Algorithmus wird aktiv, wenn ein Partikel eine neue Zelle erreicht. Ausgehend von einer laminaren Strömungsdynamik wird zu diesem Zeitpunkt vereinfachend einmalig ein Zielsegment für das Partikel berechnet.

Hierfür wird für jedes Partikel i , das eine neue Zelle betritt der Druck der erreichten Zelle S mit dem Druck der benachbarten Zellen verglichen und somit nach Formel 5.30 eine Wahrscheinlichkeit p_j für jedes Nachbarsegment mit abfallendem Druck S_j bestimmt.

$$p_j = \begin{cases} \frac{\rho(S) - \rho(S_j)}{\rho}, & \text{wenn } \rho(S_j) < \rho(S) \\ 0, & \text{sonst} \end{cases} \quad (5.30)$$

Dabei bezeichnet ρ die akkumulierte Summe aller negativen Druckunterschiede

$$\rho = \sum_j \rho(S) - \rho(S_j) \quad (5.31)$$

Abhängig von der so errechneten Wahrscheinlichkeit wird das Partikel bis zum Erreichen des nächsten Segments in die Richtung des gewählten Zielsegments mit druckabhängiger Kraft beschleunigt.

Die eigentliche Bewegungssimulation der Partikel geschieht über Verlet Integration. Sei x_i die Koordinate des Partikels i , v_i die entsprechende Geschwindigkeit und a_i die errechnete Beschleunigung. Dann gilt:

$$x_i(t + \delta) = (1 + F_f) * x_i(t) - F_f * x_i(t - \delta_t) + a_i * \delta_t * \delta_t \quad (5.32)$$

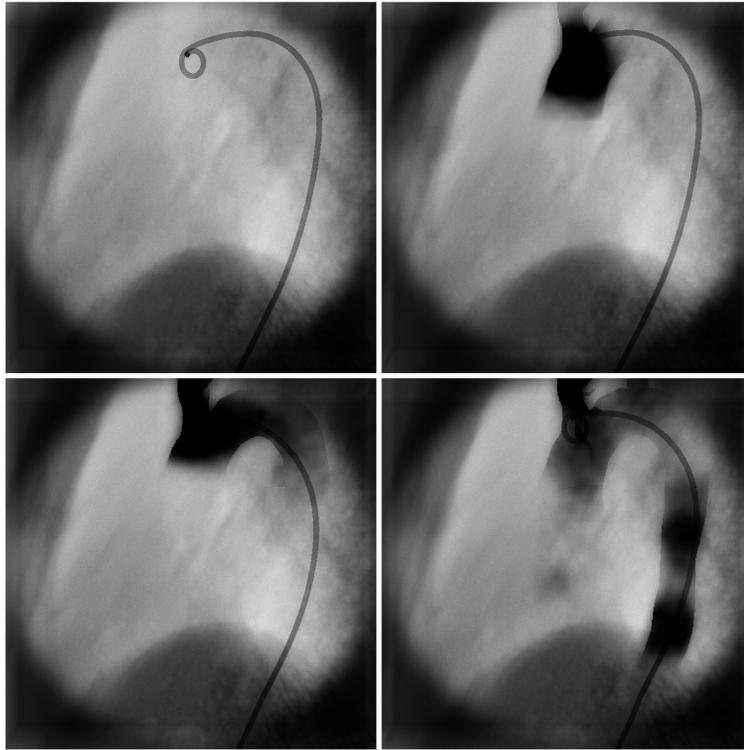


Abbildung 5.34: Verlauf des applizierten Kontrastmittels

F_f sei dabei eine dämpfende Konstante. Für die Geschwindigkeit gilt:

$$v_i(t + \delta) = \frac{x_i(t + \delta) - x(t)}{\delta} \quad (5.33)$$

Desweiteren, muss dafür gesorgt werden, dass die Partikel die Segmente nicht über Ihre Wände hinweg verlassen. Zu diesem Zweck wird nach jedem Simulationsschritt eine Kollisionserkennung zwischen dem zurückgelegten Weg des Partikels und den triangulierten Gefäßwänden durchgeführt. Wird eine Kollision festgestellt, wird eine elastische Reflektion am Auftreffpunkt berechnet (Billardkugeleffekt) und das Partikel entsprechend abgelenkt.

Ausgehend von diesem Regelwerk lässt sich eine visuell überzeugende Simulation von Kontrastmittel durchführen. Eine Darstellung des auf diese Weise simulierten Kontrastmittelverlaufs zeigt Abbildung 5.34. Weitere (auch in der Abbildung) verwendete graphische Effekte, um die Darstellungsqualität zu verbessern und einfache Wirbel zu simulieren, werden in Kapitel 6.4 ausführlicher dargestellt.

5.6 Resultate der Kontrastmittelsimulation

Wie auch die Instrumentensimulation, so muss auch die Simulation des Kontrastmittels unter den Gesichtspunkten einer vollständigen Simulationsumgebung bewertet werden. Der vorgestellte Algorithmus dient nicht dem Zweck, ein physikalisch exaktes

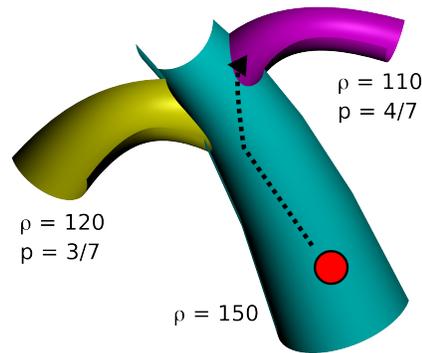


Abbildung 5.35: Schematische Darstellung der Partikelbewegung. Partikel werden durch Reflexion abgelenkt und druckabhängig in Richtung des stochastisch ermittelten Zielsegments beschleunigt.

Kontrastmittelverhalten nachzubilden. - Entscheidend ist die Auffassung der Kardiologen, die ein für den Zweck der Simulation ausreichend realistisches Verhalten bestätigt haben. Kontrastmittel fließt die gleichen Wege wie in der Realität und visualisiert Gefäße in einer Weise, wie sie auch in der Realität vorkommt.

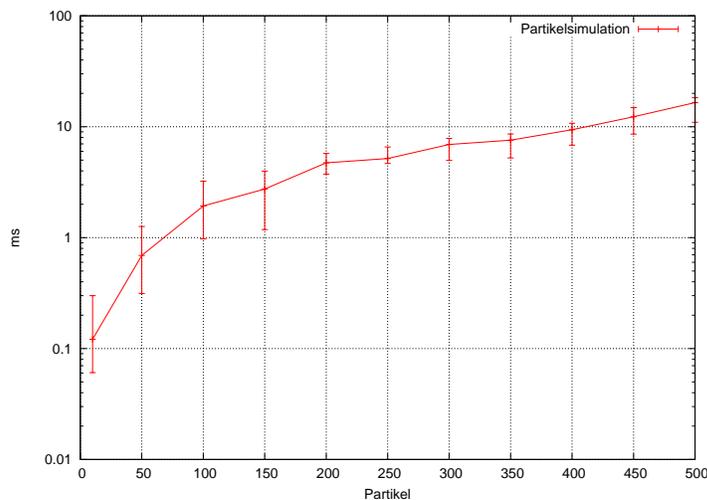


Abbildung 5.36: Rechenzeit im Zusammenhang mit Anzahl der Partikel

Der große Vorteil des vorgestellten Verfahrens ist jedoch die nahezu zu vernachlässigende Geschwindigkeit. Diagramm 5.36 zeigt die Rechenzeit in Abhängigkeit der verwendeten Partikel.

Die Messungen erfolgten während des vollen Simulationsbetriebs - die parallel laufende Instrumentenberechnung und Zeichenalgorithmen wurden bewusst nicht ab-

geschaltet um realistische Ergebnisse im Gesamtkontext darzustellen.

Eine Untersuchung des Algorithmus ließe eine lineare Rechenzeit erwarten, was die Messungen auch bestätigen. Die Komplexität des Verfahrens läuft in $O(n)$ (wobei n die Anzahl der Partikel darstellt). Allerdings ist die eigentliche Rechenzeit in jedem Iterationsschritt weiteren Faktoren wie etwa der Anzahl der umgebenden Dreiecke (Komplexität der Kollisionsdetektion steigt) unterworfen.

Selbst bei vielen hundert Partikeln ist noch kein nennenswerter Einfluss auf die Rechenzeit spürbar. Da Kontrastmittel mit dem gezeigten Verfahren schon mit 100-200 Partikeln sehr schön visuell darstellbar ist und 500 Partikel ausreichen, um einen Großteil des Gefäßsystems vollständig zu schwärzen ist die Performanz mehr als ausreichend.

Das Ziel, einen Kontrastmittelalgorithmus zu finden, der trotz komplexer Kollisionsdetektion so gut wie keine Ansprüche an die Rechenressourcen stellt und doch den Anforderungen an eine optisch realistische Simulationsumgebung genügt wurde also erreicht.

Kapitel 6

Graphische Realisierung

Die graphische Simulation verfolgt vor allem zwei Ziele. Sie soll performant sein - ausreichend performant, um nicht zu viel Rechenzeit auf der CPU zu beanspruchen und eine stabile Bildrate von mindestens 20 Bildern pro Sekunde zu gewährleisten. Auf der anderen Seite soll sie der Realität nahekommen - computergenerierte Bilder sollen so überzeugend sein, dass sie dem Nutzer den Eindruck vermitteln, einen realen Eingriff durchzuführen.

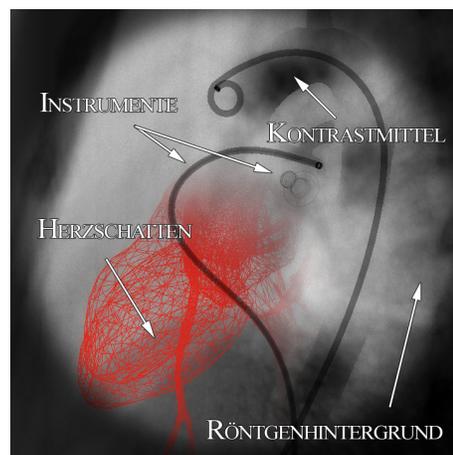


Abbildung 6.1: Darstellung der graphischen Elemente. Das hier dargestellte Bild überlagert mehrere Einzelbilder, um alle Elemente in einer einzelnen Grafik zusammenzufassen.

Die Generierung der Graphiken teilt sich grob in fünf Bereiche auf: Als Grundlage liegt die Simulation der Röntgenkamera/-geometrie. Hier werden Projektions- und Modelviewmatrizen berechnet mit Hilfe derer die Graphiken gerendert werden.

Ein weiterer Bereich ist die Visualisierung des Röntgenhintergrundes. Hier wird aus computertomographischen Volumendaten eine Projektion berechnet und durch den Einsatz von Shadern auf der Grafikkarte ein röntgenähnlicher Effekt erzeugt.

Der dritte Abschnitt umfasst die Darstellung der Gefäße bzw. des bewegten Herzschatte aus den geometrischen (Dreiecksnetze) Patientendatensätzen.

Der vierte Bereich behandelt das Rendering des Kontrastmittels. Partikel müssen gezeichnet und korrekt geblendet werden. Das Verfahren und verwendete Effekte (beispielsweise zur Generierung von Wirbeln) werden vorgestellt.

Im abschließenden fünften Teil wird das Rendering der Instrumente demonstriert. Aus der Gelenksdarstellung (siehe Kapitel 5.1) werden mittels einer Parallel Transport Frame Technik in Echtzeit Dreiecksnetze generiert und anschließend dargestellt.

6.1 Kameramatrizen

Röntgenbilder des Patienten werden in der Kardiologie durch eine auf einem C-Arm (oder C-Bogen) montierte Röntgenkamera (bzw. -quelle und -detektor) aufgenommen (siehe Abbildung 6.2).



Abbildung 6.2: Auf C-Arm montierte Röntgenanlage. Kamera und Detektor können frei um den Patienten rotiert und geneigt werden.

Mit Hilfe dieser Montierung ist es möglich, den C-Arm um die Körperachse des Patienten zu rotieren (Rotation) und Kopf- oder steißwärts zu neigen (Angulation). Ebenso kann man den Tisch auf dem der Patient liegt in allen drei Hauptachsen frei verschieben.

Die Kameramatrix, mit denen 3D Koordinaten multipliziert werden, um als 2D Koordinaten auf dem Bildschirm dargestellt werden zu können, werden (gemäß den Konventionen der gängigen Computergeometrie APIs (OpenGL, Direct3D)) in Modelview- und Projektionsmatrix aufgeteilt. Die Modelview Matrix übernimmt dabei die Aufgaben der Rotation, Translation und Skalierung (ggf. auch Scherung), während die Projektionsmatrix die Eigenschaften der Kamera (Öffnungswinkel, Seitenverhältnis) abbildet.

6.1.1 Modelview Matrix

Die Modelview Matrix der Kamera muss im Falle der kardiologischen Simulation in der Lage sein, Rotation, Angulation und Translation in allen Achsen abzubilden. Zur Vereinfachung gehen die folgenden Definitionen von einem einfach definierten euklidischen Koordinatensystem aus. Die Rotationsachse wird durch die X- (Patienten-) und die Angulationsachse bei nicht vorhandener Rotation durch die Y-Achse repräsen-

tiert. Andere Definitionen sind möglich, erfordern aber ggf. weitere Vorberechnungen und Achsentransformationen¹.

Die verwendete Matrix resultiert demnach aus der Multiplikation mehrerer Einzelmatrizen - der Translationsmatrix M_{t1} , die einer Verschiebung der Kamera in ihren Rotationsmittelpunkt (in der Regel das Herzzentrum im Patientendatensatz) entspricht, der Translationsmatrix M_{t2} , die die Translation des Tisches repräsentiert sowie der beiden Rotationsmatrizen M_r und M_a , die Rotation und Angulation beschreiben.

Dabei ist M_r eine normale Rotationsmatrix um die X Achse (Körperachse) des Patienten und M_a eine Rotationsmatrix, die die zum Körper orthogonal in der Ebene liegende und mit M_r multiplizierte Achse als Rotationsachse verwendet.

Die zusammengefasste Rotationsmatrix M_R lässt sich dann schreiben als:

$$M_R(\alpha, \beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta & 0 \\ \sin \beta \sin \alpha & \cos \alpha & -\sin \alpha \cos \beta & 0 \\ -\sin \beta \cos \alpha & \sin \alpha & \cos \beta \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.1)$$

Dabei bezeichnet α den Rotationswinkel und β den Angulationswinkel. Die Translationsmatrizen M_{t1+2} entsprechen den Angaben in der gängigen Literatur.

Die vollständige Modelviewmatrix kann also beschrieben werden durch:

$$M_{mv} = M_{t1} * M_R * (-M_{t1}) * M_{t2} \quad (6.2)$$

6.1.2 Projektionsmatrix

Um verschiedene Vergrößerungsfaktoren zu erreichen, kann bei der Angiographie der Abstand zwischen Röntgenquelle und -detektor verändert werden sowie die Größe des Detektors beschränkt werden. Diese Änderungen sind äquivalent zur Veränderung des Öffnungswinkels (der Brennweite) bei einer konventionellen Kamera, weshalb die Projektionsmatrix folgendermaßen beschrieben werden kann:

$$M_{pr} = \begin{pmatrix} \frac{1}{\tan(\text{atan2}(\frac{w}{2}, d))} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(\text{atan2}(\frac{w}{2}, d))} & 0 & 0 \\ 0 & \frac{Z_f + Z_n}{Z_n - Z_f} & \frac{2 * Z_f * Z_n}{Z_n - Z_f} & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad (6.3)$$

w bezeichnet dabei die aktive Breite des Röntgendetektors und d den Abstand zwischen Quelle und Detektor. Die vollständige Kameramatrix wird durch Multiplikation von 6.2 und 6.3 beschrieben durch:

$$M = M_{mv} * M_{pr} \quad (6.4)$$

¹Die Simulationssoftware unterstützt parametrisierbar beliebige Achskonventionen

6.2 Volumenrendering

Um einen überzeugenden Eindruck einer Angiographie zu gewährleisten, ist die Abbildung eines realistischen Röntgenhintergrundes nötig. Dafür gibt es verschiedene Techniken - mehrere kommerzielle Systemhersteller (Symbionix, Mentis) verwenden animierte Polygondaten der wichtigsten Strukturen des Brustkorbs (Knochen, Lunge, Leber...) und berechnen daraus mit Hilfe diverser Shader (hauptsächlich Weichzeichnung und Transferfunktionen im fertigen Bild) Hintergründe, die denen eines echten Röntgenbildes ähneln. Der Nachteil dieser Technik ist, dass Hintergründe zu klar wirken - zu strukturiert und sauber auf die vorhandenen Daten beschränkt. Verschiedene Gewebsstrukturen- und Dichten kommen kaum zum Tragen.

Eine andere Technik verwendet der amerikanische Hersteller SimSuite. Hier werden im Vorfeld aufgenommene, echte 2D Röntgenfilme als Hintergrund projiziert. Die Ergebnisse sind natürlich entsprechend realistisch - sind aber nicht den Anforderungen einer Echtzeitsimulation gewachsen. Selbst wenn man die Angulation und Rotation auf jeweils nur 60 feste Winkeleinstellungen diskretisierte (kontinuierliche Winkel zwischen -80 und +80 Grad sind in der Kinderkardiologie nicht unüblich) und Skalierungen und Verschiebungen vernachlässigte wären 3600 Röntgenfilme nötig, um einen realistischen Eindruck zu gewinnen. SimSuite beschränkt daher die Möglichkeiten zur Winkeleinstellung auf drei vordefinierte Ansichten.

In dieser Arbeit wurde daher ein anderer Weg gewählt, um realistische Röntgenhintergründe zu berechnen. Daten aus computertomographischen Aufnahmen bilden die Grundlage für ein Volumenrendering auf der GPU, das Dichten über eine in Echtzeit veränderbare Transferfunktion abbildet.

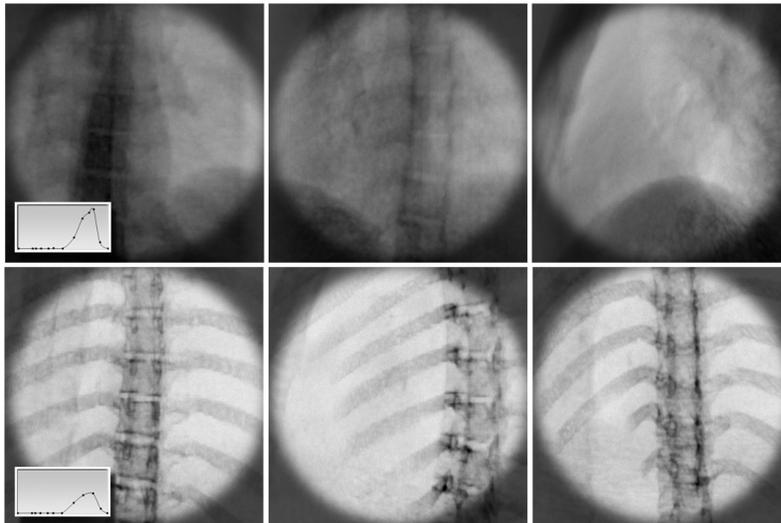


Abbildung 6.3: Röntgenhintergrund als Volumengrafik in verschiedenen Projektionen und Transferfunktionen.

Zu diesem Zweck wurde der komplette Volumendatensatz (verkleinert auf eine Auflösung von $256 \times 256 \times 256$ Pixeln) als 3D Textur in den Speicher der Grafikkarte geladen. Anschließend kann der Datensatz in Echtzeit über eine aus der Modelview-

matrix abzuleitende Texturmatrix (siehe unten) auf zum Betrachter senkrecht stehende (und auf Grundlage der Modelviewmatrix translatierte) Ebenen projiziert (siehe Abbildung 6.4) werden. Der Vorteil dieses als 3D Textur Mapping bekannten Verfahrens ist, dass die Vorzüge der Grafikkarte voll ausgenutzt werden und somit angemessen schnelle (interaktive) Rechenzeiten erzielt werden können.

Die Herleitung der Textur- und Translationsmatrix geschieht folgendermaßen. Die Texturmatrix muss die Rotation der Volumendaten abbilden - Informationen die in der Modelviewmatrix in den ersten drei Spalten und Zeilen enthalten sind. Die vierte Spalte enthält (neben dem Homogenitätswert in (3, 3)) die Translation. Um die Rotationsmatrix M_{rot} zu erhalten, muss also die Teilmatrix $M_{mv}^{0-2,0-2}$ zu einer homogenen Matrix erweitert werden.



Abbildung 6.4: Schematische Darstellung eines 3D Texture Mappings

Da der Rotationsmittelpunkt der Texturmatrix bei $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ liegt, muss die Rotationsmatrix noch mit der Translationsmatrix M_{t3} und ihrem negativen Gegenstück multipliziert werden. Zudem ist es nötig, da die Rotation der Textur gegenläufig zur Rotation der Kamera ist, die Rotationsmatrix zu invertieren.

$$M_{t3} = \begin{pmatrix} 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.5)$$

$$M_{tex} = M_{t3} * M_{rot}^{-1} * (-M_{t3}) \quad (6.6)$$

Die Translationsmatrix erhält man auf vergleichbare Weise durch Erweiterung der Teilmatrix $M_{mv}^{3,0-3}$ zur homogenen Matrix.

Da Röntgentechnik vom Prinzip her ein durchleuchtendes Verfahren ist, ist für die 3D API ein additives Blending zu wählen. Die Dichte einzelner Segmente wird durch eine Transferfunktion bestimmt, die im Falle der vorliegenden Arbeit direkt auf der Grafikkarte als Shader implementiert wurde (Ein vom Benutzer festgelegter Spline wird äquidistant abgetastet und als 1D Textur auf die Grafikkarte geladen, wo er als in Echtzeit veränderbare Transferfunktion dient - siehe Abbildung 6.3).

Um die Röntgenbilder noch realistischer erscheinen zu lassen, wird die Bildgebungskette einer echten Röntgenanlage teilweise nachimplementiert. In der Realität ist es üblich, Röntgenbilder über einen Kantendetektionsalgorithmus abzutasten und anschließend das Kantenbild über das Originalbild zu blenden. Um dieses Verfahren nachzubilden wurde auf Shaderebene ein Sobelfilter (Siehe Formeln 6.7 - 6.9. Darin

sei A das Originalbild, das sukzessive mit dem horizontalen (S_x) und vertikalen (S_y) Sobeloperator gefaltet wird) implementiert, dessen Resultate auf das fertig gerechnete Bild aufgeblendet werden. Ergebnisse sind in Abbildung 6.5 zu sehen.

$$G_x = S_x * A = \frac{1}{8} \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * A \quad (6.7)$$

$$G_y = S_y * A = \frac{1}{8} \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * A \quad (6.8)$$

$$G = \sqrt{G_x^2 + G_y^2} \quad (6.9)$$

Auch wenn sich der Hintergrund auf einer modernen Grafikkarte in ansprechender Qualität in ca. 10 - 20 ms berechnen lässt, sind weitere Optimierungen möglich und (um Rechenzeit für andere Prozesse zur Verfügung zu stellen) auch nötig. Aus diesem Grund wird der Hintergrund nur bei Veränderungen der Projektionswinkel oder Tischbewegungen neu gerechnet und anschließend direkt auf der Grafikkarte gespeichert. Das Zeichnen des auf diese Weise einmalig pro Winkeländerung/Tischbewegung berechneten Hintergrund kann anschließend in knapp 1 ms erfolgen.

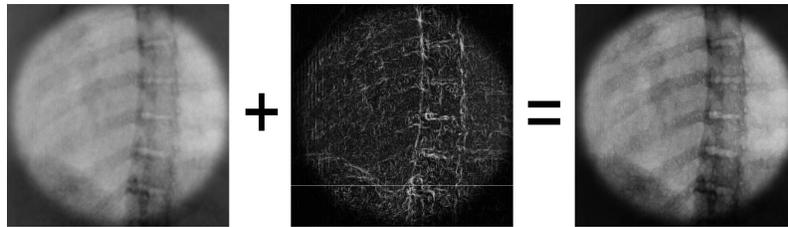


Abbildung 6.5: Auswirkungen der geblendeten Kantendetektion auf das gerenderte Hintergrundbild. Kanten werden schärfer, knöcherne Strukturen treten hervor.

Die Vorteile gegenüber den anderen Verfahren sind die Darstellbarkeit beliebiger Projektionsrichtungen und ein realistisches Bild, das von den Polygondaten verarbeitenden Techniken bisher nicht erreicht werden kann. Der Nachteil, dass die Bilder nicht animiert sind (bsp. Atmung), fällt in der Kardiologie nicht weiter ins Gewicht, da im Realfall Patienten während der Aufnahme gebeten werden, den Atem anzuhalten.

In der Kinderkardiologie, wo das in der Regel nicht möglich ist, bietet der überlagerte Herzschatten (siehe Kapitel 6.3) genug Animation, um ein realistisches Bild zu vermitteln. Wäre eine Animierung der Atembewegungen nötig, so lässt sich diese in akzeptabler Rechenzeit (zusätzlich ca. 15 ms pro Frame) über eine FFD auf den (für diesen Fall in Dreiecksnetze unterteilten) Projektionsebenen berechnen.

6.3 Herzschaten

Während einer kardiologischen Intervention orientieren sich Ärzte häufig an bekannten Strukturen. Neben dem Zwerchfell und den Rippenbögen kommt hier vor allem der Herzschaten (die Projektion der stark durchbluteten und vom Pericard umschlossenen Hohlräume) zum tragen. Der Herzschaten ist in der Röntgensicht vor allem durch seine starke Bewegung auszumachen, die dem Kardiologen gleichzeitig einen Rückschluss auf die tatsächliche Herzaktion liefert.

Aufgrund dessen, dass die Bewegung des Herzschatens aktiv von der Intervention beeinflusst werden kann (bsp. durch Komplikationen) und sich der Ort des Pericards je nach Patientendatensatz ändert - kann er graphisch nicht durch die oben geschilderte Volumentechnik dargestellt werden. Abgesehen davon, dass animierte Volumendatensätze (selbst bei der kurz beschriebenen Verwendung der FFD auf den Projektionsebenen) recht rechenzeitaufwändig sind, findet die Bewegung losgelöst von der Hintergrundgrafik statt, was eine weitere Berechnung nötig machen würde.

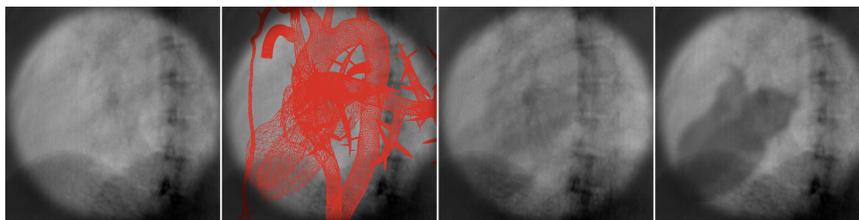


Abbildung 6.6: Generierung des Herzschatens. Von links nach rechts zeigen die Bilder den Röntgenhintergrund ohne Herzschaten, das verwendete Gittermodell, den berechneten Herzschaten in realistischer Deckkraft und den Herzschaten mit (zur besseren Visualisierung) erhöhter Deckkraft.

Stattdessen wird in der vorgestellten Simulation ein realistisch aussehender Herzschaten aus den (ohnehin vorhandenen) Polygondatensätzen erstellt. Zu Beginn jedes Bildes werden die Vertices der Polygondatensätze per FFD (siehe Kapitel 5.4) in einen festen Zustand transformiert. Anschließend werden die Polygondaten auf eine niedrig aufgelöste (256×256) Textur gerendert. Jedes Segment ist dafür mit einem eigenen Opazitätswert versehen, der als (einheitslose) Röntgendichte interpretiert werden kann. Um das unscharfe Erscheinungsbild des Herzschatens zu gewährleisten, wird die Textur anschließend (ggf. in mehreren Renderpasses) mit einem Gaußkernel (als Shaderimplementierung) gefaltet.

Abbildung 6.6 zeigt die Auswirkungen und das Rendering des Herzschatens. In der rechten Abbildung wurde der Herzschaten zur Verdeutlichung mit höherer Dichte gezeichnet. In der Realität ist er eher in der Dichte des dritten Bildes zu sehen. Deutlich ist auch zu erkennen, dass nur einige Segmente schattiert werden - andere (bsp. die Aorta) verfügen über eine geringere bis gar keine Opazität und werden daher nicht in der Projektion gezeichnet.

6.4 Kontrastmittel

Das Kontrastmittel wird in der Simulation auf Basis eines Partikelmodells gerechnet. Dementsprechend müssen für jedes Bild viele Partikel gezeichnet werden, die jedoch nicht als einzelne Punkte erscheinen dürfen. Das Rendering bedient sich daher einiger Tricks um den Eindruck einer fließenden, zusammenhängenden Substanz mit Verwirbelungen zu erzeugen.

Jedes Partikel wird nicht als Punkt sondern als dem Betrachter zugewandte Textur gezeichnet (Billboarding). Dafür wird jedem Partikel zu Beginn seines Lebenszyklus eine Textur aus einer im Vorfeld durch Perlin Rauschen erzeugten und anschließend (zur Generierung nahezu runder Elemente) mit einem Gaußkernel gefensterten Gruppe von Texturen zugeordnet, die es bis zum Ende seines Lebenszyklus beibehält. Abbildung 6.7 zeigt eine solche Gruppe von 256 Texturen.



Abbildung 6.7: Mögliche Texturen für das Partikelsystem. Texturen wurden über Perlin Rauschen erzeugt und einzelne Elemente mit einem Gaußkernel gefiltert um verschiedene Texturen für Partikel zu erhalten.

Die Partikel selbst werden bei Entstehung mit einem Durchmesser von ca. 4 mm gezeichnet - vergrößern sich im Laufe der Zeit aber linear auf bis zu 2 cm, was den Effekt des sich ausbreitenden, fließenden Kontrastmittels verstärkt. Gleichzeitig nimmt die Dichte des Partikels ab, was zu einem langsam Ausblenden führt. Eine Rotation der Texturen führt zum Eindruck von Verwirbelungen.

Ein Problem, das sich durch dieses Verfahren ergibt, ist das Entstehen von Randüberlagerungen. Partikel, die am Rand einer Gefäßstruktur liegen und mit der vollen Größe gezeichnet werden, schwärzen den Bereich außerhalb des Gefäßes (Siehe Abbildung 6.8). Um Abhilfe zu schaffen, wird der Stencilbuffer genutzt. Da jedes Partikel eindeutig einer Struktur zugeordnet werden kann, wird aufeinanderfolgend jede Struktur (gemeinsam mit den direkten Nachbarstrukturen) in einen leeren Stencilbuffer gezeichnet und erst anschließend das Partikel (unter Ausnutzung der nun vorhandenen

Schablone) übertragen. Ein solches Vorgehen ist nötig, da - würden alle Strukturen gleichzeitig im Stencilbuffer gehalten - ein Überfließen der Partikel in nicht direkt benachbarte Strukturen möglich wäre (beispielsweise könnte ein Partikel im Bereich der Aorta unerwünschterweise Segmente aus der naheliegenden Pulmonararterie schwärzen).

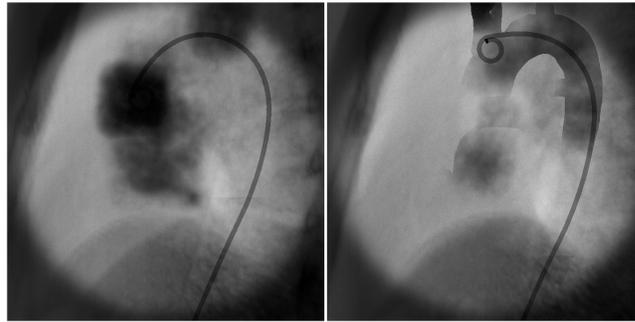


Abbildung 6.8: Auswirkung des Stencil Buffers auf die Partikelgrafiken. Randüberlagerungen verschwinden und klare Gefäßstrukturen zeichnen sich ab.

6.5 Instrumente

Instrumente sind in der Simulation durch Gelenke mit Ausrichtung und Durchmesser definiert. Zum Zeichnen werden zudem Informationen über die Dichte (Röntgendichte) und den Modus (ausgefüllt oder Gitter) an jedem Punkt benötigt. Zur Definierung dieser Parameter werden die in Kapitel 5.1 bereits vorgestellten Pakete genutzt. Dadurch ist es möglich, das gesamte Instrument kontinuierlich zu beschreiben.



Abbildung 6.9: Darstellung der Gitterstruktur bei einem Occluder. Katheter und Occluder werden in verschiedenen Modi gezeichnet, um Gitterstrukturen und ausgefüllte Segmente darzustellen.

Um auch die diskreten Gelenkskoordinaten in einen kontinuierlich definierten Rahmen zu übertragen, werden die Gelenkspositionen als Stützpunkte auf einem vollständig interpolierenden (jeden Stützpunkt berührenden) Spline genutzt. Gut geeignet sind hierfür Hermite Splines (im besonderen - als deren Unterklassen - Cardinal- oder Catmull-Rom-Splines), da sie mit nur wenigen Stützstellen auskommen und verhältnismäßig schnell zu berechnen sind. Zudem erfüllen sie die Anforderungen an eine vollständige Interpolierbarkeit und können über den Kardinalfaktor gut an die (in der Regel sehr glatten) Katheterformen angepasst werden.

Reicht es für einfache Drähte möglicherweise bereits aus, die Form des Splines mit einer Linie nachzuziehen, kommt bei dickeren Instrumenten (oder komplex geformten Instrumenten wie Occludern) die dreidimensionale Struktur zum Tragen und die Generierung eines echten 3D Modells wird nötig.

Zu diesem Zweck wird in der Simulation ein 3D Gitter für jedes Modell auf Basis der Parallel Transport Frame Methode generiert (siehe Bild 6.10).

Geben sei eine Kurve C (der Spline), ein existierender Rahmen (bei Instrumenten in der Regel ein Kreis) F_1 sowie eine Tangente T_1 zum Zeitpunkt $t - 1$ und eine Tangente T_2 zum Zeitpunkt t . Dann errechnet sich ein neuer Rahmen T_2 zum Zeitpunkt t wie folgt:

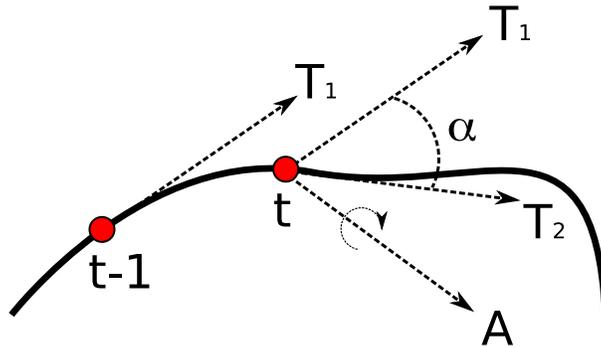


Abbildung 6.10: Skizzierung zur Parallel Transport Frame Technik

Die Position von F_2 ist die Koordinate von C an Stelle t . Die Orientierung von F_2 wird ermittelt indem F_1 mit dem Winkel α um eine Achse A rotiert wird, wobei $A = T_1 \times T_2$ und $\alpha = \arccos\left(\frac{T_1 \cdot T_2}{|T_1| |T_2|}\right)$ (siehe Abbildung 6.10).

Da die äquidistante Generierung der Gitterpunkte nur bei als Gitter dargestellten Instrumenten (Occluder, Stent) von Belang ist, kann auf die Generierung eines neuen Frames verzichtet werden, wenn sich die Dichte des Instrumentes nicht ändert und die Tangenten nahezu parallel sind.

Prinzipiell gibt es einfacherer Methoden um eine röhrenähnliche Struktur in 3D zu erzeugen - beispielsweise den Frenet Rahmen und die Methode des statischen Top Vektors. Beide führen jedoch unter bestimmten Bedingungen zu Diskontinuitäten in der Gitterstruktur - beispielsweise wenn die Tangente des Splines nahezu parallel zum Statischen Top Vektor verläuft (Abb. 6.11) oder die zweite Ableitung einer Kurve im Frenet Verfahren 0 ergibt (Abb. 6.12) - was eine Verwendung in der Simulation (wo der Pfad des Instrumentes nicht beschränkt werden kann) verbietet.

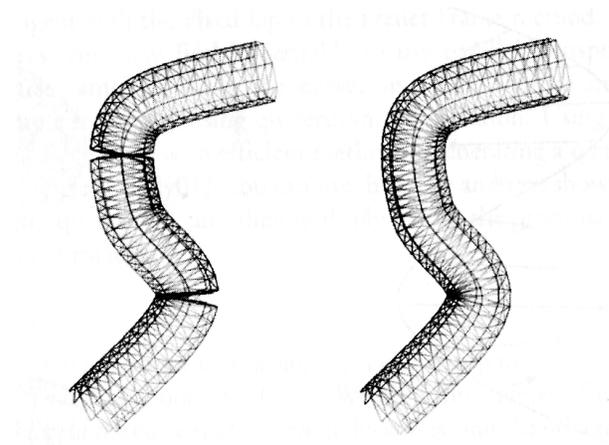


Abbildung 6.11: Berechnung einer Röhre mit Fixed Up- (links) und PTF Methode. Verläuft die Tangente des Splines fast parallel zum statischen Top Vektor entstehen Diskontinuitäten, die mit dem PFT Verfahren vermieden werden. (Bild aus [24])

6.6 Postprocessing

Sind nacheinander alle Einzelbereiche gerendert, werden noch einige Nachbearbeitungsschritte auf das generierte Bild angewandt. So kann beispielsweise nachträglich über Shader der Gamma Faktor verändert, der Kontrast angepasst und das Bild geschärft oder weichgezeichnet werden.

6.6.1 Rauschen

Auch wird dem Bild ein künstliches Rauschen hinzugefügt, das den Echtheitseindruck der Simulation weiter verstärkt. Da es aber im aktuellen Shadermodell 3.0 keine Möglichkeit gibt, Zufallszahlen auf der GPU zu erzeugen (und unter GLSL - der in

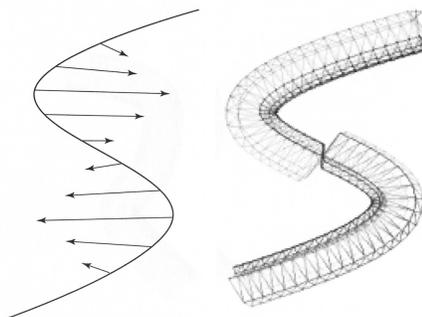


Abbildung 6.12: Darstellung der Diskontinuitäten im Frenet Verfahren. Strebt die zweite Ableitung einer Kurve gegen 0, ergeben sich Unstetigkeiten in der resultierenden Form. (Bild aus [24])

der Simulation verwendeten Shadersprache - kein Zugriff auf logische Operationen (XOR) besteht), muss das Rauschen auf andere Weise generiert und auf die Grafikkarte übertragen werden.

Die Generierung einer Rauschtextur auf der CPU in hoher Auflösung würde (fände sie in jedem Frame statt) zu viel Zeit beanspruchen. Stattdessen wird im Hauptspeicher des Rechners einmalig zum Programmstart ein großer Speicherbereich (4MB) mit Zufallswerten gefüllt. Pro Frame wird dann ein zufällig gewählter Zeiger in den Speicherbereich (ausreichend weit vom Ende entfernt) generiert, der als Beginn einer neuen Rauschtextur interpretiert wird.

Andere Methoden, Rauschen im Shader zu erzeugen (bsp. Wanderndes Rauschens über eine Translation einer statischen Rauschtextur um wenige Koordinaten pro Frame) waren entweder langsamer als die vorgeschlagenen Methode oder konnten visuell nicht überzeugen.

Kapitel 7

Technische Umsetzung

Da diese Arbeit zu einem großen Teil auch eine Machbarkeitsstudie im Sinne einer informationstechnischen Umsetzung bzw. Programmierung darstellt, soll im folgenden Kapitel die technische Umsetzung der Simulation vorgestellt werden. Zu Beginn steht ein kurzer Überblick über die verwendete Programmierumgebung, Libraries, Sprachen und Techniken.

Im weiteren Verlauf wird die verwendete Klassenstruktur, die vor allem im Hinblick auf Flexibilität, Wartbarkeit und Geschwindigkeit entwickelt wurde, vorgestellt und technische Besonderheiten aufgezeigt.

Im dritten Teil werden spezielle Performanceoptimierungen auf technischer Ebene (Threading, Mempooling, SIMD) präsentiert und ihre Effektivität besprochen.

Im abschließenden vierten Abschnitt werden kurz Werkzeuge vorgestellt (bsp. Modelleditoren), die zwar für die Erstellung der Simulation entwickelt wurden, aber nicht Bestandteil des eigentlichen Programmcodes sind. Das beinhaltet auch die Vorstellung der Trackinghardware und deren Anbindung an das Cathi^{Kid} System.

7.1 Überblick

Der komplette Programmcode der vorliegenden Simulation liegt als soweit wie möglich plattformunabhängiger C++ Code vor. Voll lauffähige Kompilate existieren unter Linux (gcc), Apple OS X (gcc) und Microsoft Windows (Visual C++). Soweit vorhanden wird der SSE (SIMD) Instruktionssatz des Prozessors durch die Routinen der eigenen mathematischen und geometrischen Libraries ausgenutzt.

Als Grafikumgebung diente Open GL. Shader wurden sowohl in Cg als auch in GLslang entworfen.

Der komplette Code (107.815 Zeilen, 117 Klassen) wurde - ohne auf eine bestehende Codebasis aufzubauen - von Grund auf neu geschrieben. Als externe Libraries fanden einzig die freien (GPL License) und plattformübergreifenden Open Source Bibliotheken boost [4] (Smartpointer, Threading, Assertions, Error checking, Reguläre Ausdrücke, Funktoren & BLAS¹), glew [1] (GL Extension Wrangler), tinycl [2] (XML Lesen/schreiben), SDL (Initialisierung des OpenGL Kontext) und Boost ASIO (Netzwerk Behandlung) Verwendung.

¹Basic Linear Algebra System

Als Entwicklungsumgebung diente IBM Eclipse. Als Debugger wurde meist gdb verwendet. Die Speicherverwaltung wurden mit Valgrind analysiert, Performance unter Zuhilfenahme der Apple Shark Collection optimiert.

Auch wenn das Programmgerüst keine vollständig flexible Simulationsengine wie etwa SOFA oder Spring darstellt, ist mit etwas Anpassung die Verwendung in anderen Simulationen als Framework denkbar. Die physikalischen Verfahren sind allerdings derzeit auf das spezialisierte Instrumentensystem beschränkt.

7.2 Klassenstruktur

Um die einzelnen Komponenten (Grafik, Physik, Eingabe...) der Simulation geschickt und vor allem auswechselbar miteinander zu verbinden, ist es nötig den Programmcode möglichst modularisiert und unter Vermeidung direkter Abhängigkeiten aufzubauen.

Im Falle der vorliegenden Arbeit wurde das durch die konsequente Anwendung der bekannten Programmierpatterns [79] erzielt. Weitgehend voneinander unabhängige Komponenten der Software wurden in Libraries implementiert und Kommunikationsmöglichkeiten über wohldefinierte Schnittstellen vorgesehen. Abbildung 7.1 zeigt eine Übersicht der 14 Bibliotheken und eine grobe Unterteilung nach ihren Zugriffsmustern.

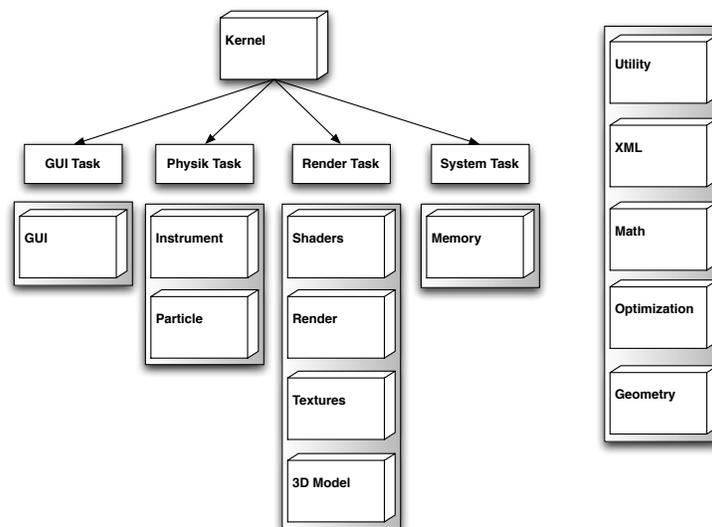


Abbildung 7.1: Übersicht der 14 innerhalb dieser Arbeit entwickelten Bibliotheken und der primär auf sie zugreifenden Tasks.

Desweiteren wurden in der Arbeit moderne Methodiken der C++ Programmierung umgesetzt. Nahezu alle Klassen, von denen Instanzen zur Laufzeit bereitgestellt werden, wurden von einer einzigen Basisklasse abgeleitet, die globale Klassenfunktionalitäten (bsp. zum Speichermanagement mittels Intrinsic Smart Pointern) zur Verfügung stellen kann.

Es folgen Erläuterungen und (teilweise vereinfachte oder gekürzte) Klassendiagramme der einzelnen Subkomponenten. Einige Subkomponenten wurden zur besseren Erläuterung im Folgenden zu gemeinsamen Komponenten zusammengefasst (beispielsweise Optimierung und Mathematik) obwohl die Granularität der Klassenstruktur eine feinere Strukturierung zuließe.

Da eine ausführliche Beschreibung aller Komponenten den Rahmen dieser Arbeit sprengen würde, wird an dieser Stelle nur eine grobe Beschreibung der wichtigsten Bibliotheken und Klassen und einige Besonderheiten im Design beschrieben. Eine ausführlichere Dokumentation wurde im doxygen [6] Standard verfasst und liegt der Firma CATHI GmbH vor.

7.2.1 System & GUI Libraries

Die folgenden Sektionen beschreiben Bibliotheken, die als Framework für systeminterne Verwaltung, sowie GUI genutzt werden und weitgehend von der eigentlichen Simulationsumgebung unabhängig sind.

Kernel Library

Herzstück der Simulation bildet der sogenannte Kernel. Die Hauptschleife des Programms wird hier ausgeführt, periodisch wiederkehrende Funktionalitäten werden zyklisch aufgerufen. Um den Programmablauf möglichst flexibel zu halten, wurde der Kernel nicht als festverdrahtete Endlosschleife implementiert. Er beinhaltet stattdessen eine Liste von Aufgaben (Tasks), die periodisch aufgerufen werden.

Tasks bestehen im Wesentlichen aus drei Teilen. Einer *start* Funktion, die initialisierende Aufgaben übernimmt und den Task in einen lauffähigen Zustand versetzt. Einer *update* Funktion, die die eigentlichen Aufgaben des Tasks aufruft und einer *stop* Funktion, die Komponenten herunterfährt und den Task beendet, bzw. ggf seine zugehörigen Objekte zerstört.

Die Update Funktion enthält als Kernstück im Idealfall nur wenige Funktionsaufrufe und leitet Aufgaben an die zugehörigen Subklassen weiter.

Ist pro Periode eine maximale Dauer festgelegt (in der Kardiologie sind beispielsweise 12.5 Bilder/s, in der Kinderkardiologie bis 50 Bilder/s üblich) und bleibt während der Dauer eines Frames nicht genug Zeit, alle Tasks durchzuführen, so können bei Bedarf (prioritätsgesteuert) Tasks ausgesetzt oder verzögert werden. Unwichtigere Aufgaben (beispielsweise Speichermanagement) können so auf Frames verlegt werden, in denen ausreichend Rechenzeit zur Verfügung steht.

Ebenso besteht die Möglichkeit Tasks dauerhaft oder temporär aus der Liste laufender Aufgaben zu entfernen (bsp. kann für eine Pausierung der Simulation der Task zur Berechnung der Physik und Instrumenteneingabe ausgesetzt werden, während Grafik und Memorymanagement weiter arbeiten).

Tasks werden generell von einer Basisklasse *KernelTask* abgeleitet und sind aufgefordert ihre entsprechenden Funktionen zur Laufzeitsteuerung (*start*, *onSuspend*, *update*, *onResume*, *stop*) zu überschreiben. Ein Klassendiagramm der Kernel Library und ihrer verwendeten Tasks zeigt Abbildung 7.2.

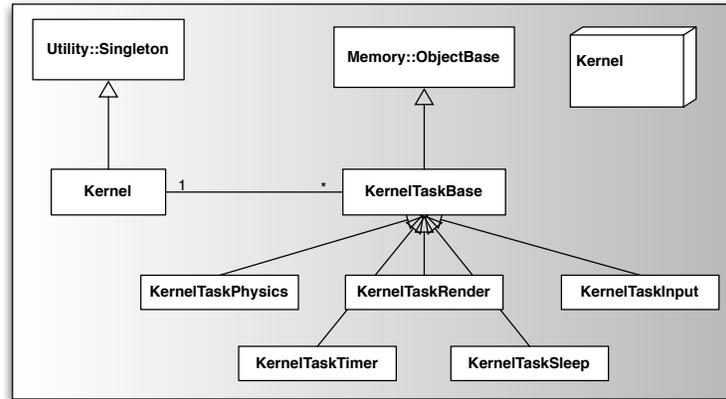


Abbildung 7.2: Klassendiagramm (vereinfacht) der Kernel Library

Speicher Library

Das Speichermodul stellt zwei Komponenten zum Memorymanagement zur Verfügung. Zum Einen existiert eine Managerklasse zur performanten Speicherverwaltung. In dieser werden statische Memmpools mit zugehöriger Speichermap zur schnellen Allokation reserviert und für variabel große Speicherblöcke dynamisch zur Verfügung gestellt.

Der Vorteil einer eigenen Speicherklasse gegenüber der Standardverwaltung des Betriebssystems über `new/delete` besteht neben der ggf. besseren Zugriffsgeschwindigkeit (in der Regel nur bei vorhersagbaren Speicheranfragen fester Größe relevant) vor allem in der besseren Abschätzbarkeit, der zentralen Kontrolle über den verbleibenden Speicher (Programmspeicher kann bereits zu Programmstart reserviert werden) und dem später einfacher handhabbaren Cacheverhalten (Objekte auf die in Reihe zugegriffen wird können leichter generisch in einem festen Speicherblock gehalten werden).

Zudem bietet ein eigenes Speichermanagement einfache Kontrollmechanismen über Programmierfehler wie bsp. Speicherlecks.

Daneben stellt die Managerklasse einen Ringbuffer für schnelle Zugriffe auf lokal benötigte dynamische Speicherbereiche (bsp. Matrixallokierung während der Optimierung) zur Verfügung.

Die zweite Aufgabe der Speicher Library ist die Verwaltung von Smartpointern und damit einer einfachen Garbagecollection. Zur Verwendung der eigenen intelligenten Pointer, muss jedes Objekt, das von der Garbage Collection erfasst werden soll vom Standard Basisobjekt des Frameworks (*MemoryObjectBase*) abgeleitet sein. In diesem Objekt werden die Operatoren `new` und `delete` überschrieben, so dass das Objekt bei herkömmlicher dynamischer Erzeugung entsprechend gezählt werden kann (abgeleitete Objekte, die bei Instanziierung Speicher vom eigenen Manager anfordern sollen, können die Operatoren erneut redefinieren).

Ein Nachteil in der Verwendung eigener Smartpointer besteht darin, dass sie nur mit nicht trivialem Aufwand threadsicher implementiert werden können und ggf. anfällig gegen zyklische Referenzierung und damit einhergehende Speicherlecks sind.

Daher wurde in einer späteren Version des Frameworks die Implementierung dahingehend verändert, dass die Smart & Weak Pointer des 2008 zur Veröffentlichung anstehenden C++ TR2 Standards genutzt werden [4].

GUI Library

Die grafische Bedienoberfläche dient vornehmlich als Experimentierumgebung. Um größtmögliche Plattformunabhängigkeit zu erreichen, wurde ein einfaches GUI System von Hand implementiert, dass die für die Simulation wichtigsten Bedienelemente (Pull Down Menus, Knöpfe, editierbare Splinekurven) auf Grundlage von OpenGL abbildet.

Zudem wurde eine Konsole implementiert, über die mit Shell ähnlichen Kommandos auf die Simulation zugegriffen werden kann. Die Konsole ist parsergesteuert und hat über den (weiter unten beschriebenen) Settingsmanager Zugriff auf alle zur Veränderung/Einsicht freigeschalteten Variablen und Einstellungen. Genau genommen werden alle Bedienungen der Simulation (abgesehen vom direkten Instrumententracking) über Konsolenkommandos und deren Parameter vorgenommen.

Um eine hohe Flexibilität zu gewährleisten, werden Oberflächen als XML Dateien gespeichert (ggf. mit referenzierten Texturen und Texturkoordinaten für graphische Bedienelemente). Schalfächen und/oder Menüs können Konsolenkommandos zugeordnet werden, was eine voll konfigurierbare Oberfläche mit der Bedienung über den Konsolenkommandos identischen Möglichkeiten erlaubt. Abbildung 7.3 zeigt die Gegenüberstellung verschiedener über XML definierter Oberflächen und der eingebundenen Konsole.



Abbildung 7.3: Gegenüberstellung verschiedener Bedienoberflächen und der Konsole. Die Oberflächen sind vollständig über XML Dateien definiert.

Um das bei GUIs häufig auftretende Problem der starken Abhängigkeiten zu lösen, können Gui Elementen OpenGL Renderkontexte zur Verfügung gestellt werden - eine von der Basisklasse der Konsole abzuleitende Klasse ist für die Implementierung anwendungsabhängiger, spezialisierter Kommandos zuständig

7.2.2 Physik & Simulation

Die in dieser Sektion vorgestellten Libraries, werden für die Simulation der Physik genutzt. Instrumente und Partikel stellen den Kern dieser Simulationen dar - greifen aber auf verschiedenste Arten von 3D Modellen zurück, deren verwaltende Bibliothek daher ebenso in diesen Abschnitt aufgenommen wurde.

Als Unterschied zu Open Source Frameworks zur physikalischen Simulation sind Instrumentenmodelle in der Simulation (abgesehen von der gemeinsamen Objektbasis-klasse) nicht mit Gewebsmodellen verwandt. Die Architektur unterscheidet sich darin insofern maßgeblich von Architekturen wie SOFA oder GiPSi, als dass nicht alle physikalisch deformierbaren Objekte als ebenbürtig behandelt werden.

Der Vorteil durch die Trennung ist im konkreten Fall dadurch gegeben, dass Schnittstellen klarer (wenn auch weniger flexibel) gestaltet werden können und eine Kompatibilität zum bestehenden Cathi Framework gewährleistet werden kann. Würde eine physikalische Deformation des Gewebsmodells aber in Zukunft angestrebt werden, sollte entsprechend ein physikalisches Basisobjektmodell in Anlehnung an das Sofa Framework verwendet werden.

Instrumenten Library

Die Instrumenten Bibliothek wurde mit besonderem Bedacht auf spätere Erweiterbarkeit und Austauschbarkeit entwickelt. Da ein Großteil der Arbeit am Simulator in die Entwicklung der Instrumentensimulation geflossen ist, sollten Veränderungen am Algorithmus ohne großen architektonischen Aufwand in die Testumgebung eingebracht werden können.

Aus diesem Grund wird die parametrische Darstellung der Instrumente (wie sie in den Datenblättern der Hersteller veröffentlicht wird), Durchmesser, Längen, Röntgendichte - strikt vom eigentlichen Simulationscode getrennt verwaltet. Ebenso sollten bildgebende Algorithmen (das Zeichnen der Instrumente) keine direkte Verbindung zur physikalischen Berechnung benötigen.

Das Design sieht daher vor, dass berechnende Klassen von den darstellenden und verwaltenden Klassen (im Folgenden als Modellklassen bezeichnet) durch eine Proxy-Klasse abgeschirmt werden (siehe Abbildung 7.4).

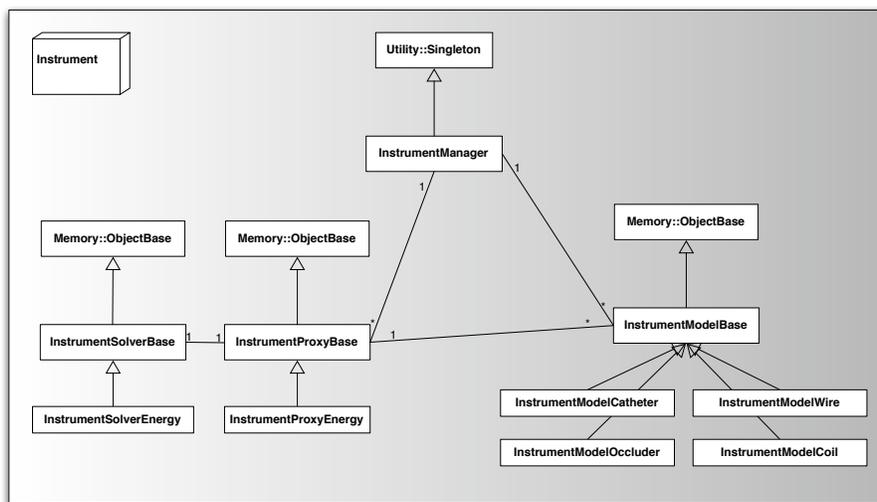


Abbildung 7.4: Klassendiagramm (vereinfacht) der Instrumenten Library

Da (siehe Kapitel 5.1) ineinander verschachtelte Instrumente (energetisch) als ein-

zernes Instrument betrachtet werden, übernimmt die Proxyklasse auch die Integration der einzelnen Instrumentenmodelle und übergibt nur die zusammengefassten Daten an die berechnenden Klassen (Solvers). Das von der Berechnung als geometrische Struktur (i.d.R. ein dreidimensionaler Spline) zurückgelieferte Ergebnis wird von der Proxyklasse der Länge nach aufgeschlüsselt und an die Modellklassen übertragen.

Spezialformen von Instrumenten mit besonderen Eigenschaften können als Ableitung der Basismodellklasse realisiert werden, da in der Regel keine zusätzlich zu simulierenden physikalischen Effekte entstehen. Der Occluder stellt beispielsweise besondere Funktionen zur Berechnung des Durchmessers eines geöffneten Schirmchens zur Verfügung (Splineprofil) - wird aber ansonsten wie oben beschrieben mit den anderen Instrumenten durch die Proxyklasse integriert. Auch die Zeichenalgorithmen können so voneinander abweichen, ohne die Simulation der Physik zu beeinflussen.

Der Zugriff auf die Instrumente und deren Instanziierung durch außerhalb liegende Klassen geschieht über eine Managerklasse, die gleichzeitig als Factory Pattern fungiert. Desweiteren werden im Manager die verschiedenen Proxies (jeder Instrumentenstrang verfügt über ein eigenes Proxyobjekt) und die Instrumentenmodelle verwaltet. Aus API Sicht ist allerdings kein direkter Zugriff auf Proxies oder berechnende Modelle vorgesehen. Ein Austausch der Klassen im Sinne einer Polymorphie ist zwar möglich - die Manipulation der Instrumente wird aber nur über die Modellklassen erreicht.

Partikel Library

Partikel werden in der Simulation zur Darstellung von Kontrastmittel genutzt. Es existiert eine abstrakte Basisklasse für Partikelmodelle, die das Interface für Berechnung, Erzeugung und Zeichnung der Partikel zur Verfügung stellt. Von dieser wurde in der Testumgebung ein Partikelmodell zur Simulation von Kontrastmittel abgeleitet, das die in den Kapiteln 5.5 und 6.4 vorgestellten Algorithmen implementiert.

Ähnlich den Instrumente existiert ein Partikelmanager, über den der Zugriff auf die Partikelmodelle stattfindet. Für die Simulation ist in der Regel das Bestehen eines einzigen globalen Partikelmodells ausreichend - für andere Simulation, die das Framework nutzen könnten, ist aber so der Einsatz mehrerer Partikelsimulationen (bsp. Blut, Kontrastmittel und andere Fluids) die sich in Zeichnung und Berechnung unterscheiden möglich.

3D Modell Library

Die vorgestellte Simulation unterstützt die Verwendung verschiedener geometrischer Modelle. Auch wenn Dreiecksnetze derzeit der Quasi-Standard in der Simulation dreidimensionaler Umgebung sind, so sieht das Framework für Modelle eine Ableitung von einer abstrakten Basisklasse vor. Die Basisklasse stellt ein Interface zur Manipulation (Rotation, Translation, Skalierung, Animation), und zur Abfrage ständig benötigter Operationen (Kollision, Abstandsabfrage, Zeichnung...) zur Verfügung. Abbildung 7.5 zeigt das zugehörige Klassendiagramm und die in der Testumgebung implementierten Klassen zur Verwendung von Polygonmodellen, Volumenmodellen und Zylinder-Kugel Modellen (Cathi Modelle). Geometrische Algorithmen wie generisch verwendbare Kollisionsdetektionen wurden in entsprechende Funktionen Toolkits ausgelagert und werden nur von den Modellfunktionen aufgerufen.

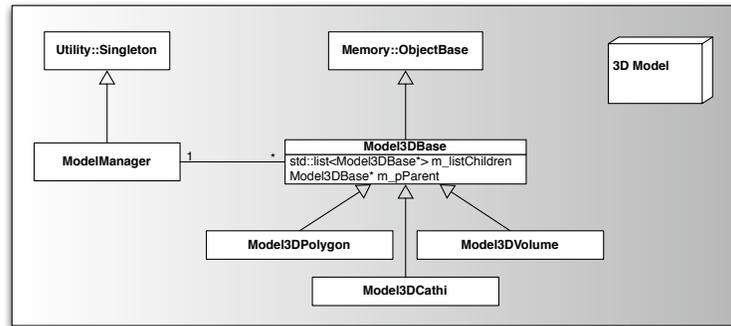


Abbildung 7.5: Klassendiagramm (vereinfacht) der 3D Modell Library

Modelle können in einer Baumstruktur verlinkt werden, um einen entsprechenden Szenengraphen abzubilden. Weiterhin wird die Baumstruktur in der Instrumentensimulation benötigt, um Suchräume (siehe Kapitel 5.1) zu beschränken.

Zugriff auf die Modelle geschieht abermals über eine Managerklasse. Neben der Verwaltung der Modelle organisiert die Managerklasse auch das schnelle Auffinden von Geometrie über die Verwendung von dynamischen Octrees (sowohl für Modelle als auch für geometrische Primitive).

Anstatt auf einzelnen Modellen zu arbeiten, können Funktionen wie Kollisionsdetektion (bsp. von Partikeln verwendet) über den Manager auch auf die globale, modellunabhängige Weltgeometrie zugreifen.

Ähnlich wie im Sofa Framework können Modelle je nach Anwendungszweck verschieden repräsentiert werden. Derzeit ist jedoch anwendungsbedingt nur eine parallele Repräsentierung als Distanzbaum und visueller Darstellung (Polygon, Volumen, Zylinder Kugel...) implementiert. Das Mapping erfolgt mittels externer Tools, da eine Onlinekonvertierung der diversen Formate derzeit noch nicht in Echtzeit erfolgen könnte.

7.2.3 Grafik

Auch die Grafik wurde in einer eigenen Bibliothek organisiert. Die Funktionsweisen ähneln den bisher beschriebenen Libraries, weshalb hier nur ein sehr knapper Überblick der vorhandenen Möglichkeiten gegeben werden soll.

Textur & Shader Library

Texturen (1D, 2D, 3D) können von Festplatte gelesen oder dynamisch erzeugt werden. Eine Managerklasse übernimmt die Verwaltung und die Kommunikation zur Grafikkarte - außerdem kümmert sich der Manager darum, dass Texturen solange wie möglich im Speicher der Grafikkarte gehalten werden und erst bei Bedarf dynamisch in den Hauptspeicher ausgelagert oder (bei extern geladenen Texturen) zum späteren Nachladen verworfen werden.

Gleiches geschieht im Shader Manager, der entsprechend Vertex oder Fragment Shader und die zugehörigen Programme verwaltet, auslagert, auf die Grafikkarte über-

trägt und bei Bedarf aktiviert.

Render Library

Diese Bibliothek ist für einen sauberen Ablauf des Renderings zuständig. Es wird sichergestellt, dass sich die Grafikkarte vor und nach jedem Frame in einem wohldefinierten Zustand befindet. Render Durchläufe werden streng nach Modellen, Partikeln, Hintergründen etc. getrennt. Eine Sortierung der Geometrien ist aufgrund des besonderen Umfelds (Simulation von Röntgenstrahlung) nicht nötig und wird daher nicht in dieser Klasse implementiert.

Es existieren Funktionen für den Einsatz globaler Shader (Verwischungseffekte, Schärfung, Rauschen) und die Aufnahme und Wiedergabe von Standbildern/Filmen sowie die folgende Zwischenspeicherung in Texturen.

7.2.4 Werkzeuge & Generische Klassen

In diesem letzten Abschnitt werden kurz die wichtigsten generischen Klassen und Werkzeuge vorgestellt, ohne auf das konkrete Design einzugehen. Bei Interesse sei hier auf die im Sourcecode vorhandene Dokumentation verwiesen.

Geometrie Library

Die Geometrie Library umfasst unter anderem Klassen für 3D (+1 homogene) Vektoren, 4×4 Matrizen, geometrische Primitive (Dreiecke, Tetraeder, Polygone, Vertices, Würfel), Octrees, Adaptive Distanzfelder und Splines und Deformatoren (FFD, Linear, Tetraederbasiert).

Alle Klassen sind teilweise assemblerbasiert (gcc intrinsics) auf Performance optimiert, falls möglich generisch als Templates implementiert und können auf 32- und 64 Bit Präzision arbeiten. Octrees können alle Strukturen verwalten, die Operationen für Überlappung mit Quadern und Bounding Boxen zur Verfügung stellen.

Mathematik Library

Implementiert wurden $n \times m$ Matrizen und Vektoren. Außerdem Algorithmen zur Invertierung von Matrizen, LU Zerlegung und Optimierung von Gleichungen mit Nelder Mead Simplex, Hillclimbing, Konjugierten Gradienten (mit Vorkonditionierung) und Quasi Newton Verfahren.

Die Vektoren und Matrizen wurden im Verlauf der Entwicklung aufgegeben um die BLAS Typen der Boost Bibliothek zu verwenden.

Utility Library

Die Utility Library beinhaltet Funktionalitäten, die zwar nicht zwingend benötigt werden, die Arbeit mit der Engine aber angenehmer gestalten können. So gibt es beispielsweise Utility Klassen für Dateioperationen und XML Parsing. Basisklassen für häufig genutzte Patterns (bsp. Singleton) werden zur Verfügung gestellt.

Außerdem gibt es Nutzklassen für plattformunabhängiges, streambasiertes, mehrkanaliges Eventlogging sowie einen Einstellungsmanager, der zur Vermeidung globaler Variablen genutzt werden kann, indem alle existierenden Variablen typunabhängig im System über eine eindeutige String ID zugreifbar gemacht werden können (siehe Beschreibung der GUI Klassen). Zu diesem Zweck wurde eine sogenannte Dator Klasse geschrieben, die Standardtypen in Strings konvertiert. Besondere Typen, die im Einstellungsmanager verwaltet werden sollen, können über Ableitungen der Dator Basisklasse implementiert werden. Der Manager kann zugleich für die Speicherung und das Einlesen von Programmparametern genutzt werden.

Weitere Klassen existieren für plattform- und compilerunabhängiges Profiling und zeitabhängige Dateninterpolation. Soll eine Variable beispielsweise über einen definierten Zeitraum zwischen zwei Werten interpolieren, so reicht eine einzelne Zeile aus, um eine Referenz der Variable in einen Interpolator (mit verschiedenen Interpolationsformeln) zu konvertieren, der sich beim Kernel als Task anmeldet und nach Ablauf der Interpolation selbstständig zerstört.

7.3 Performanceoptimierung

Da die Simulation in interaktiven Zeiten ablaufen soll (das bedeutet in der Kardiologie eine Bildrate von mindestens 12 Bildern pro Sekunde) ist eine Optimierung des Codes auf Performancesteigerung nötig. Die Optimierung selbst geschieht im Programm auf mehreren Ebenen.

Die oberste Ebene betrifft die Verbesserung der zu Grunde liegenden Algorithmen. Eine Beschreibung der entsprechenden Verfahren findet sich in den jeweiligen Kapiteln zur Simulation.

Die mittlere Optimierungsebene betrifft die Gestaltung des internen Programmablaufs. Die optimale Ausnutzung von Mehrprozessorarchitekturen sieht eine Aufteilung in mehrere Programmfäden (Threads) vor, die sich im Idealfall nicht gegenseitig behindern dürfen. Weiterhin können Threads derartig implementiert werden, dass Flaschenhälse wie die physikalische Berechnung keinen direkten Einfluss auf die Bildrate ausüben.

Die untere Optimierungsebene betrifft den konkreten Code einzelner Funktionen. Hier kann durch Ausnutzung direkter Assemblerprogrammierung, der Beachtung C++ relevanter Techniken oder SIMD Register ein weiterer Performancegewinn erzielt werden.

Im Folgenden werden die auf der mittleren und untersten Ebene verwendeten Techniken vorgestellt. Angaben auf einen Performancegewinn beziehen sich immer auf den verwendeten Testrechner (Macbook Pro mit Intel Core 2 Duo 2.33 Ghz mit 2GB 667 Mhz DDR2 Hauptspeicher und Mac OS X 10.5.1 Betriebssystem).

7.3.1 Threadoptimierung

Profilierungen des in einem einzigen Thread ablaufenden Programmcodes zeigten, dass durchschnittlich ca. 80% der Laufzeit auf die Berechnung der Instrumente entfällt. Um moderne Dual-, Quad- oder Mehrkern Architekturen auszunutzen, ist eine Ausgliederung der Berechnung in einen oder mehrere separate Threads also von großer

Bedeutung.

Jeder Instrumentenstrang wird daher von einem Thread, asynchron zum restlichen Programmablauf berechnet. Um Blockierungen durch aktive Mutexe gering zu halten, werden die zeitaufwändigen Berechnungen und Zeichenoperationen nicht in einem exklusiven Block ausgeführt. Stattdessen werden Container - sofern nicht persistent schreibend darauf zugegriffen wird - in einem kurzen exklusiven Block kopiert und dann auf diesen lokalen Kopien gearbeitet (STL garantiert keine Threadsicherheit auf Containeroperationen wie Sortierungen, einfügen oder entfernen von Elementen).

Wird mit schreibenden Operationen (deren Auftreten entsprechend minimiert wurde) gerechnet, werden kurzzeitig scope begrenzte Spin Locks eingesetzt. Eine Alternative wäre es gewesen, lockfreie Datenstrukturen [199] als Containerklassen zu verwenden, was die Anzahl der Blockierungen noch weiter minimiert hätte. Da aber auch ohne Verwendung der lockfreien Container bereits eine gute Performance erreicht werden kann, wurde in diesem Zusammenhang auf eine entsprechende Implementierung verzichtet.

Wie zu erwarten, kann durch den Einsatz mehrere Threads auf einem Prozessor mit zwei Kernen die Leistung, abhängig von der Anzahl der Instrumente und der tatsächlichen Länge um ca. 40% gesteigert werden.

Die Threads wurden unter Mac OS X mit der Systemprofilierungssoftware Shark optimiert. Abbildung 7.6 zeigt das lockfreundliche Verhalten der konkurrierenden Instrumententhreads. Deutlich zu sehen ist, wie drei Instrumenten Threads (0x43BC790, 0x58625D0 und 0x5860F20) mit dem Main Thread (0x5861E40) parallel ablaufen.

Dabei laufen die Instrumententhreads nahezu unterbrechungsfrei auf zwei Prozessorkernen, während der Main Thread nur in kurzen Intervallen den ersten Instrumententhread ablöst (lockfrei). Wartezeiten durch Locks entstehen nur sehr selten (im gezeigten Ausschnitt nur rechts als erster Event von 0x5860F20 zu erkennen) und wirken sich nicht störend auf das Scheduling des Hauptthreads aus.

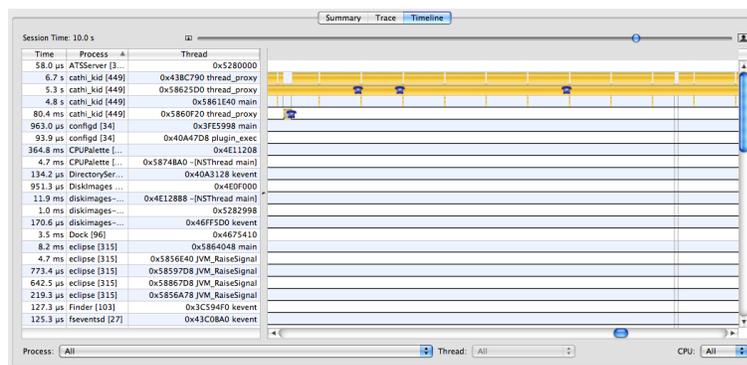


Abbildung 7.6: Systemtrace der Simulationsanwendung. Gut zu erkennen ist, wie zwei Instrumente (0x43BC790 und 0x58625D0) vollständig parallel zur restlichen Simulation (0x5861E40) berechnet werden können.

Die geringe Arbeitszeit des dritten Instrumententhreads kann dadurch erklärt werden, dass für den Testlauf beim entsprechenden Proxy keine aktiven Instrumente angemeldet waren, während die ersten beiden Proxies auf Volllast berechnet.

Ein anderes Threadmodell wird während der Berechnung der Instrumente eingesetzt. Die zeitaufwändigste Operation (nach Profilierung) ist hier die Berechnung des externen Gradienten. Abhängig von der Anzahl der Gelenke, macht die Berechnung durchschnittlich ca. 70% von der Gesamtrechenzeit aus. Daher wurde für die Berechnung ein Worker Boss Modell generiert, das abhängig von der Anzahl der Prozessorkerne die Berechnung der externen Gradienten auf mehrere Threads aufteilt.

Da der Verwaltungsaufwand trotz vorhandenem Threadpool (Threads werden nicht neu erstellt sondern nur schlafen gelegt und bei Bedarf geweckt) recht hoch ist, kann durch diese Technik auf einem Prozessor mit nur zwei Kernen keine signifikante Steigerung der Performance erzielt werden. Allerdings ist die Implementierung effizient genug, um den durch die Synchronisation der Threads entstehenden Performanceverlust auszugleichen. Es ist also zu erwarten, dass auf einer Maschine mit mehreren zur Verfügung stehenden Kernen eine Steigerung der Performance erreicht werden kann².

Abgesehen von den performancesteigernden Threads existieren nur noch zwei weitere Threads. Einer um die User Interaktion mit den Instrumenten netzwerkbasierend asynchron zu tracken und der eigentliche Hauptthread. Unnötige Verzögerungen aufgrund von Synchronisationsoverhead werden somit vermieden.

7.3.2 Funktionsoptimierung

Die Optimierung einzelner Funktionen wurde profilierungsabhängig vorgenommen. Funktionen, die in Analysen als Hot Spots auftauchen, wurden soweit als möglich unter Verwendung typischer C++ Regeln (Inlining, Late Construction, Memory Aliasing...) optimiert und mittels Shark auf ihre Cacheeffizienz untersucht. Einige Funktionen wurden nach dieser Analyse für eine effizientere Cache & Prefetch Nutzung hin optimiert.

Besondere Beachtung wurde dabei auf die Optimierung der am häufigsten genutzten Funktionen der Geometrie Bibliothek gelegt. Beinahe alle Operationen der Matrix- und Vektorklasse liegen in einer SIMD (SSE 3 unterstützenden) Version vor. Die während der Berechnung der externen Gradienten häufig verwendete Vektor Matrix Multiplikation wurde beispielsweise template spezialisierend für 32- und 64 bit Präzision in einer Batchversion entworfen, um zeitaufwändige Schreiboperationen auf SSE Vektoren einzudämmen. Die Matrix wird nur einmalig in 4 (8 bei 64 Bit) SSE Vektoren geschrieben, um anschließend mit einem Array von Vektoren multipliziert zu werden.

Der Performancegewinn bei Ausnutzung der SIMD Funktionen schwankt zwischen 10 (kleine Funktionen) und 50%. Höhere Gewinne sind in der Regel nicht zu erzielen, da die Verwendung von SSE Vektoren aufwändige Schreib- & Leseoperationen mit sich bringt, die einen hohen Overhead einführen.

Desweiteren wurden mathematische Basisfunktionen wie sin und cos optimiert. Da bei der Instrumentenberechnung keine extreme Genauigkeit (Submillimeterbereich) gefordert ist, kann auf die rechenlastige Verwendung der in der Standardlibrary implementierten trigonometrischen Funktionen verzichtet werden. Stattdessen wurde eine polynomielle Abschätzung (siehe Formel 7.1) implementiert (tabellenbasierte

²Zu beachten ist, dass bei steigender Anzahl von Prozessoren die Anzahl an Operationen pro Thread abnimmt. Ein wirklicher Performancegewinn ist also nur bei Instrumenten mit entsprechend vielen Gelenken zu erwarten.

Verfahren sind aufgrund des nicht cacheeffizienten Speicherzugriffs auf heutigen Prozessorarchitekturen nicht mehr zu empfehlen), durch die auf dem verwendeten Testrechner ein lokaler (auf die Funktion beschränkter) Performancegewinn von ca. 300% bei einem maximalen Fehler von $1.7e - 04$ erreicht werden konnte.

$$\sin \alpha \approx \frac{4}{\pi} \alpha - \frac{4}{\pi^2} \alpha^3 \quad (7.1)$$

7.4 Werkzeuge

Während der Entwicklung der Simulation wurden einige Tools zur Datenkonvertierung und Modellmanipulation geschrieben, die hier kurz vorgestellt werden sollen.

7.4.1 Cathi^{Kid} Designer

Der Cathi^{Kid} Designer ist das Werkzeug, mit dem unter anderem Datensätze generiert & manipuliert, Distanzfelder gerechnet, Octrees erzeugt und generelle Bearbeitungen durchgeführt werden können.

Angelehnt an die Benutzungsweise kommerzieller 3D Editoren können Segmente des Gefäßbaums als Objekte behandelt und einzeln oder als Gruppe transformiert werden.



Abbildung 7.7: Screenshot von Cathi^{Kid} Designer

Das Tool unterstützt diverse Dateiformate zum Im- und Export (cathi, stl, dxf, raw) und kann XML Dateien zum import im Simulator schreiben.

7.4.2 dtr2raw, stl2dtr, cathi2dtr

Kommandozeilenprogramme, die verschiedene Datenformate lesen und in adaptive Distanzfelder oder Rohdaten wandeln können.

7.4.3 Cinema 4D Skripte

Für komplexere 3D Datenbearbeitungen, wurde das kommerzielle Programm Cinema 4D R10 verwendet. Da Cinema 4D einige Operationen nicht unterstützt, wurden diese als Coffee³ oder C++ Skripte nachimplementiert (Export von Cathi^{Kid} Datensätzen, Export und Manipulation von FFD Volumina).

³Cinema 4D Skriptsprache

Kapitel 8

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurden die einzelnen Module (Kontrastmittel, Instrumentensimulation, Grafik...) zu einer voll funktionsfähigen Simulationsumgebung zusammengefügt, mit der die Behandlung eines persistierenden Ductus komplett und realistisch simuliert werden kann. Andere Defekte wie ASD sind, dank des offenen Frameworks und dem gänzlichen Verzicht auf vereinfachende, fallbezogene Annahmen, durch einen Austausch bzw. eine Veränderung des existierenden Patientendatensatzes umsetzbar.

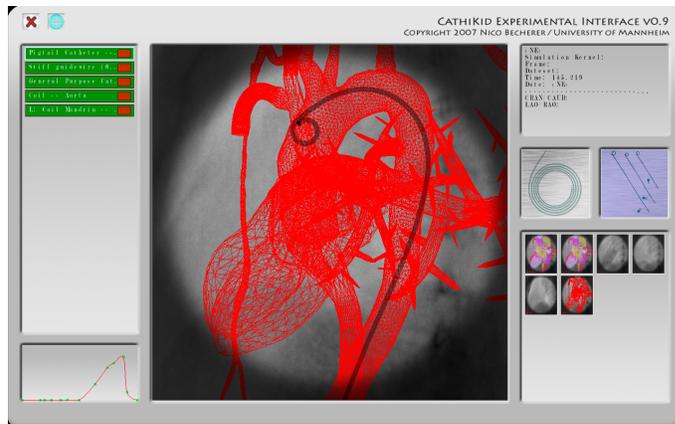


Abbildung 8.1: Die vollständige Simulationsumgebung in Aktion

Im Vergleich zu anderen wissenschaftlichen Arbeiten, ist es schwierig ein so komplexes Produkt wie eine Simulationsumgebung objektiv zu bewerten. Vieles, vor allem das Trainingseffekt und das Nutzungsempfinden ist zu stark von subjektiven Komponenten geprägt.

Daher soll in diesem Kapitel die Gesamtsimulation in ihrem einzusetzenden Rahmen als Ergebnis vorgestellt werden. Es wird ein Eingriff exemplarisch an Bildern gezeigt und dokumentiert. Anschließend findet ein Vergleich mit und eine Abgrenzung zu den existierenden Simulationen statt.

Abschließend folgt ein kurzer Abschnitt über einen Ausblick auf zukünftige Entwicklungsschritte.

8.1 Beispieleingriff

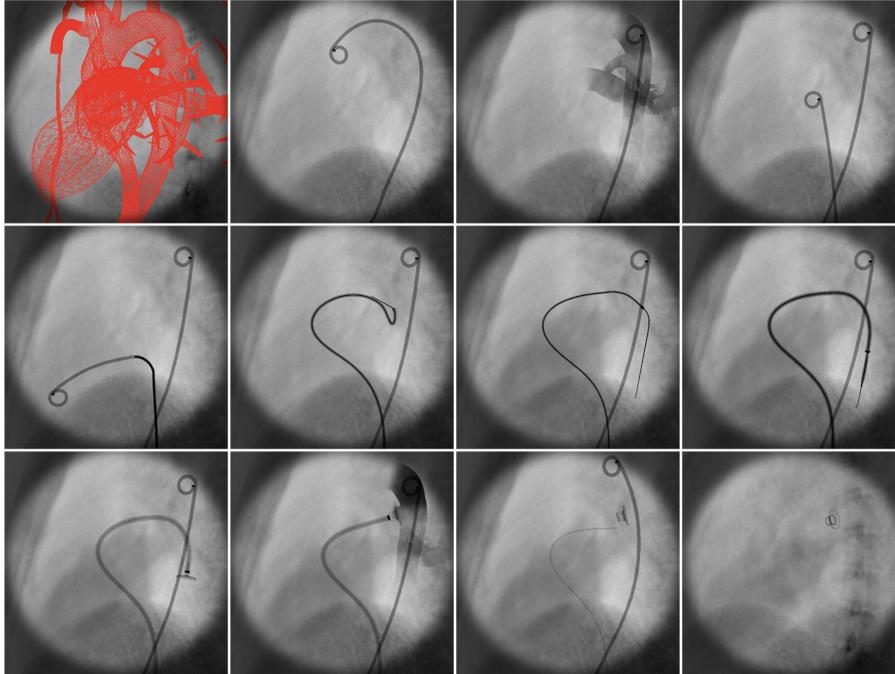


Abbildung 8.2: Exemplarisch simulierter Ductuseingriff

Abbildung 8.2 zeigt exemplarisch Bilder aus einem simulierten Ductusverschluss. Von links nach rechts und von oben nach unten stellen die Einzelbilder Folgendes dar:

1. Das als Drahtgitter eingeblendete Herzmodell. Der zu verschließende Ductus ist als gekrümmte Verbindung zwischen absteigender Aorta und Lungenarterie in der rechten oberen Ecke zu erkennen.
2. Der zur Visualisierung eingesetzte Pigtailkatheter krümmt sich im transversalen Teil der Aorta.
3. Zum Sichtbarmachen des Ductus wird der Pigtailkatheter kurz vor der Einmündung platziert und Kontrastmittel injiziert.
4. Ein zweiter Pigtailkatheter wird in die Vena Cava Inferior eingeführt und ist aktuell im rechten Atrium platziert.
5. Um den Katheter in das rechte Ventrikel vorschieben zu können, wird er mit einem steifen und stark vorgebogenem 0.35 mm Draht nach unten hin gekrümmt und durch die Herzklappe geschoben.
6. Der steife Draht wird entfernt und der Pigtailkatheter durch die Pulmonararterie in der Lungenbifurkation platziert. Ein weicher (0.14 mm) Draht wird in den Katheter geschoben und der Katheter durch einen Allzweckkatheter ausgetauscht mit dem der Ductus sondiert werden kann.

7. Nach Sondierung des Ductus wird der weiche Draht in der absteigenden Aorta platziert.
8. Der Katheter wird entfernt und durch eine große Schleuse mit Führungsmandrin ersetzt.
9. Der Führungsmandrin und der weiche Draht werden entfernt. Ein Occluder mit passender Größe wird durch die Schleuse geschoben und entfaltet sich beim Verlassen der Schleuse.
10. Abermalige Gabe von Kontrastmittel zeigt, dass der Occluder den Ductus hinreichend verschließt.
11. Der Occluder wird durch Drehung des Zubringerdrahts abgelöst und Schleuse sowie Zubringerdraht und Pigtailkatheter werden entfernt.
12. Der Occluder verbleibt als Ductusverschluss im Körper des Patienten.

Natürlich können Standbilder nur einen Teil des wirklichen Simulationserlebnisses abbilden. Die gesamte Simulation läuft durchgehend flüssig in 20-30 Bildern pro Sekunde und verhält sich zu jedem Zeitpunkt realitätsnah.

8.1.1 Evaluation

Kardiologen haben das System anhand des gezeigten Eingriffs getestet und beurteilt und es für Ausbildungszwecke unter Anleitung eines erfahrenen Kardiologen als äußerst geeignet empfunden. Das Ziel eine Simulationsumgebung für die Kinderkardiologie zu schaffen wurde also erreicht.

Im Vergleich zu anderen Simulationssystemen stellt Cathi^{Kid} das erste Simulationssystem dar, mit dem Herzkathetereingriffe an Kleinkindern trainiert werden können. Laut Beurteilung von Kardiologen und eigenen Versuchen (PCR 2007, Barcelona), ist kein anderes kommerzielles System derzeit in der Lage komplexe Instrumente in Hohlräumen mit vergleichbar hoher Stabilität und Genauigkeit in Echtzeit zu simulieren.

Das einzige System, das Eingriffe in den Hohlräumen des Herzens simuliert (Sim-Suite) basiert zu großen Teilen auf dem Abspielen im Vorfeld aufgenommener Filme und dem Ablaufen vorberechneter Wege, wogegen die in dieser Arbeit vorgestellte Simulation gezielt alle simulationsrelevanten Bedingungen in Echtzeit auswertet und berechnet und damit auch die Simulation nicht im Vorfeld bekannter Arbeitsschritte und Situationen ermöglicht.

8.1.2 Ausblick

Auch wenn die vorgestellte Simulation schon viele Eingriffe trainierbar macht, so wurden doch zeitbedingt einige Punkte in der Entwicklung nicht oder nur zu geringen Teilen berücksichtigt, was entsprechend Raum für Erweiterungen und Verbesserungen der Verfahren lässt.

Instrumente agieren korrekt mit dem umliegenden Gewebe - allerdings wird die daraus resultierende Deformation des Gewebes selbst noch nicht auf die Modelldaten

des Gewebes abgebildet. Dadurch kann es stellenweise bei sehr weichen Strukturen und sehr harten Instrumenten optisch so erscheinen, als ob das Instrument das Gewebe durchdringe obwohl tatsächlich nur eine Deformation stattfindet.

Auch können zwei verschiedene Instrumentenstränge sich derzeit nicht gegenseitig beeinflussen. Es existieren Eingriffe, in denen Instrumente in Schlingen geführt werden und gegenseitig Wechselwirkungen ausüben. Hier wären Algorithmen zur Kollisionserkennung einzuführen, die entsprechend Gegenkraft ausüben und Verbindungen zwischen den Gelenken zweier Instrumentenstränge herstellen können.

Grafisch ließe sich das Kontrastmittel noch weiter verbessern. Aktuelle Rechnergenerationen lassen eine Echtzeitberechnung von Fluids in engen, komplexen Räumen nach Navier Stokes derzeit noch nicht zu. Es ist aber davon auszugehen, dass in 1-2 Jahren eine entsprechende Berechnung (ggf. mit GPU Unterstützung) möglich sein wird.

Als Zwischenschritt könnten die derzeit verwendeten Texturcluster durch dynamische prozedural generierte Texturen (bsp. Flow Noise - siehe Kapitel 4.5.3) ersetzt werden, um eine noch bessere Visualisierung von Verwirbelungseffekten zu erzielen.

Abgesehen von diesen kleinen Details kann die vorgestellte Arbeit auch heute schon dazu beitragen, dass zukünftige Kinderkardiologen eine bessere, sicherere und kostengünstige Ausbildung erhalten können.

Der große Dank für die Betreuung und Finanzierung dieser Arbeit gilt Herrn Prof. Dr. Hesser von der Klinik für Strahlentherapie und Radioonkologie am Universitätsklinikum Mannheim, der Fakultät für Medizin Mannheim der Universität Heidelberg und dem ICM der Universität Heidelberg sowie Herrn Prof. Dr. Schranz vom Kinderherzzentrum des Justus-Liebig Klinikums Gießen und Frau Dr. Kornmesser von der Firma CATHI GmbH.

Literaturverzeichnis

- [1] <http://glew.sourceforge.net>.
- [2] <http://sourceforge.net/projects/tinyxml>.
- [3] <http://www.amplatz.com/us/simsuite/index.html>.
- [4] <http://www.boost.org>.
- [5] <http://www.chai3d.org>.
- [6] <http://www.doxygen.org>.
- [7] <http://www.h3d.org>.
- [8] <http://www.immersion.com/medical/>.
- [9] <http://www.immersion.com/medical/products/endovascular/demo.php>.
- [10] <http://www.maxon.net>.
- [11] <http://www.medicalsim.org/eve/>.
- [12] <http://www.medsimulation.com/>.
- [13] <http://www.mentice.com/>.
- [14] <http://www.physiome.org.nz/>.
- [15] <http://www.sstml.com>.
- [16] Der Herzkatheter-Selbstversuch: Dichtung und Wahrheit, 2004.
- [17] Bessere Prognose für Kinder mit Herzfehlern. *Ärzte Zeitung*, 04/05, 2005.
- [18] Kinderherzstiftung. *Jahresbericht der deutschen Gesellschaft für Herzforschung*, 1:14, 2006.
- [19] New simulation-technology training for interventionalists in south africa: Baroque medical and abbott vascular satellite institute. *Cardiovasc J S Afr*, 17(4):198–199, 2006.
- [20] <http://www.cathi.de/>, September 2007.
- [21] <http://www.mecora.de/>, 10 2007.

- [22] <http://www.sawbones.com/>, 12 2007.
- [23] Nationales Register fuer angeborene Herzfehler. <http://www.kompetenznetz-angeboreneherzfehler.de/index.php?id=47>, 2007.
- [24] 1-58450-054-9. *Game Programming Gems 2*, volume 2. Charles River Media, 2001.
- [25] H. Aanaes and J. A. Baerentzen. Pseudo-normals for signed distance computation. In *Proceedings of 2003 Conference on Vision, Modeling and Visualization*, pages 407–413, 2003.
- [26] C. Abkai. Echtzeitsimulation von intravaskulärem Ultraschall (IVUS), 2006.
- [27] C. Abkai, N. Becherer, J. Hesser, and R. Männer. Real-time simulator for intravascular ultrasound (ivus). *Medical Imaging 2007: Ultrasonic Imaging and Signal Processing*, 6513, 2007.
- [28] H. Adams. Patent foramen ovale: paradoxical embolism and paradoxical data. *Mayo. Clin. Proc.*, 79:15–20, 2004.
- [29] Tomas Akenine-Möller. Fast 3d triangle-box overlap testing.
- [30] T. Alderliesten, M. Konings, and W. Niessen. Simulation of guide wire propagation for minimally invasive vascular interventions. *Proc. MICCAI 2002*, pages 245–252, 2002.
- [31] Tanja Alderliesten, Peter A N Bosman, and Wiro J Niessen. Towards a real-time minimally-invasive vascular intervention simulation system. *IEEE Trans Med Imaging*, 26(1):128–132, 2007.
- [32] Tanja Alderliesten, Maurits K Konings, and Wiro J Niessen. Modeling friction, intrinsic curvature, and rotation of guide wires for simulation of minimally invasive vascular interventions. *IEEE Trans Biomed Eng*, 54(1):29–38, 2007.
- [33] A. J. Alexander, G. W. Bandiera, and L. Mazurik. A multiphase disaster training exercise for emergency medicine residents: Opportunity knocks. *Acad. Emerg. Med.*, 12(5):404–409, 2005.
- [34] J Allard, S Cotin, F Faure, P-J Bensoussan, F Poyer, C Duriez, H Delingette, and L Grisoni. Sofa—an open source framework for medical simulation. *Stud Health Technol Inform*, 125:13–18, 2007.
- [35] Takashi Amada, Masataka Imura, Yoshihiro Yasumuro, Yoshitsugu Manabe, and Kunihiro Chihara. Particle based fluid simulation on gpu.
- [36] A. Augustin and F. Neyret. Flow-noise en temps réel. Rapport de stage d’option, École polytechnique - Evasion, laboratoire GRAVIR, juin 2006.
- [37] B. Azarbal, J. Tobias, W. Suh, V. Chan, and R. Gaster C. Dao. Association of interatrial shunts and migraine headaches: impact of transcatheter closure. *Journal Am. Coll. Cardiology*, 45:489–492, 2005.

- [38] J. Bacon, N. Tardella, J. Pratt, and J. English. The surgical simulation and training markup language: An xml-based language for medical simulation. In *Proceedings of MMVR*, pages 37–42, 2006.
- [39] J. A. Baerenzen and H. Aanaes. Signed distance computation using the angle weighted pseudo-normal. *IEEE Trans. Vis. Comput. Graph*, 11(3):243–253, 2005.
- [40] U. Bauer, B. Peters, and A. Schmaltz. Meeting abstract gmds2004. *Nationales Register für angeborene Herzfehler*, 2004.
- [41] C. Baur, D. Guzzone, and O. Georg. A virtual reality and force feedback based endoscopic surgery simulator. In *MMVR 98*, pages 110–116, 1998.
- [42] N. Becherer. Simulation von Komplikationen bei der Durchführung von PTCA Eingriffen. Master’s thesis, University of Mannheim, 2004.
- [43] L. Benson, Y. Coe, C. Houde, D. Human, and M. Paquet. Training standards for pediatric cardiac catheterization and interventional cardiology. *J. Cardiol.*, 14:907–910, 1998.
- [44] W. F. Bond, R. L. Lammers, L. L. Spillane, R. Smith-Coggins, R. Fernandez, M. A. Reznick, J. A. Vozenilek, and J. A. Gordon. The use of simulation in emergency medicine: A research agenda. *Acad. Emerg. Med.*, 14(4):353–363, 2007.
- [45] P. A. N. Bosman and T. Alderliesten. Bringing ideas into practica: Optimization in a minimally invasive vascular intervention simulation system. Technical Report UU-CS-2004-049, Utrecht University, 2004.
- [46] B. Bürger, C. Abkai, and J. Hesser. Simulation of dynamic ultrasound based on ct models for medical education. In *MMVR 16*, page acc. for publ., 2008.
- [47] H. K. Cakmak and U. Kuehnappel. Animation and simulation techniques for vr-training systems in endoscopic surgery. In *Computer Animation and Simulation 2000*, pages 173–185, 2000.
- [48] S. Cameron. A comparison of two fast algorithms for computing the distance between convex polyhedra. In *IEEE TR and A*, 1996.
- [49] S. Cameron. Enhancing GJK: Computing minimum and penetration distances between convex polyhedra. In *Int. Conf. Robotics & Automation*, April 1997.
- [50] E. Catmull and R. Rom. A class of local interpolating splines. *Computer Aided Geometric Design*, 1987.
- [51] Rabih A Chaer, Brian G Derubertis, Stephanie C Lin, Harry L Bush, John K Karwowski, Daniel Birk, Nicholas J Morrissey, Peter L Faries, James F McKinsey, and K Craig Kent. Simulation improves resident performance in catheter-based intervention: results of a randomized, controlled study. *Ann Surg*, 244(3):343–352, 2006.

- [52] CK. Chui and WL. Nowinski. Simulation of interventional neuro-radiology procedures. In *Proceedings of International Workshop on Medical Imaging and Augmented Reality*, volume June 2001, pages 87–94. MIAR, 2001.
- [53] R. W. Clough. The finite element method in plane stress analysis. In *2nd Conference on Electronic Computation*, Pittsburgh, Pennsylvania, 1960. A.S.C.E. Structural Division.
- [54] J. D. Cohen, M. C. Lin, D. Manocha, and M. Ponamgi. I-collide: an interactive and exact collision detection system for large-scale environments. In *SI3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 189–ff., New York, NY, USA, 1995. ACM.
- [55] S. Coquillart. Extended free-form deformation: A sculpturing tool for 3d geometric modeling. In F. Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, pages 187–196, 1990.
- [56] S Cotin, S L Dawson, D Meglan, D W Shaffer, M A Ferrell, R S Bardsley, F M Morgan, T Nagano, J Nikom, P Sherman, M T Walterman, and J Wendlandt. Icts, an interventional cardiology training system. *Stud Health Technol Inform*, 70:59–65, 2000.
- [57] S. Cotin, D. W. Shaffer, D. A. Meglan, M. P. Ottensmeyer, P. S. Berry, and S. L. Dawson. Caml: A general framework for the development of medical simulations. *Proceedings of SPIE: Battlefield Biomedical Technologies II*, 4037, 2000.
- [58] S. A. Cover, N. F. Ezquerra, and J. F. O'Brien. Interactively deformable models for surgery simulation. In *CG & Appl.*, volume 13, pages 68–75, 1993.
- [59] Keenan Crane. Real-time simulation and rendering of 3d fluids. In *GPU Gems 3*, pages 633–673. Addison Wesley, 2007.
- [60] F. Dacheille and A. Kaufman. Incremental triangle voxelization. In *Proceedings on Graphics Interface*, pages 205–212, 2000.
- [61] P. K. Davis. Distributed interactive simulation in the evolution of dod warfare modeling and simulation. In *Proceedings of IEEE*, volume 83, 1995.
- [62] David L Dawson, Jennifer Meyer, Eugene S Lee, and William C Pevec. Training with simulation improves residents' endovascular procedure skills. *J Vasc Surg*, 45(1):149–154, 2007.
- [63] S L Dawson, S Cotin, D Meglan, D W Shaffer, and M A Ferrell. Designing a computer-based simulator for interventional cardiology training. *Catheter Cardiovasc Interv*, 51(4):522–527, 2000.
- [64] M. Desbrun and M. P. Cani. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation '96*, pages 61–76. Springer Verlag, 1996.

- [65] K. Dinesh and Pai. Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum*, 21(3), 2002.
- [66] James R Duncan and Craig B Glaiberman. Analysis of simulated angiographic procedures: part 1—capture and presentation of audio and video recordings. *J Vasc Interv Radiol*, 17(12):1979–1989, 2006.
- [67] James R Duncan, Benjamin Kline, and Craig B Glaiberman. Analysis of simulated angiographic procedures. part 2: extracting efficiency data from audio and video recordings. *J Vasc Interv Radiol*, 18(4):535–544, 2007.
- [68] C Duriez, S Cotin, J Lenoir, and P Neumann. New approaches to catheter navigation for interventional radiology simulation. *Comput Aided Surg*, 11(6):300–308, 2006.
- [69] J. English, C. Chang, N. Tardella, and J. Hu. A vision based surgical tool tracking approach for untethered surgery simulations and training. In *MMVR 13*. IOS Press, 2005.
- [70] Kenny Erleben and Henrik Dohlmann. Scan conversion of signed distance fields. In *Proceedings fra den 15. Danske Konference i Monstergenkendelse og Billedanalyse*, pages 81–91, 2006.
- [71] C. Fares and Y. Hamam. Collision detection for rigid bodies: A state of the art review. *Graphicon*, 2005.
- [72] C. Fares and Y. Hamam. Collision detection for virtual reality using ellipsoidal fitting. *BioMedSim*, pages 26–27, 2005.
- [73] N. Foster and D. Metaxas. Realistic animation of liquids. *Graphical Models and Image Processing*, 58(5):471–483, 1996.
- [74] N. Foster and D. Metaxas. Modeling the motion of a hot, turbulent gas. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 181–188, 1997.
- [75] R. M. Freund. Penalty and barrier methods for constrained optimization. Online, February 2004.
- [76] D. M. Gaba. The future vision of simulation in health care. *Qual. Saf. Health Care*, 13:i2–i10, 2004.
- [77] J. E. Gain and N. A. Dodgson. Preventing self-intersection under free-form deformation. *IEEE Trans. Vis. Comput. Graph*, 7(4):289–298, 2001.
- [78] Anthony G Gallagher and Christopher U Cates. Virtual reality training for the operating room and cardiac catheterisation laboratory. *Lancet*, 364(9444):1538–1540, 2004.
- [79] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design patterns: Abstraction and reuse of object-oriented design. *Lecture Notes in Computer Science*, 707:406–431, 1993.

- [80] M. Gardner. Mathematical games: The fantastic combinations of John Conway's new solitaire game 'Life'. 223(4):120–123, October 1970. The original description of Conway's game of LIFE.
- [81] Russell Gayle, Paul Segars, Ming C. Lin, and Dinesh Manocha. Path planning for deformable robots in complex environments. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.
- [82] LE. Geddes. The catheter introducers. pages 38–39, 1993.
- [83] S. Gibson, J. Samosky, and A. Mor. Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback. In *TR96-19*. Mitsubishi Electric Res. Lab, 1996.
- [84] S. F. Gibson and B. Mirtich. A survey of deformable models in computer graphics. MA TR-97-19, Mitsubishi Electric Research Laboratories, Cambridge, November 1997.
- [85] S. F. Frisken Gibson, R. N. Perry, A. P. Rockwood, and T. R. Jones. Adaptively sampled distance fields: A general representation of shape for computer graphics. *Proc. SIGGRAPH 2000*, pages 249–254, 2000.
- [86] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. In *IEEE Transactions on Robotics and Automation*, volume 4, pages 193–203, 1988.
- [87] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(375-398), 1977.
- [88] T. Gotekin, M. Cavusoglu, and F. Tendick. Gipsi: An open source software development framework for surgical simulation. In *International Symposium on Medical Simulation*, pages 240–248, 2004.
- [89] N. Govindaraju, S. Redon, M. Lin, and D. Manocha. Cullide: Interactive collision detection between complex models in large environments using graphics hardware, 2003.
- [90] A. Gueziec. Meshsweeper: Dynamic point-to-polygonal mesh distance and applications. *IEEE Trans. Vis. Comput. Graph*, 7(1):47–60, 2001.
- [91] L. J. Guibas, A. Nguyen, and L. Zhang. Zonotopes as bounding volumes. In *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 803–812, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.
- [92] V. Guilloux, P. Haignon, C. Goksu, C. Kulik, and A. Lucas. Simulation of guidewire navigation in complex vascular structures. *Proc. SPIE Medical Imaging 2006*, 2006.
- [93] V. Guilloux, P. Haignon, C. Goksu, C. Kulik, and A. Lucas. Guide wire modeling for the planning of catheterization. In *MICCAI*, 2006.

- [94] R. Gulrajani, F. Roberge, and P. Savard. Moving dipole inverse ecg and eeg solutions. *IEEE Transactions on Biomedical Engineering*, BME-31(12):903–910, 1984.
- [95] J. C. Halpin and J. L. Kardos. The halpin-tsai equations: A review. *Polym. Eng. Sci.*, 16:344–352, 1976.
- [96] R. M. Harden. Trends and the future of postgraduate medical education. *Emerg. Med. J.*, 23(10):798–802, 2006.
- [97] D. Hestenes. Primer on geometric algebra for introductory mathematics and physics. Web contents, July 2005.
- [98] G. Hirota, R. Maheshwari, and M. C. Lin. Fast volume-preserving free form deformation using multi-level optimization. In *Proceedings of the Fifth ACM Symposium on Solid Modeling and Applications*, pages 234–245. ACM Press, 1999.
- [99] W. M. Hsu, J. F. Hughes, and H. Kaufman. Direct manipulation of free-form deformations. *SIGGRAPH*, pages 177–184, 1992.
- [100] <http://xmr.ucsfmedicalcenter.org/>. Expanding the frontiers of imaging guided intervention and therapy. 2007.
- [101] S. M. Hu, H. Zhang, C. L. Tai, and J. G. Sun. Direct manipulation of ffd: Efficient and explicit solutions and decomposable multiple point constraints. *The Visual Computer*, 17(6):370–779, 2001.
- [102] R. H. Beekman III, W. E. Hellenbrand, T. R. Lloyd, J. E. Lock, C. E. Mullins, J. J. Rome, and D. F. Teitel. Training guidelines for pediatric cardiac catheterization and interventional cardiology. *JACC*, 46(7):1388–1390, 2005.
- [103] M. W. Jones. 3d distance from a point to a triangle. Technical Report CSR-5-95, Department of Computer Science, University of Wales Swansea, 1995.
- [104] M. W. Jones. The production of volume data from triangular meshes using voxelisation. *Computer Graphics Forum*, 15(5):311–318, 1996.
- [105] G. Joos. *Lehrbuch der theoretischen Physik*, volume 10. Akademische Verlagsgesellschaft Gees und Portig, 1959.
- [106] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. In *Proceedings 29th Annual Conference on Computer Graphics and Interactive Techniques*, pages 339–346, 2002.
- [107] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. Simulation of facial muscle actions based on rational free form deformations. *The Int. Journal of EuroGraphics Association Computer Graphics Forum*, 2(3):59–69, 1992.
- [108] M. Kass and G. Miller. Rapid, stable fluid dynamics for computer graphics. In *SIGGRAPH 1990 Conference Proceedings*, pages 49–57, 1990.

- [109] M. Kawasaki, M. Rissanem, N. Kume, Y. Kuroda, M. Nakao, T. Kuroda, and H. Yoshihara. Vrass (virtual reality aided simulation).
- [110] Y. L. Kim, M. C. Lin, and D. Manocha. Deep: Dual space expansion for estimating penetration depth between convex polytopes. In *IEEE Conference on Robotics and Automation*, 2002.
- [111] L W Klein. Computerized patient simulation to train the next generation of interventional cardiologists: can virtual reality take the place of real life? *Catheter Cardiovasc Interv*, 51(4):528, 2000.
- [112] K. G. Kobayashi and K. Ootsubo. t-ffd: Free-form deformation by using triangular meshes. In *SM '03: Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications*, pages 226–234. ACM Press, 2003.
- [113] L. Kobbelt, S. Campagne, J. Vorsatz, and H. P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of Siggraph 98*, pages 105–114, 1998.
- [114] R. M. Koch, M. H. Gross, and D. F. Büren. Simulating facial surgery using finite element models. In *Computer Graphics*, editor, *Proc. of Siggraph 96*, volume 30, 1996.
- [115] Sebastian König and Jürgen Hesser. 3d live-wires on mosaic volumes. *Stud Health Technol Inform*, 119:264–266, 2006.
- [116] M K Konings, E B van de Kraats, T Alderliesten, and W J Niessen. Analytical guide wire motion algorithm for simulation of endovascular interventions. *Med Biol Eng Comput*, 41(6):689–700, 2003.
- [117] M.K. Konings and E.B. van de Kraats. Discretized analytical guide wire movement algorithm. Utrecht technical report 015319, Utrecht Medical Center, 2000.
- [118] Dr. Ulrike Kornmesser. *CATHI - Catheter Instruction System. Ein virtuelles Ausbildungs- und Trainingssystem für Kardiologen*. PhD thesis, University of Mannheim, 2003.
- [119] U Kornmesser, J Hesser, W Voelker, and R Manner. [cathi-training on virtual patients for catheter interventions]. *Biomed Tech (Berl)*, 47 Suppl 1 Pt 1:121–123, 2002.
- [120] S. Krishnan, A. Pattekar, M. Lin, and D. Manocha. Spherical shells: A higher-order bounding volume for fast proximity queries, 1997.
- [121] K. Krissian, X. Wu, and V. Luboz. Smooth vasculature reconstruction with circular and elliptic cross sections. *Studies in Health Technology and Informatics*, 119:273–278, 2006.
- [122] Markus Kukuk and Bernhard Geiger. A real-time deformable model for flexible instruments inserted into tubular structures. In *MICCAI*, pages 331–338, 2002.

- [123] P. Kumar, J. Mitchell, and E. Yldrm. Approximate minimum enclosing balls in high dimensions using core sets, 2003.
- [124] Chao-Hung Lai and Jiunn-Lin Wu. Temporal texture synthesis by patch-based sampling and morphing interpolation. *Comput. Animat. Virtual Worlds*, 18(4-5):415–428, 2007.
- [125] H. J. Lamousin and W. N. W. Jr. Nurbs-based free form deformations. *IEEE Computer Graphics and Applications*, 14(6):59–65, 1994.
- [126] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *Knowledge Discovery and Data Mining*, pages 16–22, 1999.
- [127] W. Lawton, R. Raghavan, S.R. Ranjan, and R. Viswanathan. Ribbons and groups: a thin rod theory for catheters and filaments. *Phys. A: Math. Gen.*, 32:1709–1735, 1999.
- [128] W. Lawton, R. Raghavan, S.R. Ranjan, and R.R. Viswanathan. Tubes in tubes: catheter navigation in blood vessels and its applications. *International Journal of Solids and Structures*, 37(22):3031–3054, 2000.
- [129] M. LeDuc, S. Payandeh, and J. Dill. Toward modeling of a suturing task. In *Graphics Interface'03 Conference*, pages 273–279, Halifax, 2003.
- [130] Julien Lenoir, Stephane Cotin, Christian Duriez, and Paul Neumann. Physics-based models for catheter, guidewire and stent simulation. *Stud Health Technol Inform*, 119:305–310, 2006.
- [131] H. Lier, S. Schroeder, and R. Hering. Patent foramen ovale: an underrated risk for divers? *Dtsch. Med. Wochenschr.*, 129:27–30, 2004.
- [132] M. C. Lin and J. F. Canny. A fast algorithm for incremental distance calculation. In *IEEE International Conference on Robotics and Automation*, pages 1008–1014, 1991.
- [133] I. Llamas, B. Kim, J. Gargus, J. Rossignac, and C. D. Shaw. Twister: A space-warp operator for the two handed editing of 3d shapes. In *Proceedings of Siggraph 03*, pages 663–668, 2003.
- [134] G. Lloyd, J. Kendall, S. Meek, and P. Younge. High-level simulators in emergency department education: thoughts from the trainers' perspective. *Emerg. Med. J.*, 24(4):288–291, 2007.
- [135] I. P. Logan, D. P. M. Willis, and N. J. Avis. Virtual environment knee arthroscopy training system. In *Society for Computer Simulation, Simulation Series*, volume 28, pages 17–22, 1996.
- [136] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer graphics*, 21(4):163–168, July 1987.

- [137] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013–1024, 1977.
- [138] R. H. Major. The history of taking the blood pressure. *Ann. Med. History* 2, page 47.
- [139] G. Marcus, P. Berczik, A. Kugel, G. Lienhart, I. Berentzen, H. Vasquez, T. Naab, R. Männer, R. Spurzem, and A. Burkert. Accelerating smoothed particle hydrodynamics for astrophysical simulations: A comparison of fpgas and gpus. *Computing Frontiers*, Noch nicht veröffentlicht, 2008.
- [140] A. Masiero and A. Chiuso. Non linear temporal textures synthesis: A monte carlo approach. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *EC-CV (2)*, volume 3952 of *Lecture Notes in Computer Science*, pages 283–294. Springer, 2006.
- [141] Michael J Jr Mastrangelo, Gina Adrales, Rod McKinlay, Ivan George, Wayne Witzke, Margaret Plymale, Don Witzke, Mike Donnelly, Jeremy Stich, Mathew Nichols, and Adrian E Park. Inclusion of 3-d computed tomography rendering and immersive vr in a third year medical student surgery curriculum. *Stud Health Technol Inform*, 94:199–203, 2003.
- [142] S. Mauch. A fast algorithm for computing the closest point and distance transform. Technical Report caltechASCI/2000.077, Applied and Computational Math., Calif. Inst. of Technology, 2000.
- [143] S. Mauch. *Efficient Algorithms for Solving Static Hamilton Jacobi Equations*. PhD thesis, Calif. Inst. of Technology, 2003.
- [144] D. Meagher. Geometric modelin using octree encoding. *Computer Graphics and Image Processing*, 19(2):26–147, 1982.
- [145] Prof. Dr. med. Achim A. Schmaltz. Rettende Eingriffe - Interventionelle Kinderkardiologie. *Herzblatt*, 3:2–9, 2002.
- [146] D. Metaxas and D. Terzopoulos. Dynamic deformation of solid primitives with constraints. *Computer Graphics*, 26(2):309–312, 1992.
- [147] B. Mirtich. V-clip: fast and robust polyhedral collision detection. *ACM Trans. Graph.*, 17(3):177–208, 1998.
- [148] K. Montgomery. Spring: A general framework for collaborative, real-time surgical simulation. In *Proceedings of MMVR*, pages 23–26, 2002.
- [149] M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of 2003 ACM Siggraph Symposium on Computer Animation*, pages 154–159, 2003.
- [150] M. Müller, S. Schirm, and M. Teschner. Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics. *Journal of Technology and Health Care*, 12(1):25–31, 2004.

- [151] Manivannan Muniyandi, Stephane Cotin, Mandayam Srinivasan, and Steven Dawson. Real-time pc based x-ray simulation for interventional radiology training. *Stud Health Technol Inform*, 94:233–239, 2003.
- [152] A. Murta and J. Miller. Modelling and rendering liquids in motion. In *Proceedings of WSCG '99*, pages 194–201, 1999.
- [153] F. S. Nooruddin and M. Sramek. Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph*, 9(2):191–205, 2003.
- [154] W. M. O’Fallon and W. H. Weidman. Long-term follow-up of congenital aortic stenosis, pulmonary stenosis, and ventricular septal defect. report from the second joint study on the natural history of congenital heart defects. *Circulation*, 87:1–126, 1993.
- [155] C. O’Sullivan and J. Dingliana. Real-time collision detection and response using sphere-trees. *Proc. 15th Spring Conference on Computer Graphics*, pages 83–92, 1999.
- [156] I. J. Palmer and R. L. Grimsdale. Collision detection for animation using sphere-trees. *Computer Graphics Forum*, 14(2):105–116, 1995.
- [157] Marc A Passman, Paul S Fleser, Jeffery B Dattilo, Raul J Guzman, and Thomas C Naslund. Should simulator-based endovascular training be integrated into general surgery residency programs? *Am J Surg*, 194(2):212–219, 2007.
- [158] B. A. Payne and A. W. Toga. Distance field manipulation of surface models. *Computer Graphics and Applications*, 12(1), 1992.
- [159] K. Perlin. An image synthesizer. In *ACM Computer Graphics (Siggraph 85)*, volume 19, pages 287–296, 1985.
- [160] Ken Perlin and Fabrice Neyret. Flow noise. In *Siggraph Technical Sketches and Applications*, page 187, Aug 2001.
- [161] R. N. Perry and S. F. Frisken. Kiazmu: A system for sculpting digital characters. *Proc. SIGGRAPH 2001*, pages 46–56, 2001.
- [162] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, August 2007.
- [163] L. Raghupathi. Simulation of bleeding and other visual effects for virtual laparoscopic surgery. Master’s thesis, University of Texas at Arlington, 2002.
- [164] Philipp Rebholz, Ulrike Kornmesser, and Jurgen Hesser. Catheter simulation system cathi: from patient data generation to cardiological training systems. *Stud Health Technol Inform*, 98:313–315, 2004.
- [165] Stephane Redon and Ming C. Lin. A fast method for local penetration depth computation. *journal of graphics tools*, 11(2):37–50, 2006.

- [166] W. T. Reeves. Particle systems - a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, 2(2):91–108, 1983.
- [167] M.D. Reller, M. Ziegler, M.J. Rice, R. C. Solin, and R.W. McDonald. Duration of ductal shunting in healthy preterm infants: An echocardiographic color flow doppler study. *J. Pediatr.*, 112:441–446, 1988.
- [168] Torsten Rohlfing and Calvin R. Maurer. Volume-preserving nonrigid registration of mr breast images using free-form deformation with an incompressibility constraint. *IEEE Transactions on medical imaging*, pages 730–741, 2000.
- [169] A. Rosenfeld and J.L. Pfaltz. Sequential operations in digital picture processing. *J. ACM.*, 13(4):471–494, 1966.
- [170] Rost and J. Randi. *OpenGL(R) Shading Language (2nd Edition)*. Addison-Wesley Professional, January 2006.
- [171] Ruiz, Mullins, and Rocchini. Core curriculum for the training of pediatric invasive/interventional cardiologists: report of the society for cardiac angiography and interventions committee on pediatric cardiology training standards. *Cath Cardiovasc Diag*, 37:409–24, 1996.
- [172] Y. Nian Wu S. Soatto, G. Doretto. Dynamic Textures. In *International Conference on Computer Vision*, pages 439–446, July 2001.
- [173] M. Saary and G. Gray. A review of the relationship between patent foramen ovale and type ii decompression sickness. *Aviat Space Environ Med.*, 72:1113–1120, 2001.
- [174] I Sakurai. Coronary artery spasm and vascular biology. cholinergic constriction. *Acta Pathol Jpn*, 41(12):865–873, 1991.
- [175] H. Samet and R. E. Webber. Hierarchical data structures and algorithms for computer graphics. part i. *IEEE Comput. Graph. Appl.*, 8(3):48–68, 1988.
- [176] S. Schein. Hardware-based free-form deformation(hffd). In *Israel-Korea Bi-National Conference on New Technologies and Visualization Methods for Product Development on Design and Reverse Engineering*, 2005.
- [177] A. Schmaltz. Gedanken eines Kinderkardiologen. *Herzblatt*, 1:8–10, 2000.
- [178] Dave Schreiner, Dave (Ed.) Schreiner, and Dave Shreiner. *OpenGL Reference Manual: The Official Reference Document to OpenGL, Version 1.2*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [179] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *Computer Graphics*, 20:151–160, 1986.
- [180] Guy Sela, Sagi Schein, and Gershon Elber. Real-time incision simulation using discontinuous free form deformation. In *ISMS*, pages 114–123, 2004.

- [181] M. J. Shapiro, J. C. Morey, S. D. Small, V. Langford, C. J. Kaylor, L. Jagminas, S. Suner, M. L. Salisbury, R. Simon, and G. D. Jay. Simulation based teamwork training for emergency department staff: does it improve clinical team performance when added to an existing didactic teamwork curriculum? *Qual. Saf. Health Care*, 13(6):417–421, 2004.
- [182] J. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. *Painless CG*, August 1994.
- [183] C. Sigg, R. Peikert, and M. Gross. Signed distance transform using graphics hardware. In *Proceedings IEEE Conf. Visualization '03*, pages 89–90, 2003.
- [184] Cláudio T. Silva and Joseph S. B. Mitchell. Greedy cuts: an advancing front terrain triangulation algorithm. In *GIS '98: Proceedings of the 6th ACM international symposium on Advances in geographic information systems*, pages 137–144, New York, NY, USA, 1998. ACM Press.
- [185] J. Stam. Interacting with smoke and fire in real time. *Communications of the ACM*, 43(7):76–83, 2000.
- [186] J. Stam. A simple fluid solver based on the fft. *Graphics Tools*, 6(2):43–52, 2001.
- [187] J. Stam. Real-time fluid dynamics for games. In *Proceedings of the Game Developer Conference*, 2003.
- [188] Jos Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128. ACM Press, 1999.
- [189] Jos Stam and Eugene Fiume. Depicting fire and other gaseous phenomena using diffusion processes. *Computer Graphics*, 29:129–136, 1995.
- [190] J. Strain. Fast tree-based redistancing for level set computations. *J. Computational Physics*, 152(2):664–686, 1999.
- [191] Avneesh Sud, Naga Govindaraju, Russell Gayle, Ilknur Kabul, and Dinesh Manocha. Fast proximity computation among deformable models using discrete voronoi diagrams. *ACM Trans. Graph.*, 25(3):1144–1153, 2006.
- [192] G. Szekely, C. Brechbühler, and R. Hutter. Modelling soft tissue deformation for laparoscopic surgery simulation. In *MICCAI 98*, pages 550–561, 1998.
- [193] Y. M. Tang, A. F. Zhou, and K. C. Hui. Comparison of fem and bem for interactive object simulation. *Computer-Aided Design*, 38(8):874–886, 2006.
- [194] Taskran, Hasan Dogu, and Ugur Gudukbay. Physically-based simulation of hair strips in real-time. *WSCG (Short Papers)*, pages 153–156, 2005.
- [195] G. Thuermer and C. A. Wuethrich. Computing vertex normals from polygonal facets. *J. Graphics Tools*, 3(1):43–46, 1998.
- [196] D. Tonnesen. *Dynamically Coupled Particle Systems for Geometric Modeling, Reconstruction and Animation*. PhD thesis, University of Toronto, 1998.

- [197] Henrik Turbell. *Cone-Beam Reconstruction Using Filtered Backprojection*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, February 2001. Dissertation No. 672, ISBN 91-7219-919-9.
- [198] R. MacCracken und K. I. Joy. Free form deformations with lattices of arbitrary topology. In *SIGGRAPH*, pages 181–188, 1996.
- [199] John D. Valois. Lock-free linked lists using compare-and-swap. In *Symposium on Principles of Distributed Computing*, pages 214–222, 1995.
- [200] Johannes H van Oostrom, S Kentgens, J E W Beneken, and J S Gravenstein. An integrated coronary circulation teaching model. *J Clin Monit Comput*, 20(4):235–242, 2006.
- [201] J. von Neumann and A. W. Burks. Theory of self-reproducing automata. *University of Illinois*, 1966.
- [202] J. Vozenilek, J. S. Huff, M. Reznik, and J. A. Gordon. See one, do one, teach one: Advanced technology in medical education. *Acad. Emerg. Med.*, 11(11):1149–1154, 2004.
- [203] Gabriele Wagner. CT mit Doppel-Röhre: schneller, schärfer - und die Strahlendosis sinkt, 2006.
- [204] Fei Wang, Lindo Duratti, Evren Samur, Ulrich Spaelter, and Hannes Bleuler. A Computer-Based Real-Time Simulation of Interventional Radiology. In *27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE-EMBS)*, 2007.
- [205] George Webb, Alex Norcliffe, Peter Cannings, Paul Sharkey, and Dave Roberts. Virtual reality and interactive 3d as effective tools for medical training. *Stud Health Technol Inform*, 94:392–394, 2003.
- [206] X. Wei, W. Li, K. Mueller, and A. Kaufmann. Simulating fire with texture splats. In *Proceedings of the conference on Visualization 02*, pages 227–235, 2002.
- [207] X. Wei, W. Li, K. Mueller, and A. Kaufmann. The lattice-boltzmann method for simulating gaseous phenomena. In *IEEE Transactions on Visualization and Computer Graphics*, volume 10, pages 164–176, 2004.
- [208] Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [209] J. Youn and K. Wohn. Realtime collision detection for virtual reality applications. In *VR*, pages 415–421, 1993.
- [210] J. Zátönyi, R. Paget, G. Székely, and M. Bajka. Real-time synthesis of bleeding for virtual hysteroscopy. In Randy E. Ellis and Terry M. Peters, editors, *MICCAI (1)*, volume 2878 of *Lecture Notes in Computer Science*, pages 67–74. Springer, 2003.

- [211] Yanci Zhang, Barbara Solenthaler, and Renato Pajarola. Gpu accelerated sph particle simulation and rendering. In *International Conference on Computer Graphics and Rendering*. ACM Press, 2007.