

Otto-von-Guericke-Universität Magdeburg



Fakultät für Informatik  
Institut für Simulation und Graphik

# Interaktive und dynamische Visualisierung für die chirurgische Ausbildung und Interventionsplanung

Dissertation  
zur Erlangung des akademischen Grades  
Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik  
der Otto-von-Guericke-Universität Magdeburg

von: Dipl.-Ing. Ragnar Bade  
geb. am 06.05.1978 in Jena

Gutachter: Prof. Dr. Bernhard Preim  
Prof. Dr. Helwig Hauser  
Prof. Dr. Heinz Handels

Magdeburg, den 11. Juni 2008

**Bade, Ragnar (Dipl.-Ing.):**

*Interaktive und dynamische Visualisierung  
für die chirurgische Ausbildung und Inter-  
ventionsplanung*

Dissertation, Otto-von-Guericke-Universität  
Magdeburg, 11. Juni 2008.

## **Zusammenfassung**

Computerunterstützte Interventionsplanung für chirurgische Eingriffe kann die Qualität, Quantität und auch Machbarkeit komplizierter Eingriffe verbessern. Computerunterstützung erfolgt dabei bei der Bildaufnahme mit modernen CT- und MRT-Geräten, der Analyse und Extraktion von Informationen aus den aufgenommenen Bilddaten, der angemessenen Visualisierung und Präsentation dieser Daten sowie bei der Therapieentscheidungsfindung, zum Beispiel durch Abschätzung und Simulation von zu erwartenden Therapieergebnissen. Herausforderungen für die Computerunterstützung stellen unter anderem die Reduktion von Artefakten aus Bildaufnahme und Bildanalyse sowie die standardisierte dynamische Visualisierung patientenspezifischer Bilddaten dar.

Im Rahmen dieser Arbeit werden Techniken zur Extraktion von qualitativ hochwertigen Oberflächenmodellen für die Visualisierung der durch Bildanalyse gewonnenen dreidimensionalen Patientenanatomie untersucht sowie Techniken zur standardisierten dynamischen Präsentation und effizienten interaktiven Exploration der dreidimensionalen Daten analysiert.

Darauf aufbauend werden zwei effiziente Verfahren für die aufgabenangepasste Extraktion von Oberflächenmodellen aus Bildanalyseergebnissen entwickelt. Weiterhin werden Interaktionstechniken zur effizienten Exploration der dreidimensionalen Patientenanatomie evaluiert und es wird ein skriptbasiertes Verfahren zur Unterstützung der interaktiven Exploration und zur Erzeugung standardisierter dynamischer Visualisierungen vorgestellt.

Abschließend werden alle erarbeiteten Methoden, Techniken und Erkenntnisse in der Konzeption und Entwicklung eines Lernsystems für die chirurgische Interventionsplanung in der Leberchirurgie zusammengeführt.



# Danksagung

Die vorliegende Arbeit entstand in direkter und indirekter Zusammenarbeit mit einer Vielzahl an Kollegen, Studenten und Kooperationspartnern. Für deren intensive und freundschaftliche Zusammenarbeit möchte ich mich herzlich bedanken. Mein besonderer Dank gilt Prof. Bernhard Preim, der mir immer durch seine wissenschaftliche Unterstützung und sein aktives Vorbild ein inspirierender Doktorvater war.

Bedanken möchte ich mich bei allen Kollegen, mit denen ich inhaltlich zusammenarbeiten konnte. Mein Dank geht vor allem an Arno Krüger, Steffen Oeltze, Konrad Mühler, Jeanette Cordes, Alexandra Baer, Christian Tietjen und Jana Dornheim. Arno Krüger und Steffen Oeltze bereicherten die Arbeit durch zahlreiche inspirierende Ideen und Diskussionen. Unter anderem haben sie sich als Alpha-Tester bei den Evaluationsstudien engagiert und trugen damit entscheidend zu deren erfolgreichem Gelingen bei. Konrad Mühler trug bereits mit seiner Diplomarbeit die Arbeit an adaptiven skriptbasierten Animationen praktisch voran. Konrad Mühler zusammen mit Jeanette Cordes haben meine Ideen zum LiverSurgeryTrainer aufgegriffen und tragen mit ihrer Kompetenz in der Softwareentwicklung und Lernpsychologie zu dessen Weiterentwicklung bei. Mit Alexandra Baer und Christian Tietjen verbindet mich das Interesse an illustrativer Visualisierung, welches in den Visualisierungen des LiverSurgeryTrainer mehrfach zum Ausdruck kommt.

Inhaltlich und praktisch wurde die Arbeit bei der Konzeption und Entwicklung des LIVER SURGERY TRAINER in großen Teilen durch die Diplomarbeiten von Sebastian Mirschel, Ivonne Riedel und Lars Schmitt unterstützt. Zur Artefaktreduktion in medizinischen Oberflächenmodellen trugen die Diplomarbeiten von Jens Haase und Christian Schumann einen großen Anteil bei. Weiterführende Arbeiten bei der Netzgenerierung und Flusssimulation wurden durch Dr. Stefan Zachow am ZIB unterstützt.

Großer Dank geht an Prof. Karl J. Oldhafer. Ich bin ihm und seinem Team am AKH Celle nicht nur zu Dank verpflichtet, weil er uns freundlich in seinen OP aufnahm, uns Filmaufnahmen von Operationen ermöglichte und aufbereitete Fälle für den LiverSurgeryTrainer bereitstellte, sondern auch dafür, dass er mit großer Hingabe die Entwicklung auf medizinischer Seite stark unterstützte. Dies trug erheblich zur Entwicklung eines Systems bei, welches auch von Ärzten ernst genommen werden kann.

Ich möchte mich außerdem bei MeVis Research für die Aufarbeitung und Bereitstellung von Fällen für das Lernsystem sowie für zur Verfügung gestellte erweiterte Komponenten der MeVisLab-Bibliothek (insbesondere der WEM-Bibliothek) bedanken. Mein Dank geht besonders an Dr. Felix Ritter, Olaf Konrad, Florian Link und Milo Hindennach, welche mich als Koautoren unterstützten und/oder mir bei der Umsetzung in MeVisLab beistanden.

Besonderer Dank geht auch an meine Frau Andrea und meine Kinder Jeldrik und Jenna, die auch außerhalb der „üblichen“ Arbeitszeiten Verständnis für meine Arbeit aufbrachten und mich unterstützten. Lieben Dank an euch.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Visualisierung . . . . .	2
1.2	Interaktion . . . . .	3
1.3	Ausbildung und Training . . . . .	4
1.4	Fragestellung und Aufgaben . . . . .	4
1.5	Gliederung . . . . .	5
<b>2</b>	<b>Computerunterstützte Interventionsplanung und chirurgische Ausbildung</b>	<b>7</b>
2.1	Vorbetrachtungen . . . . .	7
2.2	Leberoperationsplanung . . . . .	8
2.2.1	Medizinischer Hintergrund . . . . .	8
2.2.2	Computerunterstützte Planung . . . . .	10
2.3	Planung von Neck Dissections . . . . .	11
2.4	3D-Osteotomieplanung . . . . .	14
2.5	Computerunterstützte Anatomieausbildung . . . . .	16
2.5.1	VoxelMan . . . . .	16
2.5.2	Digital Anatomist . . . . .	16
2.5.3	AnatomyBrowser . . . . .	17
2.5.4	ZoomIllustrator und 3D Puzzle . . . . .	18
2.6	Allgemeine CBTs in der Medizin . . . . .	19
2.6.1	CASUS . . . . .	19
2.6.2	Dogs 'n Drugs . . . . .	20
2.6.3	CAMPUS . . . . .	20
2.7	Chirurgiesimulatoren . . . . .	21
2.7.1	Karlsruhe Endoscopic Surgery Trainer . . . . .	22
2.7.2	Leberchirurgiesimulation . . . . .	22
2.7.3	Hysteroscopy Simulator . . . . .	22
2.8	Zusammenfassung . . . . .	24
2.8.1	Herausforderungen der computerunterstützten Interventionsplanung . . . . .	24
2.8.2	Herausforderungen der computerunterstützten Ausbildung . . . . .	25
<b>3</b>	<b>Artefaktreduktion in medizinischen Oberflächenmodellen:</b>	
	<i>Grundlagen</i>	<b>27</b>
3.1	Warum binäre Segmentierung? . . . . .	28
3.2	Probleme bei der Oberflächenextraktion . . . . .	29
3.2.1	Konturkorrespondenzproblem . . . . .	29
3.2.2	Oberflächenextraktion aus binären Bilddaten . . . . .	30

3.3	Zielsetzung und Herangehensweise . . . . .	31
3.4	Verwendete Notation . . . . .	32
3.4.1	Nachbarschaften . . . . .	32
3.4.2	Explizite vs. Implizite Integration . . . . .	32
3.5	Bestimmung von Krümmungsmaßen . . . . .	33
3.6	Datenstrukturen für polygonale Oberflächennetze . . . . .	37
3.6.1	Explizite Repräsentation . . . . .	37
3.6.2	Indizierte Eckpunkt- und Kantenlisten . . . . .	37
3.6.3	<i>Winged-Edge</i> Repräsentation . . . . .	38
3.6.4	<i>Halfedge</i> Repräsentation . . . . .	39
3.6.5	Effiziente Repräsentation für die diskrete Krümmungsberechnung . . . . .	39
<b>4</b>	<b>Strategien zur Artefaktreduktion</b>	<b>41</b>
4.1	Glättung von binären Segmentierungsmasken . . . . .	41
4.1.1	Tiefpass-Filterung . . . . .	41
4.1.2	Fuzzy Boundaries . . . . .	42
4.1.3	Sub-Voxel Interpolation . . . . .	43
4.1.4	Level-Sets . . . . .	44
4.2	Artefaktreduktion bei der Oberflächenextraktion . . . . .	44
4.2.1	Trilineare Interpolation . . . . .	44
4.2.2	Sub-Voxel Interpolation . . . . .	45
4.2.3	Approximation mit impliziten Oberflächen . . . . .	45
4.2.4	Kombinierte Oberflächenextraktion und Glättung . . . . .	45
4.3	Glättung von extrahierten Oberflächenmodellen . . . . .	46
4.3.1	Laplace-Glättung . . . . .	47
4.3.2	Laplace+ $\mathcal{HC}$ -Glättung . . . . .	48
4.3.3	Taubin's $\lambda$ - $\mu$ -Glättung . . . . .	49
4.3.4	Mean Curvature Flow - Glättung . . . . .	50
4.3.5	Mean- und Median-Glättung der Normalen . . . . .	51
4.3.6	Merkmalerhaltende Glättung . . . . .	53
4.3.7	Subdivision Surfaces . . . . .	55
4.3.8	Vergleich von Glättungsverfahren . . . . .	56
4.4	Zusammenfassung und Diskussion . . . . .	57
<b>5</b>	<b>Evaluation von Glättungsverfahren</b>	<b>59</b>
5.1	Objektanalyse und -klassifizierung . . . . .	59
5.1.1	Kompakte Objekte . . . . .	60
5.1.2	Flache Objekte . . . . .	61
5.1.3	Elongierte Objekte . . . . .	61
5.2	Anforderungen an die Glättung . . . . .	62
5.3	Kriterien für die Bewertung der Glättungsergebnisse . . . . .	63
5.3.1	Bewertung der Glattheit . . . . .	63
5.3.2	Bewertung der Genauigkeit . . . . .	64
5.3.3	Bewertung der praktischen Anwendbarkeit . . . . .	64
5.4	Evaluation: Konzept und Durchführung . . . . .	65
5.4.1	Glättungsverfahren und Parametrisierung . . . . .	65
5.4.2	Auswahl der Oberflächenmodelle . . . . .	68
5.4.3	Ausgewählte Bewertungskriterien . . . . .	69



5.5	Umsetzung und Implementierung . . . . .	69
5.5.1	Implementierungskonzept . . . . .	70
5.5.2	Modularisierung der Glättungsverfahren . . . . .	70
5.5.3	Durchführung . . . . .	73
5.6	Ergebnisse . . . . .	74
5.6.1	Glättungsergebnisse kompakter Objekte . . . . .	74
5.6.2	Glättungsergebnisse flacher Objekte . . . . .	76
5.6.3	Glättungsergebnisse elongierter Objekte . . . . .	79
5.7	Glättungsempfehlungen . . . . .	81
5.8	Diskussion und Fazit . . . . .	82
<b>6</b>	<b>Voxelbegrenzte Oberflächennetzglättung</b>	<b>85</b>
6.1	Vorbetrachtungen . . . . .	85
6.2	Begrenzung der Glättungsbewegung . . . . .	86
6.2.1	Cell-Size Constraint . . . . .	86
6.2.2	Cell-Center Constraint . . . . .	86
6.2.3	Cell-Edge Constraint . . . . .	87
6.2.4	Diamond Constraint . . . . .	87
6.3	Umsetzung und Implementierung . . . . .	88
6.3.1	Constraints Submodul . . . . .	88
6.3.2	Umsetzung des Diamond Constraint . . . . .	89
6.4	Ergebnisse . . . . .	90
6.5	Zusammenfassung und Diskussion . . . . .	92
<b>7</b>	<b>Approximation von Oberflächenmodellen segmentierter Gefäßstrukturen</b>	<b>95</b>
7.1	Vorbetrachtungen . . . . .	95
7.2	Verwandte Arbeiten . . . . .	96
7.2.1	Modellfreie Extraktion von Oberflächenmodellen . . . . .	96
7.2.2	Modellbasierte Erzeugung von Oberflächenmodellen . . . . .	96
7.2.3	Fazit . . . . .	97
7.3	Approximation von Gefäßstrukturen mit MPUIs . . . . .	97
7.3.1	MPUIs . . . . .	97
7.3.2	Punktwolkenextraktion . . . . .	98
7.3.3	Automatische Parameterwahl . . . . .	101
7.4	Umsetzung . . . . .	102
7.5	Ergebnisse . . . . .	102
7.6	Zusammenfassung und Diskussion . . . . .	104
<b>8</b>	<b>Effiziente 3D-Navigation</b>	<b>107</b>
8.1	Problem- und Anforderungsanalyse . . . . .	107
8.1.1	Problemanalyse . . . . .	107
8.1.2	Anforderungsanalyse . . . . .	110
8.2	3D-Rotationstechniken – Stand der Technik . . . . .	111
8.2.1	Chen's Virtual Trackball . . . . .	111
8.2.2	Bell's Virtual Trackball . . . . .	112
8.2.3	Shoemake's Virtual Trackball . . . . .	113
8.2.4	Two-Axis Valuator Trackball . . . . .	114

8.2.5	Two-Axis Valuator with Fixed Up-vector . . . . .	115
8.2.6	Zusammenfassung . . . . .	115
8.3	Evaluation von Rotationstechniken . . . . .	116
8.3.1	Vorarbeiten und Stand der Forschung . . . . .	116
8.3.2	Evaluationskonzept . . . . .	117
8.3.3	Statistische Analyse . . . . .	120
8.3.4	Ergebnisse und Diskussion . . . . .	120
8.4	Kritische Anmerkungen und Ausblick . . . . .	123
<b>9</b>	<b>Dynamische Visualisierung und Interaktion</b>	<b>125</b>
9.1	Aufgaben und Möglichkeiten dynamischer Visualisierungen . . . . .	126
9.2	Dynamische Visualisierung patientenspezifischer Daten . . . . .	127
9.3	Verwandte Arbeiten . . . . .	129
9.3.1	Dynamische <i>Story Telling Informed</i> Visualisierungen . . . . .	130
9.3.2	Ansätze adaptiver dynamischer Visualisierungen . . . . .	132
9.3.3	Dynamische Visualisierungen in Anwendungen . . . . .	133
9.4	Skriptkonzept für adaptive dynamische Visualisierungen . . . . .	133
9.4.1	Hierarchisches Konzept . . . . .	133
9.4.2	Befehlssyntax und Aufbau der Skriptsprache . . . . .	134
9.4.3	Ersetzungsregeln . . . . .	138
9.4.4	Funktionalität des Skriptinterpreters . . . . .	141
9.5	Anwendungsszenarien des Skriptkonzeptes . . . . .	141
9.5.1	Generierung standardisierter Ansichten und Animationen . . . . .	143
9.5.2	Dynamische Visualisierung im Interaktionskontext . . . . .	145
9.6	Zusammenfassung und kritische Anmerkungen . . . . .	146
<b>10</b>	<b>Computerunterstütztes Training der Leberoperationsplanung</b>	<b>149</b>
10.1	Motivation . . . . .	149
10.1.1	Trainings- und Ausbildungsbedarf bei der Interventionsplanung . . .	150
10.1.2	Herausforderungen der Interventionsplanung in der Leberchirurgie . .	150
10.2	Anforderungen und Ziele . . . . .	151
10.2.1	Ziele . . . . .	151
10.2.2	Zielgruppenspezifische Anforderungen . . . . .	151
10.2.3	Anforderungen aus dem medizinischen Lerninhalt . . . . .	152
10.2.4	Lerntheoretische Anforderungen . . . . .	153
10.3	Konzept des LIVER SURGERY TRAINER . . . . .	153
10.3.1	Falldatenbank und Datenbasis . . . . .	154
10.3.2	Fallbearbeitung . . . . .	154
10.3.3	Interaktive Patientenakte . . . . .	155
10.3.4	Hilfekonzept . . . . .	156
10.3.5	Implementierungsaspekte . . . . .	158
10.4	Implementierungsvarianten . . . . .	159
10.5	Interaktions- und Visualisierungsaspekte . . . . .	161
10.5.1	Fallauswahl . . . . .	161
10.5.2	3D-Visualisierung patientenspezifischer Daten . . . . .	166
10.5.3	Interaktion in 3D-Szenen . . . . .	168
10.6	Zusammenfassung . . . . .	168

<b>11 Fazit und Ausblick</b>	<b>171</b>
11.1 Beitrag zur Reduktion von Artefakten in medizinischen Oberflächenmodellen	171
11.2 Beitrag zur effizienten 3D-Navigation	172
11.3 Beitrag zur interaktiven dynamischen Visualisierung	173
11.4 Beitrag zur computerunterstützten chirurgischen Ausbildung	174
11.5 Ausblick	174
11.5.1 Extraktion von qualitativ hochwertigen Oberflächenmodellen	175
11.5.2 Interaktion und Navigation in 3D-Visualisierungen	175
11.5.3 Dynamische Visualisierung patientenspezifischer Daten	175
11.5.4 LIVER SURGERY TRAINER	175
<b>Literaturverzeichnis</b>	<b>177</b>
<b>A Ergebnisdiagramme der Evaluation der Glättungsverfahren</b>	<b>197</b>
A.1 Oberflächenmodell der Leber	198
A.1.1 Symmetrische Hausdorff-Distanz	198
A.1.2 Volumenerhalt	199
A.1.3 Rechenzeit	200
A.2 Oberflächenmodell des Lymphknotens	201
A.2.1 Symmetrische Hausdorff-Distanz	201
A.2.2 Volumenerhalt	202
A.2.3 Rechenzeit	203
A.3 Oberflächenmodell des Beckens	204
A.3.1 Symmetrische Hausdorff-Distanz	204
A.3.2 Volumenerhalt	205
A.3.3 Rechenzeit	206
A.4 Oberflächenmodell des Halsmuskels	207
A.4.1 Symmetrische Hausdorff-Distanz	207
A.4.2 Volumenerhalt	208
A.4.3 Rechenzeit	209
A.5 Oberflächenmodell des Gefäßbaumes	210
A.5.1 Symmetrische Hausdorff-Distanz	210
A.5.2 Volumenerhalt	211
A.5.3 Rechenzeit	212
A.6 Oberflächenmodell der A. carotis	213
A.6.1 Symmetrische Hausdorff-Distanz	213
A.6.2 Volumenerhalt	214
A.6.3 Rechenzeit	215



# Einleitung

---

Fortschritte in der medizinischen Bildgebung konnten in den letzten Jahrzehnten die medizinische Diagnostik grundlegend revolutionieren. Das Vorliegen der Bilddaten in digitaler Form ermöglicht dabei eine computerunterstützte Bildanalyse und Diagnostik. So werden mit geeigneter Software z.B. kleine Lungenrundherde bei der Lungenkrebsdiagnostik automatisch detektiert [Marten et al. 2005] oder die Diagnostik und Klassifikation von Brusttumoren unterstützt [MeVis BreastCare 2007].

Ein ähnlich revolutionärer Fortschritt konnte bei den immer differenzierter zur Verfügung stehenden Therapiemöglichkeiten erreicht werden. Das Gebiet der onkologischen Leberchirurgie stellt hierbei ein besonders überzeugendes Beispiel dar, bei dem Möglichkeiten und Grenzen des Behandelbaren unter anderem durch Chemotherapie, Radiofrequenzablation, Kryotherapie und die mögliche Kombination dieser und anderer Behandlungsmethoden enorm ausgeweitet werden konnten.

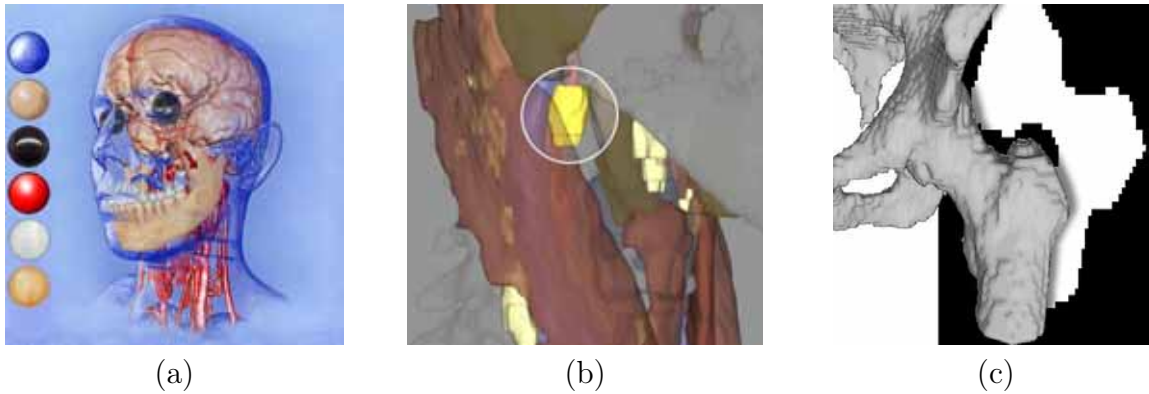
Eine entsprechende Vielzahl an Therapie- und Therapiekombinationsmöglichkeiten erfordert komplizierte, sehr differenzierte und patientenindividuelle Therapieentscheidungen und eine entsprechend aufwändige Therapie- bzw. Interventionsplanung. Um die komplizierten Therapieentscheidungsprozesse zu unterstützen und eine patientenindividuelle Entscheidungsfindung und Interventionsplanung zu ermöglichen, gewinnen computerunterstützte Entscheidungs- und Planungsprozesse zunehmend an Bedeutung.

Ankerpunkte für die Computerunterstützung bei der Interventionsplanung und Entscheidungsfindung sind hierbei 1. die Segmentierung und Quantifizierung von bildinhärenten Informationen [Glüer et al. 2006] (z.B. Bestimmung von Volumina und Abständen anatomischer und pathologischer Strukturen im 3D-Bildraum), 2. die Berechnung bzw. Simulation von zu erwartenden Interventionsergebnissen (z.B. Weichgewebsdeformation nach Knochenumstellung [Zachow 2005]), 3. die Visualisierung von Bildanalyseergebnissen, Patientenanatomie, quantifizierten Informationen, Simulationsergebnissen und möglichen Interventionsvarianten sowie 4. Interaktionsmöglichkeiten zur Exploration dieser Daten und zur Definition von Interventionsvarianten, um die Entscheidungsfindung zu unterstützen. Darüber hinaus ist auch 5. die Computerunterstützung bei der Ausbildung und dem Training von entsprechend computerunterstützten Therapieentscheidungs- und Interventionsplanungsprozessen nötig, um diese erlernbar und damit anwendbar zu machen.

Spezielle Anforderungen auf der Ebene der Visualisierung und Mensch-Computer-Interaktion werden dabei gestellt an:

- Techniken zur direkten oder indirekten Visualisierung der komplexen dreidimensionalen Patientenanatomie
- Techniken zur Erzeugung von patientenindividuellen anatomischen Oberflächenmodellen aus segmentierten Volumendaten
- Techniken zur standardisierten Präsentation und dynamischen Visualisierung (Standardisierte Animationen)
- Techniken zur effizienten Exploration und Navigation im 3D-Raum (Rotation)

Der Entwicklung und Erprobung derartiger Techniken ist diese Arbeit gewidmet.



**Abbildung 1.1:** Visualisierung medizinischer Daten: (a) Illustrative direkte Volumendarstellung eines Schädels mittels *Style Transfer Functions* [Bruckner und Gröller 2007]; (b) Indirekte Volumenvisualisierung für die Interventionsplanung bei Halslymphknotenausräumungen [Krüger et al. 2005a]; (c) Oberflächenmodell mit Aliasing-Artefakten und zugehörige binäre Segmentierungsmaske.

## 1.1 Visualisierung

Einen Vorteil der computerunterstützten Interventionsplanung stellen die Möglichkeiten zur dreidimensionalen Visualisierung der schichtbasierten medizinischen Bilddaten dar. Hierbei finden Techniken mit einer Vielzahl an Visualisierungsvarianten und -parametern zur direkten als auch indirekten Volumenvisualisierung Anwendung. Bei der Entwicklung neuer direkter Volumenvisualisierungstechniken sind vor allem die Arbeitsgruppe um Prof. Eduard Gröller (siehe z.B. Abb. 1.1a, [Viola et al. 2006; Bruckner und Gröller 2007]) sowie die Arbeitsgruppe um Peter Hastreiter und das VRVis (siehe z.B. [Engel et al. 2006]) richtungsweisend. Das Gebiet der indirekten Volumenvisualisierung für die Interventionsplanung ist ein Fokus der Arbeitsgruppe des Autors (siehe z.B. Abb. 1.1b, [Krüger et al. 2005b; Baer et al. 2007]).

### Reduktion von Artefakten

Die Erzeugung qualitativ hochwertiger Visualisierungen aus binär segmentierten Volumendaten (siehe Abb. 1.1c) erweist sich jedoch als problematisch. Sowohl direkte als auch indirekte Volumenvisualisierung leiden hier unter Aliasing-Artefakten.

Für die indirekte Volumenvisualisierung werden im Rahmen dieser Arbeit existierende Techniken, die eine Reduktion von Aliasing-Artefakten in aus binären Volumendaten extrahierten Oberflächenmodellen ermöglichen, systematisch verglichen und neue spezialisierte Methoden entwickelt.

### Dynamische Visualisierung

Eine weitere Herausforderung für die Visualisierung und Präsentation patientenspezifischer Daten in der chirurgischen Ausbildung und Interventionsplanung stellt die dynamische Visualisierung bzw. Animationen dar. In der (computerunterstützten) Ausbildung sind Bewegtbilder (Lehrvideos) weit verbreitet. Bei der Interventionsplanung finden solche dynamischen Visualisierungen unter anderem bei interdisziplinären Besprechungen wie Tumorboards Anwendung, um z.B. kurz und prägnant einen Überblick über die patientenspezifische Anatomie und die Lage von Tumoren zu vermitteln.

Mit Blick auf die dynamische Visualisierung von patientenspezifischen Daten stellt vor allem die Erzeugung von an die aktuellen Daten angepassten Animationen eine Herausforderung dar. Aus Effizienzgründen (hohe Fallzahlen) und aus Gründen der Standardisierung und Reproduzierbarkeit [Iserhardt-Bauer et al. 2001] ist es daher wünschenswert, Animationen sowie vordefinierte Ansichten nur einmal spezifizieren zu müssen und dann adaptiv auf jeden Patientendatensatz anwenden zu können.

Im Rahmen dieser Arbeit wird ein skriptbasierter Ansatz entwickelt, der die daten-unabhängige Spezifikation von Animationen für die Interventionsplanung und medizinische Ausbildung ermöglicht. Die Interpretation bzw. Erzeugung von Animationen erfolgt dabei adaptiv für die aktuell gegebenen Patientendaten, so dass ein Animationsskript standardisiert für andere Patienten eingesetzt werden kann.

## 1.2 Interaktion

Neben allen Vorteilen computerunterstützter dreidimensionaler Visualisierungen werden diese nur von einer Minderheit potentieller Nutzer und von diesen wiederum nur für eine geringe Teilmenge möglicher Aufgaben verwendet. Stattdessen bevorzugen Ärzte das „Lesen“ von CT- und MRT-Aufnahmen 2D-Schicht für 2D-Schicht.

Die vorliegende Arbeit verfolgt die Hypothese, dass eine Ursache für die unzureichende Nutzerakzeptanz in der mangelnden Unterstützung bei der Exploration und Navigation in dreidimensionalen Visualisierungen zu sehen ist. Dies betrifft sowohl die freie anwendergesteuerte Navigation und Exploration als auch die zielgerichtete Abarbeitung von Aufgaben in 3D-Visualisierungen.

### Rotation zur Exploration und Navigation

Eine grundlegende Technik zur freien Exploration und Navigation stellt die interaktive Rotation dreidimensionaler Szenen dar. Entsprechende Interaktionstechniken sind in medizinischen Workstations, CAD/CAM-Anwendungen und in jeder 3D-Visualisierungsbibliothek (siehe z.B. OpenInventor und VTK) zu finden. Die zur Verfügung gestellten Techniken unterscheiden sich dabei jedoch meist gravierend sowohl in den zugrunde liegenden Interaktionsmetaphern als auch in ihrem Interaktionsverhalten.

Wünschenswert für die Nutzerinteraktion und eine höhere Nutzerakzeptanz ist dabei die zielgerichtete Verwendung von einer für die aktuelle Anwendung geeigneten Rotationstechnik. Aussagen über die Eignung verschiedener Rotationstechniken für bestimmte Anwendungen sind jedoch weitgehend unklar, was eine geeignete Wahl behindert.

Im Rahmen der Arbeit werden Anforderungen an eine steuerbare 3D-Rotation erarbeitet. Daraufhin werden bestehende Techniken analytisch verglichen und empirisch evaluiert. Ergebnisse dieser Untersuchungen fließen dann in Empfehlungen zur Verwendung von Rotationstechniken zusammen.

### Aufgabenspezifische Navigationsunterstützung

Bei der zielgerichteten Abarbeitung von Aufgaben in dreidimensionalen Visualisierungen, zum Beispiel beim Inspizieren aller Objekte eines bestimmten Typs, müssen diese Objekte durch Navigation im dreidimensionalen Visualisierungsraum „abgearbeitet“ werden. Im Gegensatz dazu ist eine solche Aufgabe beispielsweise in 2D-Schichtbildern durch einfaches

sequentielles Absuchen einer Schicht nach der anderen über simples Weiterschalten zur jeweils nächsten Schicht möglich.

Folglich entsteht in dreidimensionalen Visualisierungen ein Navigationsoverhead, dessen Reduktion durch die Unterstützung von Navigationsaufgaben wünschenswert erscheint.

Hierzu wird der im Rahmen der vorliegenden Arbeit entwickelte adaptive skriptbasierte Ansatz für die Erzeugung dynamischer Visualisierungen eingesetzt, um vordefinierte Navigationsaufgaben „on the fly“ über Animationen zu realisieren.

## 1.3 Ausbildung und Training

Computerunterstützung (bzw. Computer Based Training CBT) bei der chirurgischen Ausbildung findet ergänzend zu herkömmlichen Lern- und Lehrmitteln Anwendung. Hierbei können CBT-Systeme alle drei grundlegenden Komponenten: *theoretische Chirurgie*, *klinische Chirurgie* und *Operationstechniken* der Chirurgieausbildung unterstützen [Mehrabi et al. 2000].

Die Attraktivität und das Potential von CBTs liegt hierbei sowohl in interaktiv explorierbaren 3D-Visualisierungen als auch im gefahrlosen Ausprobieren und Trainieren von Therapieentscheidungen oder Eingriffen. So können z.B. mit Chirurgesimulatoren Eingriffe und Operationstechniken geübt sowie allgemein erforderliche motorische Fertigkeiten trainiert werden (siehe z.B. [Delingette und Ayache 2005; Suzuki et al. 2007]).

Der Fokus dieser Arbeit liegt auf dem Erlernen und Trainieren von Interventionsplanung und Entscheidungsfindung im Rahmen der *klinischen Chirurgie*. Dies soll das gefahrlose Ausprobieren und Testen verschiedener Therapiealternativen ermöglichen und die Planung neuartiger Therapien erlernbar und somit breiter klinisch verfügbar machen. Die didaktische und systemtechnische Konzeption entsprechender Systeme stellt hierbei die spezielle Herausforderung dar.

Alle bisher angerissenen Aspekte der interaktiven und dynamischen Visualisierung in der chirurgischen Ausbildung und Interventionsplanung fließen dabei gemeinsam in CBT-Systeme für das Erlernen und Trainieren chirurgischer Interventionsplanung ein. Die Anforderungen an die interaktive und dynamische Visualisierung werden hierbei durch didaktische Ansprüche noch verstärkt.

## 1.4 Fragestellung und Aufgaben

Ausgehend von diesen Vorbetrachtungen befasst sich die vorliegende Arbeit mit folgenden Fragestellungen:

- Welche Artefakte können in Oberflächenmodellen aus segmentierten radiologischen Bilddaten auftreten und welche Strategien können zur Reduktion dieser beitragen?
- Welche Glättungsverfahren von Oberflächenmodellen können effizient und mit bestmöglicher Genauigkeit zur Artefaktreduktion in Oberflächenmodellen patientenindividueller Bilddaten angewendet werden?
- Wie kann die Genauigkeit und Detailtreue der Rekonstruktion artefaktfreier Oberflächenmodelle gesteigert werden?



- Wie kann der Benutzer bei der Navigation in und beim Begreifen von komplexen 3D-Szenen unterstützt werden?
- Wie kann die dynamische Visualisierung patientenindividueller 3D-Szenen standardisiert und automatisiert werden?
- Wie können die betrachteten Aspekte der interaktiven und dynamischen Visualisierung die chirurgische Ausbildung und Interventionsplanung unterstützen?

Die Beantwortung dieser Fragestellungen erfolgt mit Fokus auf die chirurgische Ausbildung und Interventionsplanung. Alle Ergebnisse finden dabei gemeinsam Anwendung in einem im Rahmen dieser Arbeit konzipierten und prototypisch umgesetzten Lernsystem zur Interventionsplanung in der Leberchirurgie.

Die konkrete Einbindung der Techniken zur dynamischen Visualisierung von und Interaktion in 3D-Daten in den gewählten Anwendungsrahmen trägt dabei dazu bei, die grundlegenden Konzepte und Techniken mit gewonnenem Anwenderfeedback und den gesammelten Erfahrungen für andere Anwendungen verallgemeinern zu können.

## 1.5 Gliederung

Die Gliederung der Arbeit richtet sich grundlegend nach einem Bottom-Up-Ansatz. Hierbei werden Konzepte und Arbeiten im Detail entwickelt, die zusammen zu anspruchsvollen Lösungen in der computerunterstützten chirurgischen Ausbildung und Interventionsplanung beitragen.

**Kapitel 2** betrachtet bestehende Interventionsplanungs- und Ausbildungssysteme mit Fokus auf die Erzeugung von Oberflächenmodellen dreidimensionaler Patientenanatomie, die Generierung standardisierter dynamischer Visualisierungen und die effiziente Navigation in 3D-Visualisierungen. Zusätzlich werden grundlegende Begriffe und nötiges medizinisches Wissen für die Interventionsplanung in der Leberchirurgie eingeführt.

**Kapitel 3** legt die Probleme bei der Oberflächenextraktion aus binär segmentierten Volumendaten dar und erläutert Zielsetzung und Herangehensweise dieser Arbeit zur Artefaktreduktion in medizinischen Oberflächenmodellen. Gleichzeitig werden nötige Grundlagen zur verwendeten Notation, Krümmungsberechnung und zu Datenstrukturen dargelegt.

**Kapitel 4** diskutiert Strategien zur Artefaktreduktion in der Oberflächenextraktionspipeline am Stand der Technik in den verschiedenen Bereichen. Dabei wird auf die Eignung existierender Techniken in den unterschiedlichen Schritten der Oberflächenextraktionspipeline eingegangen.

**Kapitel 5** geht der Frage nach, welche Glättungsverfahren in der klinischen Praxis effizient und mit bestmöglicher Genauigkeit zur Artefaktreduktion in Oberflächenmodellen aus radiologischen Bilddaten genutzt werden können. Hierzu werden medizinische Oberflächenmodelle klassifiziert und Glättungsempfehlungen für jede der Klassen erarbeitet. Gleichzeitig werden offene Probleme dargelegt, derer sich die nachfolgenden Kapitel annehmen.

**Kapitel 6** betrachtet Verfahren zur voxelbegrenzten Oberflächennetzglättung. Gleichzeitig wird die *Diamond Constraint* Glättung für medizinische Oberflächenmodelle entwickelt, die eine korrekte Innen-Außen Repräsentation der ursprünglichen Volumendaten garantiert.

**Kapitel 7** betrachtet Verfahren zur Approximation von Oberflächenmodellen mit Blick auf eine möglichst genaue Rekonstruktion segmentierter Gefäßstrukturen. Hierbei wird ein auf die Extraktion von Oberflächenmodellen von Blutgefäßen spezialisiertes Verfahren eingeführt.

**Kapitel 8** betrachtet Techniken der direkten Manipulation zur interaktiven Exploration von medizinischen Oberflächenmodellen. Hierzu werden Techniken zur mausbasierten Rotation von 3D-Szenen detailliert untersucht und es wird der Frage nachgegangen, wie Benutzer bei der Navigation in komplexen 3D-Szenen unterstützt werden können.

**Kapitel 9** untersucht und entwickelt Techniken zur dynamischen Visualisierung und Interaktion, wobei Techniken zur Standardisierung und Automatisierung von dynamischen Visualisierungen patientenindividueller Daten im Vordergrund stehen.

**Kapitel 10** bespricht und entwickelt ein Lernsystem für die Interventionsplanung in der Leberchirurgie. Hierbei fließen die Erfahrungen und entwickelten Techniken aus den vorangegangenen Kapiteln in die Entwicklung ein.

**Kapitel 11** fasst die Ergebnisse dieser Arbeit noch einmal zusammen und weist über die Benennung offener Fragen auf weiterführende Themen hin.

# Computerunterstützte Interventionsplanung und chirurgische Ausbildung

---

Dieses Kapitel betrachtet relevante Aspekte der interaktiven und dynamischen Visualisierung im Anwendungskontext der Interventionsplanung und chirurgischen Ausbildung. Hierfür werden bestehende Interventionsplanungs- und Ausbildungssysteme mit Fokus auf die Erzeugung von Oberflächenmodellen dreidimensionaler Patientenanatomie, die Generierung standardisierter dynamischer Visualisierungen und die effiziente Navigation in 3D-Visualisierungen betrachtet.

## 2.1 Vorbetrachtungen

Die computerunterstützte Interventionsplanung spielt vor allem dann eine entscheidende Rolle, wenn:

- (a) komplexe dreidimensionale Lagebeziehungen berücksichtigt, bewertet und/oder manipuliert werden müssen,
- (b) eine Verbesserung der Qualität (selten auch der Quantität) durch die sorgfältige Planung zu erwarten ist und
- (c) die Eingriffe planbar erfolgen (keine Notfallversorgung) und ausreichend Zeit zur sorgfältigen Planung zur Verfügung steht.

Im Folgenden werden Anwendungsbeispiele aus der Viszeralchirurgie (Leberoperationsplanung), Hals-Nasen-Ohren-Chirurgie (Neck Dissections) und Mund-Kiefer-Gesichts-Chirurgie (Osteotomieplanung) als repräsentative Auswahl komplexer Interventionsplanungen mit Computerunterstützung näher vorgestellt.

Computerunterstützung für Ausbildung und Training (*Computer Based Training, CBT*) kann auf eine über 80-jährige Geschichte [Pressey 1926] zurückgeführt werden. Attraktiv war die Computerunterstützung in den Anfängen vor allem durch direktes Feedback und damit beobachtete Verstärkung gewollten Verhaltens [Skinner 1968; McCue 2005]. In der weiteren Entwicklung kamen Möglichkeiten der multimedialen Präsentation und der weltweiten Vernetzung der Systeme (Web Based Training, WBT) hinzu.

Aktuelle CBTs für die medizinische Ausbildung werden vor allem im curricularen Umfeld des Medizinstudiums nach Novellierung der Ärztlichen Approbationsordnung (ÄAppO) vorangetrieben. Der große Vorteil der CBT-Systeme im Vergleich zu konventionellen Lehrmethoden ist hier besonders aus ökonomischer Sicht [Leven et al. 2006] gegeben. Mögliche Lernvorteile durch Computerunterstützung für Ausbildung und Training gegenüber Büchern und anderen Lernmethoden sind noch nicht komplett untersucht.

Die Betrachtung in den folgenden Abschnitten beschränken sich dabei auf relevante und im wissenschaftlichen Umfeld gut dokumentierte Lernsysteme. Als relevant werden jene Systeme eingestuft, die eine für die computerunterstützte Interventionsplanung wichtige Berücksichtigung, Bewertung und/oder Manipulation komplexer dreidimensionaler Lagebeziehungen unterstützen und/oder in der chirurgischen Ausbildung bereits Anwendung finden bzw. fanden.

Ein Verzeichnis mit aktuell mehr als 1900 verfügbaren multimedialen Lehr- und Lernmodulen in der Medizin kann auf dem *Learning Resource Server Medizin* [LRSMed 2007] gefunden werden.

## 2.2 Leberoperationsplanung

Ein herausragendes Beispiel für eine komplexe computerunterstützte Interventionsplanung, die bereits umfassend in der klinischen Praxis Anwendung findet, betrifft die Planung von Leberoperationen.

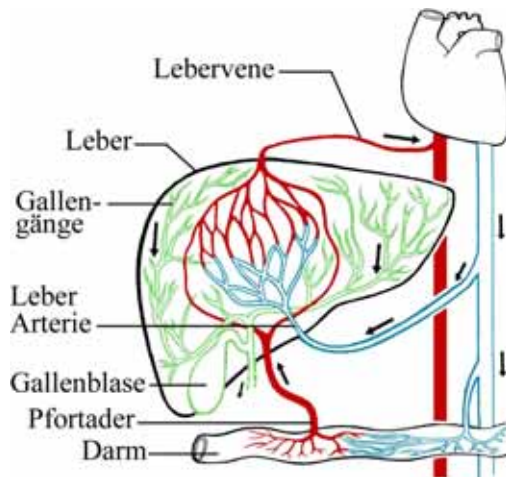
Die spezialisierten Softwareassistenten LIVERANALYZER (früher HEPAVISION) zur Segmentierung und Bildanalyse von CT-Leberdatensätzen und MEVISLIVER (früher INTERVENTIONPLANNER) von [MeVis 2007] als genereller Assistent zur 3D-Visualisierung, Resektions- und Zugangsplanung sowie zur Dokumentation wurden dabei bis 2006 von über 36 Krankenhäusern in mehr als 2000 Fällen zur Operationsplanung eingesetzt [Preim und Bartz 2007]. Darunter waren ca. 700 Leber-Tumor-Operationen, 950 Leberlebendspenden und 180 ähnliche Operationen, zum Beispiel der Niere oder Bauchspeicheldrüse.

### 2.2.1 Medizinischer Hintergrund

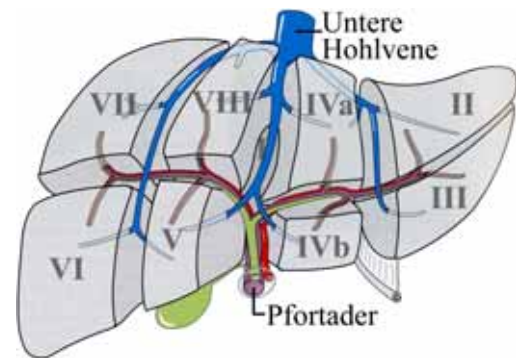
Die Planung von Eingriffen an der Leber ist durch die komplexe Anatomie der intrahepatischen Gefäßsysteme (Leberarterie, Pfortader, Lebervene und Gallengänge siehe Abb. 2.1) eine besondere Herausforderung. So muss bei Operationen sichergestellt werden können, dass das gesunde im Patienten verbleibende Lebergewebe noch ausreichend von allen Gefäßsystemen ver- und entsorgt wird. Anhand des Verlaufs von Pfortader und Lebervene kann die Leber grob in neun autonom versorgte Gebiete unterteilt werden (Abb. 2.2). Diese Gebiete können dann vom Chirurgen einzeln herausgenommen bzw. verschont werden, ohne die Gefäßversorgung der zu erhaltenden Segmente zu gefährden. Gesunde Lebern weisen jedoch eine hohe Formvariation zwischen verschiedenen Menschen (siehe Abb. 2.3) auf. Durch krankhafte Veränderungen und vorherige Eingriffe können Form und Gefäßsystemanatomie zusätzlich stark von der Norm abweichen.

Das Hauptproblem der Leberchirurgie besteht darin, die patientenindividuelle Gefäß- und Segmentanatomie zu berücksichtigen. Besonders schwer fällt dies, da die Segmentgrenzen weder an Landmarken auf der Leber noch in radiologischen Aufnahmen detektiert werden können. Entsprechend müssen diese Informationen in der Planung abgeschätzt werden, um die verschiedenen Eingriffe, die im Folgenden kurz erläutert werden, zu unterstützen.

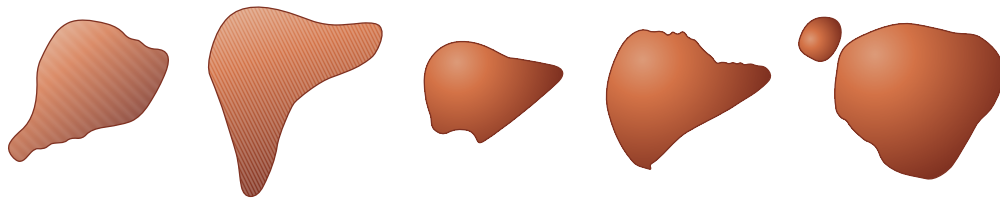
**Resektion von Lebertumoren:** Das Herausschneiden (Resezieren) von gutartigen (benignen) und bösartigen (malignen) Tumoren ist die etablierteste Therapie, mit der besten Langzeitüberlebensrate bei Leberkarzinomen. Ziel der Resektion ist es, die Tumore vollkommen zu entfernen. Bei malignen Tumoren wird zusätzlich darauf geachtet, diese mit einem ausreichend großen tumorfreien Rand zu resezieren, um ein erneutes Tumorwachstum und



**Abbildung 2.1:** Blutgefäßsysteme und Gallengangssystem der menschlichen Leber. (modifiziert aus [Diagram Group 1983])



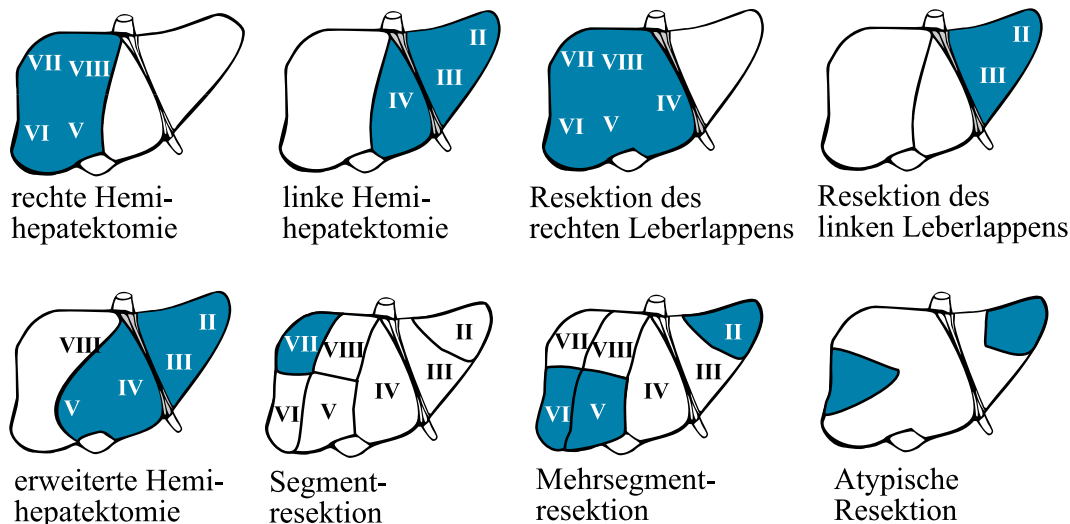
**Abbildung 2.2:** Einteilung der Leber in Segmente entsprechend dem Gefäßverlauf nach Couinaud. (modifiziert aus [Benninghoff und Drenckhahn 2002])



**Abbildung 2.3:** Beispiele möglicher Formvariationen der gesunden Leber. (modifiziert aus [Köckerling und Waclawiczek 1999])

weiteres Streuen des Tumors (Bildung von Satelliten/Metastasen) weitgehend auszuschließen. Ein tumorfreier Sicherheitsrand von mindestens 10mm hat dabei einen bestätigt positiven Einfluss auf die Langzeitüberlebensrate. Entsprechend ist in der Planung die räumliche Lage und Ausdehnung des/der Tumore zu angrenzenden Strukturen relevant. Beim Heraus-schneiden wird sich dann an den Segmenten orientiert und alle betroffenen Segmente entfernt (siehe Abb. 2.4).

**Ablation von Lebertumoren:** Ist die Resektion von Lebertumoren aufgrund ihrer Lage, Größe, anderer Schädigungen der Leber (Zirrhose) oder der Verfassung des Patienten nicht möglich, stellt die lokale Ablation der Tumore eine mögliche Alternative dar. Hierbei wird eine Ablationsnadel meist von außen durch die Haut des Patienten gestochen (Abb. 2.5) und bis in den Tumor vorgeschoben. Dort wird dieser durch Hitze bei *Radiofrequenz-Ablation (RF)* und *Laserinduzierter Ablation (LITT)* koaguliert/zerstört. Gegenstand der präoperativen Planung bei der Tumorablelation sind die Bestimmung der idealen Lage des Kopfes der Ablationsnadel im Tumor sowie die Wahl eines günstigen Zugangsweges, auf dem die Ablationsnadel, ohne andere Organe zu verletzen, bis zum Tumor geschoben wird. Die ideale Lage ist dabei hauptsächlich abhängig von Tumorgöße, -form, Gewebeeigenschaften und der Nähe zu anderen Organen bzw. großen Blutgefäßen. Diese Faktoren bestimmen maßgeblich, ob ein Tumor ausreichend durch die Ablation zerstört werden kann und ob andere wichtige Bereiche in Mitleidenschaft gezogen werden könnten. Zusätzlich gilt es zu berücksichtigen, ob der Zugangsweg der Ablationsnadel von außen bis in den Tumor nicht zur Gefährdung



**Abbildung 2.4:** Ausgewählte Varianten der Leberresektion, differenziert nach Lage und Ausmaß. (modifiziert aus [Priesching 1986])

anderer Organe oder großer Blutgefäße führt – bzw. z.B. durch Knochen vollkommen verspermt ist. Mathematische Modelle und Simulationen sollen hierbei die Platzierungs- und Zugangsplanung unterstützen.

**Leberlebendspenden:** Bei einigen Erkrankungen (nicht resezierbare Tumore, Hepatitis C, Leberzirrhose, ...) der Leber stellt die Transplantation einer Spenderleber die einzig angemessene Therapie dar. Um dem Mangel an Spenderorganen vor allem bei Kindern zu entgegenen, wurde 1989 die chirurgische Möglichkeit der Leberteiltransplantation (*Split-Liver Transplantation*) von lebenden Spendern eingeführt [Raia et al. 1989; Broelsch et al. 1990]. Besonderer Planungsbedarf besteht hierbei darin, eine passende Spenderleber zu ermitteln. Neben vielen anderen klinischen Parametern muss hierbei vor allem die Größe und Gefäßanatomie der Spenderleber so gegeben sein, dass die geteilte Leber sowohl Spender als auch Empfänger ausreichend versorgen kann. Besonderes Augenmerk gilt hierbei den Gefäßästen und der Entscheidung welche Äste im Spender verbleiben, transplantiert werden, bzw. rekonstruiert werden müssen. Chirurgisch kann dann ein Teil der Spenderleber unter Berücksichtigung der Lebersegmente ähnlich der *linken Hemihepatektomie* oder der *Resektion des linken Leberlappens* (siehe Abb.2.4) vorgenommen werden.

### 2.2.2 Computerunterstützte Planung

Die Computerunterstützung bei der Leberoperationsplanung besteht dabei grundlegend aus Segmentierung, Quantifizierung, Visualisierung, interaktiver Planung, Simulation und Dokumentation (siehe Preim und Bartz [2007] für einen Überblick).

**Segmentierung:** Zunächst müssen Leber, Tumor(e) und intrahepatische Blutgefäße segmentiert werden. Im auf Leberdatensätze spezialisierten Segmentierungsassistenten LIVER-ANALYZER werden dazu semiautomatische Verfahren wie *Live-Wire* und *Region Growing* sowie bei der Abgrenzung kontrastschwacher Strukturen manuelle Verfahren verwendet. Die Segmentierungsergebnisse werden als binäre Segmentierungsmasken gespeichert und dienen dann als Ausgang für die weitere Analyse und Quantifizierung.

**Quantifizierung:** Quantifiziert werden müssen zunächst die Lebersegmente und die Risikogebiete um die Tumore. Zur Abschätzung der Lebersegmente und Segmentgrenzen können anhand der Verzweigungsstruktur der segmentierten Blutgefäße autonome Versorgungsgebiete approximiert werden (siehe Abb. 2.6) [Selle et al. 2002]. Risikogebiete um Tumore können über den Abstand der Tumore zu Blutgefäßen und die von ihnen versorgten Gebiete (siehe Abb. 2.7) abgeschätzt werden. Weiterhin ist bei der Planung von Resektionen oder Leberlebenspenden möglicherweise unterversorgtes Gewebe zu quantifizieren. Dieses kann über die Berechnung der Versorgungsgebiete aller nicht mehr verbundenen Blutgefäße im nicht resezierten Teil der Leber, wie in Abbildung 2.8 illustriert, ermittelt werden. Entsprechend lässt sich auch das verbleibende gesunde Lebervolumen ermitteln.

**Visualisierung und Interaktion:** Die Planung wird durch Möglichkeiten zum Einzeichnen und Definieren einer Resektionsebene sowie zum Platzieren von Applikatornadeln in 2D- und 3D-Ansichten angeboten (Abb. 2.9). Für die dreidimensionale Darstellung kommen dabei hauptsächlich Oberflächenmodelle der zuvor segmentierten Strukturen zum Einsatz. Die 3D-Szenen können durch interaktives Rotieren erkundet werden. Zusätzlich können die Resektionsebene und die Applikatorplatzierung interaktiv manipuliert werden.

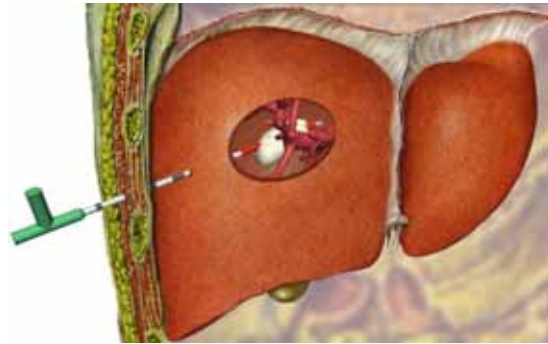
**Simulation:** Um bei der Tumorablationsplanung verlässliche Aussagen über die durch Ablation erreichte Gewebeerstörung treffen zu können, muss diese über Simulation der Wärmeausbreitung im Gewebe simuliert werden. Für eine aussagekräftige Simulation sind alle intrahepatischen Strukturen zu segmentieren und ihre physikalisch-thermischen Eigenschaften zu bestimmen. Besonderen Einfluss auf das erreichbare Schädigungsvolumen haben dabei Blutgefäße, die umliegendes Gewebe stark abkühlen können. Abbildung 2.10 stellt Simulationsergebnisse mit und ohne Berücksichtigung kühlender Blutgefäße dar.

## 2.3 Planung von Neck Dissections

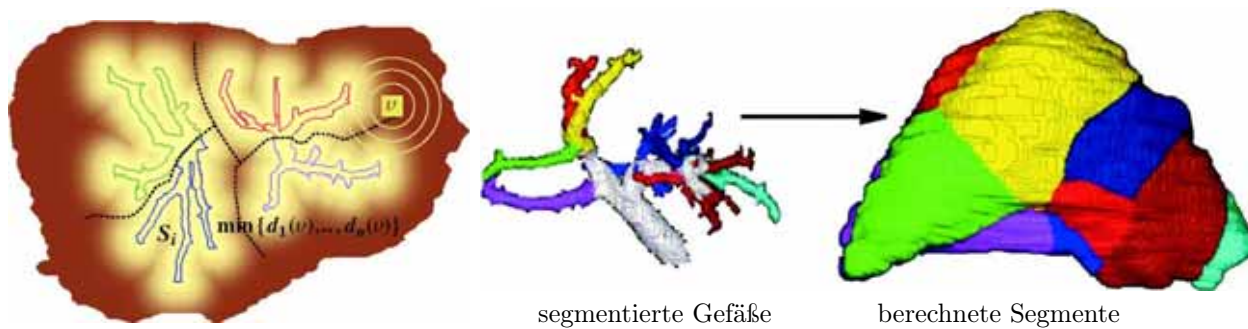
Bei der Planung von Halslymphknotenausräumungen (*Neck Dissections*) sind nach einem Tumorbefall im Halsbereich [Kuntz und Weymuller 1999] zur angemessenen und schonenden Entfernung von Lymphknoten unter anderem deren dreidimensionale Größe und räumliche Nähe zu anderen Risikostrukturen (z.B. durch das Hineinwachsen in andere Strukturen (Infiltration)) therapieentscheidend.

Die komplexe Anatomie des Halses, der ähnlich einem „Kabelschacht“ wichtige Blutgefäße, Nerven, Muskeln, Luft- und Speiseröhre bündelt, erschwert dabei die Beurteilung der räumlichen Verhältnisse. Hinzu kommen ästhetische und funktionelle Aspekte, da bei einer radikalen Lymphknotenausräumung unter anderem Muskeln des Halses entfernt werden müssen, was zu einer eingeschränkten Halsbewegung und ästhetischen Veränderung des Halses führen kann. Entsprechend bestehen für diese Anwendung Bestrebungen durch computerunterstützte Planung unnötige bzw. unnötig radikale Ausräumungen zu vermeiden und eventuell erforderliche plastische Rekonstruktionen im Voraus abschätzen zu können [Strauß et al. 2006].

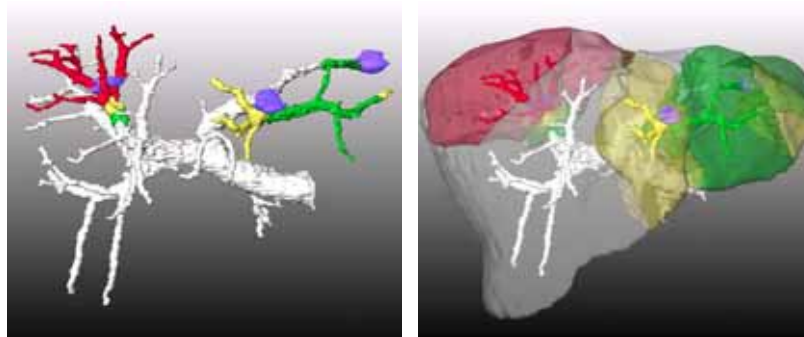
**Segmentierung und Quantifizierung:** Für die computerunterstützte Planung müssen zunächst alle relevanten Strukturen (Tumor(e), Lymphknoten, Blutgefäße, Nerven, ...) im Halsbereich identifiziert und segmentiert (abgegrenzt) werden. Die Segmentierung erfolgt dabei mit auf die jeweiligen Strukturen angepassten Segmentierungsverfahren (Schwellwert, Re-



**Abbildung 2.5:** Minimal-invasives Einbringen einer Radiofrequenz-Ablationsnadel in einen Lebertumor, um diesen lokal zu zerstören.

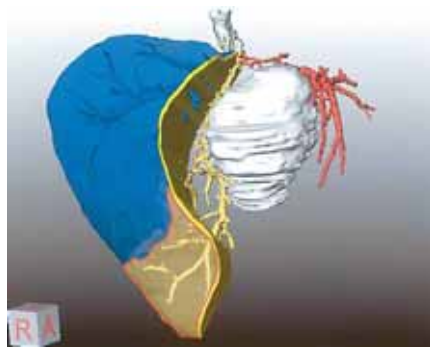


**Abbildung 2.6:** Approximation der Lebersegmente über die euklidische Distanz zu den Gefäßen. Links: Minimum der Distanzfunktionen aller Gefäßäste ergibt die Segmentgrenzen. Mitte und Rechts: Approximierte Segmente ausgehend von segmentiertem Gefäßbaum. (modifiziert aus [Selle et al. 2002])

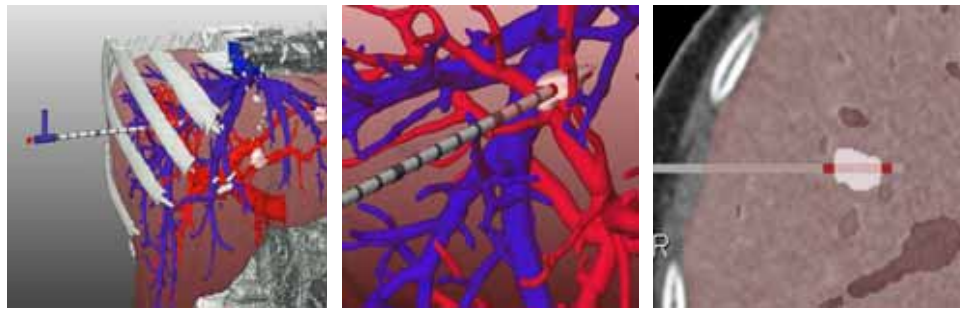


**Abbildung 2.7:** Approximation von Risikogebieten auf der Pfortader bei drei Tumoren. Rote, gelbe und grüne Gefäße und Versorgungsgebiete repräsentieren dabei betroffene Strukturen im 5-, 10- und 15-mm Sicherheitsabstand. [Preim und Bartz 2007]

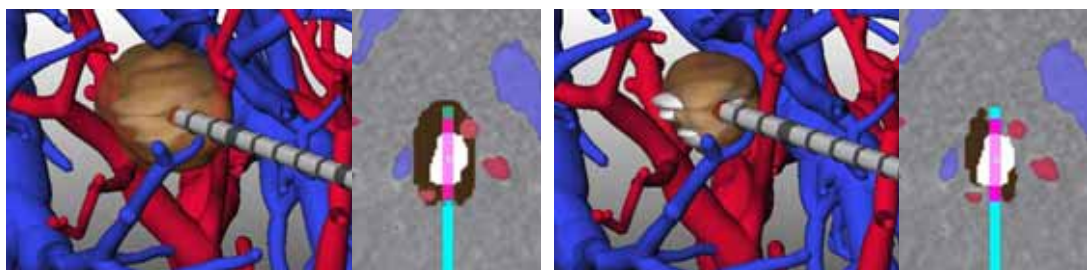




**Abbildung 2.8:** Risikoanalyse unterversorgter Leberregionen bei Resektion eines zentralen Tumors. Die rot umrandete Region würde nach Resektion des Tumors nicht mehr von der mittleren Lebervene (gelb) versorgt werden. [Lang et al. 2005a]



**Abbildung 2.9:** Synchronisierte Ansichten bei der Tumorablationsplanung, um die genaue Lage in Bezug auf Rippen und Nachbarorgane, intrahepatische Blutgefäße und den Tumor beurteilen zu können. [Preim und Bartz 2007]

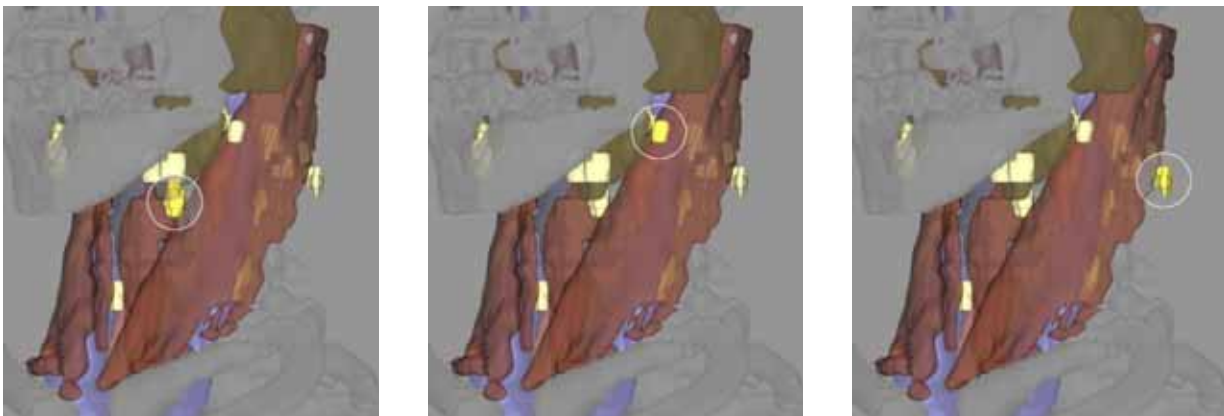


**Abbildung 2.10:** Simulierte Schädigungsvolumina (schwarz) bei Tumorablation in 2D- und 3D-Ansicht. Links: ohne und Rechts: mit Berücksichtigung des Kühleffektes angrenzender Blutgefäße. (modifiziert aus [Preim und Bartz 2007])

gionenwachstum, interaktive Watershed-Transformation, semi-automatisches Live-Wire und Feder-Masse-Modelle) [Cordes et al. 2006a; Dornheim et al. 2006]. Die Segmentierungsergebnisse werden in Form von binären Segmentierungsmasken gespeichert. Daraufhin können Anzahl, Größe, räumliche Nähe sowie mögliche Infiltrationen der Strukturen quantifiziert werden.

**Visualisierung:** Zur Unterstützung der Entscheidungsfindung und Therapieplanung werden diese Ergebnisse zusammen mit der Anatomie des Patienten dreidimensional visualisiert. Zur effektiven dreidimensionalen Visualisierung werden dazu Oberflächenmodelle der segmentierten Strukturen erzeugt, die dann systematisch visuell inspiziert werden können. Hierzu wird zum Beispiel eine separate Exploration von allen segmentierten Lymphknoten, beim größten (pathologisch auffälligstem) beginnend, angeboten [Krüger et al. 2005b] (Abb. 2.11). Um die Sichtbarkeit des aktuell inspizierten Lymphknoten zu gewährleisten, kommen Cutaway-Techniken zum Einsatz.

**Standardisierte Präsentation:** Zur Präsentation und Dokumentation der Segmentierungs- und Quantifizierungsergebnisse werden standardisierte Ansichten aus definierten Richtungen sowie Rotationen um die Körperachse generiert. Im entwickelten Planungssystem NECK SURGERY PLANNER [Janke et al. 2006] können die Strukturen zusätzlich interaktiv gedreht und erkundet werden.



**Abbildung 2.11:** Sukzessive Hervorhebung vergrößerter Lymphknoten für die Planung von Halslymphknotenausräumungen. [Krüger et al. 2005b]

## 2.4 3D-Osteotomieplanung

Bei der Planung von knochenverlagernden Operationen (Osteotomie) in der Mund-Kiefer-Gesichts-Chirurgie (MKG-Chirurgie) stehen vermehrt Anforderungen an Ästhetik und Funktionalität im Vordergrund. Bei Knochenverlagerungen, z.B. bei Fehlbildungen oder Frakturen der Kieferknochen, wird unter ästhetischen Gesichtspunkten ein bündiges Abschließen von Unter- und Oberkiefer sowie Symmetrie und Normangepasstheit von Gesicht und Schädel angestrebt. Mit Blick auf die Funktionalität sollen entsprechende Eingriffe zusätzlich die uneingeschränkte Kaubewegung des Kiefergelenks herstellen bzw. erhalten.

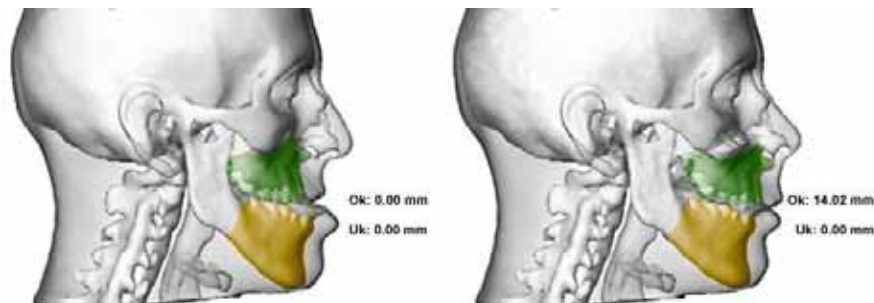
Um Aspekte der Ästhetik und Funktionalität planbar zu gestalten und komplizierte Knochenverlagerungen gefahrlos ausprobieren zu können, ist eine Computerunterstützung (siehe [Zachow 2005]) sinnvoll.

**Segmentierung:** Zur computerunterstützten Therapieplanung von Knochenverlagerungen in der MKG-Chirurgie kommen bei der Segmentierung neben der Grobsegmentierung mittels Schwellwerten auch automatische Kantenverfolgungs-, Regionenwachstums- und Level-Set-Verfahren zum Einsatz [Zachow 2005].

**Oberflächenextraktion und -aufbereitung:** Aus den Segmentierungsergebnissen werden daraufhin polygonale 3D-Oberflächenmodelle über spezielle Verfahren zur Behandlung von nicht-mannigfaltigen Oberflächen erzeugt [Hege et al. 1997]. Diese Modelle werden daraufhin mit lokal steuerbaren Fehlergrenzen vergrößert und für die Erzeugung von Volumengittern optimiert [Stalling et al. 1999].

**Visualisierung und Interaktion:** Zur Planung der Knochenverlagerung werden Interaktionstechniken angeboten, um das Zerteilen und Repositionieren von Knochen oder Knochenanteilen virtuell definieren zu können [Zachow et al. 2006].

**Simulation:** Um Aussagen über die zu erwartende Funktionalität und Ästhetik treffen zu können, werden geeignete Volumengitter generiert und die resultierende Bewegung von betroffenen Gelenken sowie die Verformung des Weichgewebes simuliert [Gladilin et al. 2004] (Abb. 2.12).



**Abbildung 2.12:** Prädiktion der Therapieergebnisse nach Knochenverlagerung durch Weichgewebesimulation. (modifiziert aus [Zachow et al. 2006])

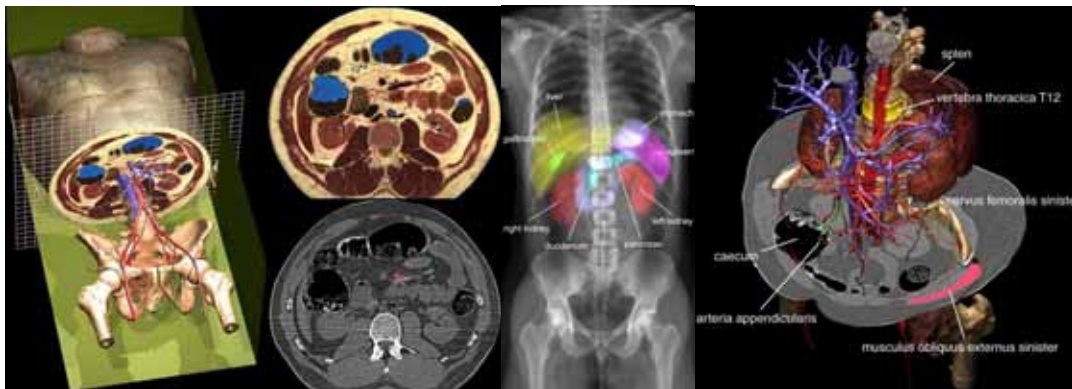
## 2.5 Computerunterstützte Anatomieausbildung

Die Anatomieausbildung ist eine wichtige Grundlage für die Chirurgie und ist konfrontiert mit der Vermittlung komplexer dreidimensionaler Lagebeziehungen.

Im Folgenden werden Computerapplikationen zur Unterstützung der Anatomieausbildung vorgestellt. Der Fokus liegt hierbei auf Systemen mit interaktiven dreidimensionalen Visualisierungen. Systeme, die nur auf gescannten Zeichnungen basieren und/oder eine Eins-zu-eins-Umsetzung konventioneller Bücher sind, werden nicht betrachtet.

### 2.5.1 VoxelMan

Der VOXELMAN ist ein in erster Generation auf segmentierten MRT-Bilddaten des Kopfes und in zweiter Generation auf dem *Visible-Human*-Datensatz beruhendes Anatomielernsystem. Für eine überzeugende Qualität der Segmentierungsergebnisse wurden diese zusätzlich per Hand mit dem *Volume Editor* nachbearbeitet und schlecht zu segmentierende Blutgefäße und Nerven komplett modelliert [Pommert et al. 2001]. Alle Visualisierungen im System wurden aufwändig qualitativ hochwertig vorgerendert und können interaktiv, z.B. durch Verschieben einer clipping plane, erkundet werden (Abb. 2.13). Auf dem VOXELMAN beruhende Simulationen von Endoskopie, 2D-Röntgenaufnahmen und Ultraschalluntersuchungen liefern verschiedene Bildmodalitäten für verschiedene Aspekte der Anatomieausbildung [Pommert et al. 2006]. Eigene Ansichten bzw. die Manipulation durch Setzen von Resektionslinien kann durch die Beschränkung auf vorgerenderte Grafiken in den publizierten Lernsystemen nicht realisiert werden.

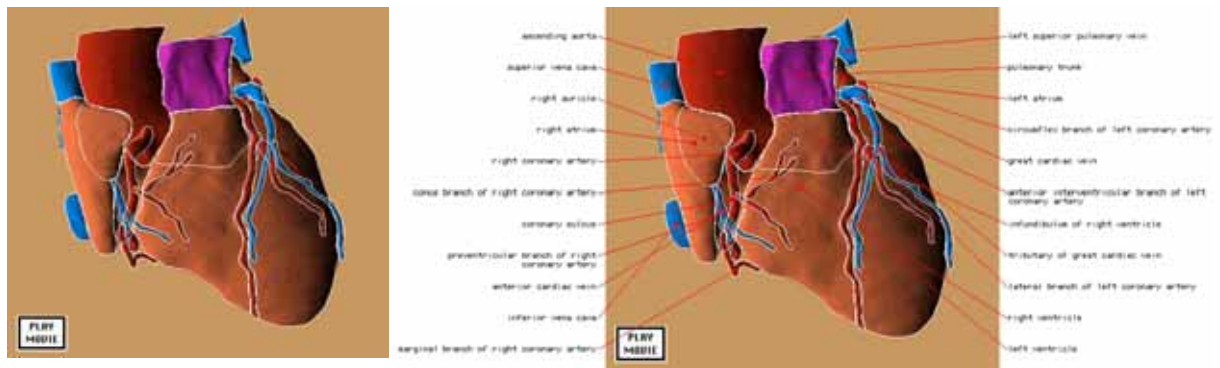


**Abbildung 2.13:** Explorations- und Visualisierungsmodi beim VOXELMAN: (v.l.n.r.) 3D-Visualisierung mit interaktiver Clipping-Plane, zugehörige Schnittebene des fotografischen und CT-Datensatzes, simulierte Röntgenaufnahme und kombinierte Visualisierung von 3D-Modell und CT-Schicht. [Preim und Bartz 2007]

### 2.5.2 Digital Anatomist

Vorgerenderte Ansichten und Animationen sind die Grundlage des DIGITAL ANATOMIST [Brinkley und Rosse 1997; Brinkley et al. 1999]. In den Animationen werden Strukturen sukzessiv eingeblendet, rotiert, vergrößert, in Explosionsdarstellungen zerlegt und beschriftet. Die interaktive Exploration der durch Oberflächenmodelle dargestellten Anatomie beschränkt sich jedoch auf das Abspielen der Videos und das selektive Einblenden von Umrissen



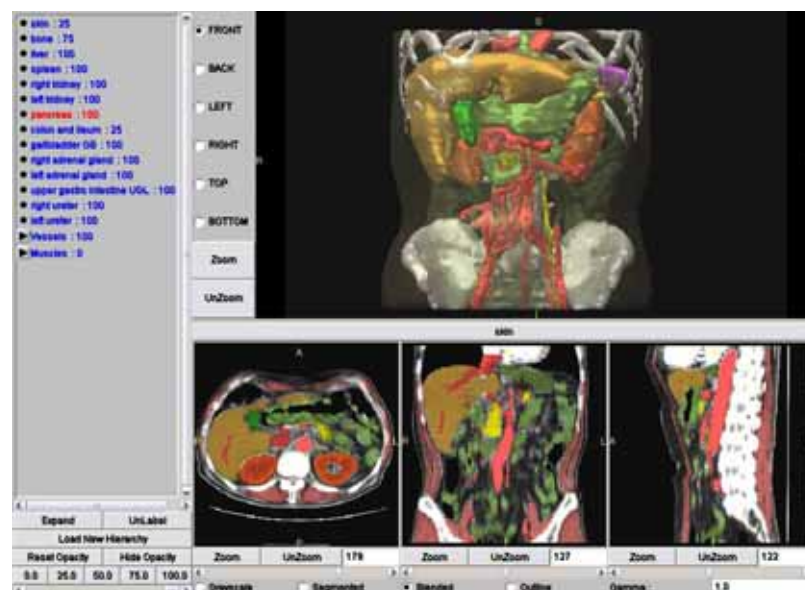


**Abbildung 2.14:** Screenshot vom DIGITAL ANATOMIST: Thoracic Viscera. Alle Visualisierungen können mit Umrissen (links) und Beschriftungen (rechts) generiert werden. [Structural Informatics Group 2007]

und Beschriftungen (Abb. 2.14). Ergänzend enthält das System eine *drill and practice* Komponente zum Lokalisieren und Benennen von Strukturen in Form eines Quiz.

### 2.5.3 AnatomyBrowser

Der ANATOMYBROWSER setzt beim Rendering auf einen zweistufigen Prozess. Hierzu werden im ersten Schritt Ansichten aller Strukturen mittels high-end Surfacerendering vorgerendert. In einem zweiten (Compositing-) Schritt werden diese Ansichten dann zu einer Visualisierung zusammengefügt [Golland et al. 1999]. Dieser Ansatz ermöglicht mehr Freiheiten beim Explorieren der Ansichten, so kann z.B. die Transparenz von Strukturen trotz vorgerenderten Bildern frei eingestellt werden (Abb. 2.15). Die Manipulation der Oberflächenmodelle sowie das Wählen eigener Blickrichtungen ist abweichend von den vorberechneten Visualisierungen jedoch nicht möglich. Im System existiert eine Vielzahl unterschiedlicher Datensätze (überwiegend für den Kopf) [Kikinis et al. 1996], so dass mit dem ANATOMYBROWSER auch die vergleichende Exploration der Anatomie unterstützt wird.



**Abbildung 2.15:** Screenshot vom ANATOMYBROWSER: Abdomen. modifiziert aus [Surgical Planning Lab 2007]

### 2.5.4 ZoomIllustrator und 3D Puzzle

Der ZOOMILLUSTRATOR [Preim et al. 1996b, 1997] setzt auf das interaktive Rendern von hochau aufgelösten Oberflächenmodellen von Viewpoint Datalabs (Abb. 2.16). Techniken zur Integration von textuellen Beschreibungen und der dreidimensionalen Visualisierung ermöglichen eine unterstützte und dennoch freie Exploration (z.B. freies Rotieren, Zoomen, Ein- und Ausblenden) der Anatomie.

**3D Puzzle:** Das 3D PUZZLE baut auf der Datengrundlage des ZOOMILLUSTRATOR auf und beabsichtigt eine tiefere Auseinandersetzung der Anwender mit den dreidimensionalen Strukturen [Ritter et al. 2000, 2001]. Hierzu können 3D-Szenen aus einzelnen Strukturen ähnlich einem Puzzle zusammengesetzt werden (Abb. 2.17). Zur Unterstützung der Raumwahrnehmung und Navigation werden Stereo Rendering, Schattenprojektionen und 3D-Eingabegeräte verwendet. Die Unterstützung weiter gefasster Lernziele ist jedoch auf Basis des 3D Puzzle nicht möglich.

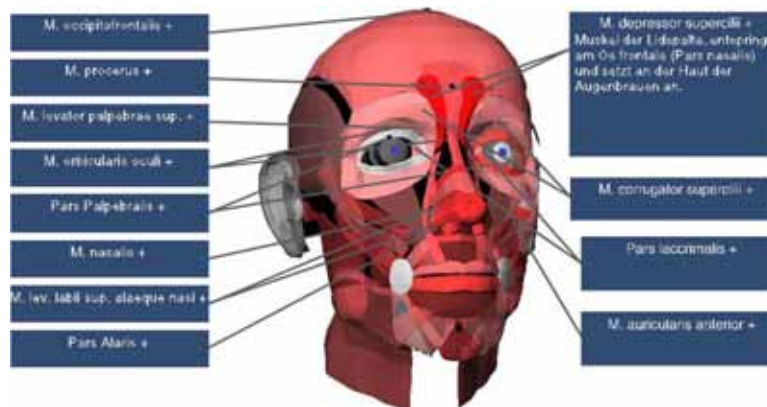


Abbildung 2.16: Screenshot vom ZOOMILLUSTRATOR. (modifiziert aus [Preim und Bartz 2007])

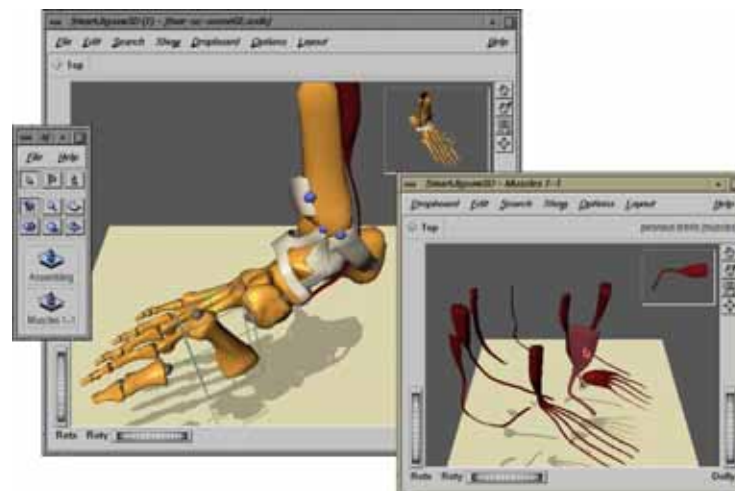


Abbildung 2.17: 3D PUZZLE: Das linke Fenster zeigt wie eine neue Struktur in das bis dahin zusammengesetzte Modell eines Fußes eingefügt wird. Im rechten Fenster liegen die noch einzufügenden Muskeln bereit. [Ritter et al. 2000]

## 2.6 Allgemeine CBTs in der Medizin

Der Vorteil der bisher betrachteten Lernsysteme gegenüber konventionellen Lernmitteln besteht weitestgehend in Möglichkeiten der interaktiven Exploration. So kann meist auf Basis eines „idealen“ 3D-Modells die Anatomie erkundet werden. Dabei werden aber weder Lernsteuerungsmechanismen, noch fallbasierte, patientenindividuelle Daten verwendet.

Es existiert jedoch eine Vielzahl an allgemeinen medizinischen Lernsystemen, die fallbasiertes und problemorientiertes Lernen (POL) ermöglichen. Unabhängig vom verwendeten Präsentationsmedium (z.B. lokale Installation, Client-Server, Webbrowser, ...) kommen dabei grundlegend starre Bildschirmseiten oder situierte Lernumgebungen zur Präsentation der Lerninhalte zum Einsatz. Repräsentativ für eine Vielzahl ähnlicher Systeme werden im Folgenden drei ausgesuchte CBTs in der Medizin vorgestellt.

### 2.6.1 CASUS

CASUS ist eine fallbasierte multimediale Lern- und Autorenplattform für die Aus- und Weiterbildung von Medizinern [Fischer et al. 1996]. Anwendungen auf Basis dieser Plattform existieren auch für andere Fachbereiche, wie z.B. Rechtswissenschaften, Anglistik oder Veterinärmedizin (siehe [Casus 2007]). Alle CASUS Anwendungen bestehen hauptsächlich aus aufeinander folgenden Bildschirmseiten. Jede Bildschirmseite kann hierbei aus einem Infotext-Bereich, einem audiovisuellen Präsentationsbereich, einem Frage-Antwort-Bereich sowie einem Feedbackbereich (siehe Abb. 2.18) bestehen. Mögliche Frage- und Antwortmodalitäten sind Multiple-Choice-, Unterstreichungs-, Reihenfolge- und Freitext-Aufgaben.

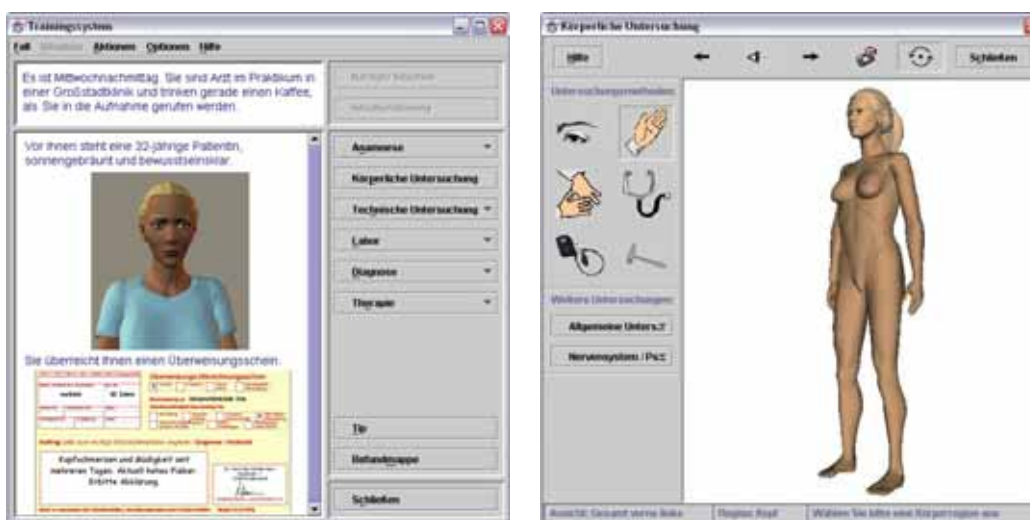
Der Fokus des Systems liegt hauptsächlich auf Lernanwendungen in der Diagnostik und allen damit verbundenen Entscheidungen. Sehr hilfreich ist hierbei die Möglichkeit, Differentialdiagnosen aufstellen und verschiedenen Symptomen zuordnen zu können (Abb. 2.18 rechts). Anwendung findet CASUS derzeit extensiv mit über 100.000 abgeschlossenen Fallsitzungen in 80 medizinischen Ausbildungseinrichtung der USA und Kanada (Zahlen Ende 2006) im *Computer-assisted Learning in Pediatrics Program (CLIPP)* [Fall et al. 2005; iInTIME 2007].



**Abbildung 2.18:** Pädriatischer Fall in CASUS. Links: typische Bildschirmseite mit Infotext, Bild/Video, MC-Fragen und Feedback. Rechts: Form zur Eingabe und Verwaltung von Differentialdiagnosen. (modifiziert aus [iInTIME 2007])

### 2.6.2 Dogs 'n Drugs

DOGS 'N DRUGS ist ein auf dem Internet basierendes System, das an der Universität Ulm zur Unterstützung der curricularen Ausbildung im Medizinstudium entwickelt wurde [Martens et al. 1999, 2001]. Der Fokus des Systems liegt auf der Diagnostik und adäquaten Anordnung von Therapien. Das System baut auf einer medizinischen Wissensbasis auf, welche unter anderem auf ein Terminology System für Anatomie und Diagnose (nach ICD-10) zurückgreift. Über diese Wissensbasis kann das Erstellen neuer Lernfälle unterstützt und ein automatisches Feedback generiert werden. Zusätzlich liegt dem Konzept von DOGS 'N DRUGS ein Lehrprozessmodell zugrunde, das die medizinischen Fälle in Situationen (Arzt-Patient-Kontakt Szenen, siehe z.B. Abb. 2.19 links) bestehend aus unterschiedlichen Lehrtasks aufteilt. Je nach Zielgruppe des jeweiligen Falles steht eine große Vielfalt an Diagnosemöglichkeiten, wie der körperlichen Untersuchung (siehe Abb. 2.19 rechts), dem Anfordern von Laborwerten und medizinischer Bildgebung zur Verfügung. Über Info-Fenster werden Rückmeldungen über eingeholte Laborwerte und Untersuchungsergebnisse präsentiert.

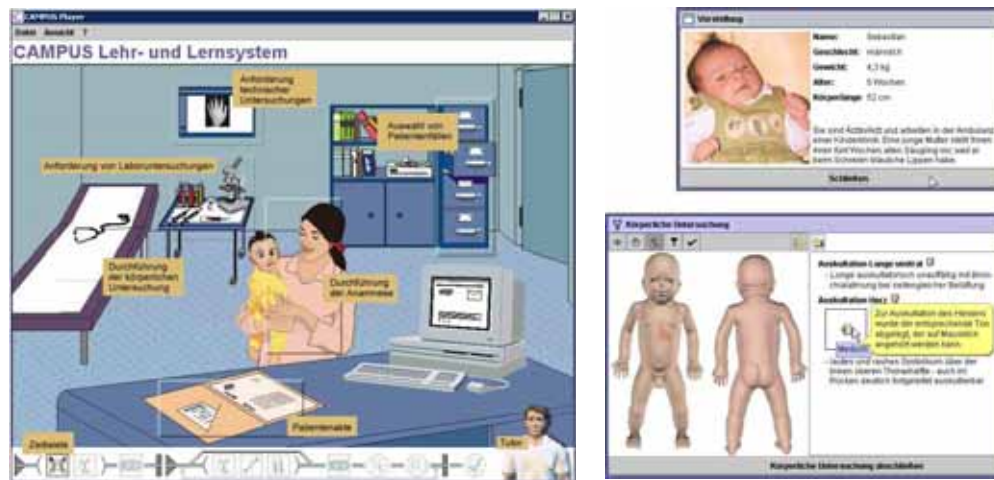


**Abbildung 2.19:** Beispielfall in DOGS 'N DRUGS. Links: initiale Situationsbeschreibung mit Navigation zu den nächsten möglichen Aktionen. Rechts: Virtuelle körperliche Untersuchung des Patienten mit Möglichkeiten z.B. zum Abhören der Herztöne oder Abtasten der Bauchregion. (modifiziert aus [Docs 'n Drugs 2007])

### 2.6.3 CAMPUS

CAMPUS ist ein fallbasiertes Lehr- und Lernsystem, das unter anderem seit über fünf Jahren in der curricularen Ausbildung an der Medizinischen Fakultät der Universität Heidelberg im Fach Pädiatrie eingesetzt wird [Riedel et al. 2000; Garde et al. 2005; Huwendiek et al. 2006]. CAMPUS bietet zwei unterschiedliche Player-Varianten an, mit denen die gleichen Fälle abgearbeitet werden können [Reichert et al. 2007]. Beim *Classic-Player* finden sich die Studierenden in der Rolle eines Ambulanzarztes im grafisch dargestellten Arztzimmer wieder (Abb. 2.20 links). In diesem können die Lernenden frei den aktuellen Fall z.B. durch Aufnehmen der Anamnese, Durchführen einer körperlichen Untersuchung und Anfordern von Laboruntersuchungen bearbeiten (siehe Abb. 2.20 rechts). Diese sogenannte situierte Lernumgebung (auch simulierte (Lern-)Situation genannt) gleicht dabei dem Konzept von DOGS 'N DRUGS. Der zweite Modus *Thin-Player* entspricht dem bildschirm- bzw. seitenba-





**Abbildung 2.20:** Beispielfall in CAMPUS. Links: Arztzimmer als Interaktionsmetapher im CAMPUS *Classic-Player*. Rechts oben: Infofenster für die Patientenvorstellung. Rechts unten: Virtuelle körperliche Untersuchung des Patienten mit Möglichkeiten z.B. zum Abhören der Herztöne oder Abtasten der Bauchregion. (modifiziert aus [MediCase 2007])

sierten Vorgehen von CASUS. Hierbei ist eine feste Reihenfolge bei der Bearbeitung der Fälle vorgegeben und es wird nach und nach von einer Bildschirmseite zur nächsten gegangen.

Der *Classic-Player* bietet neben der selbstgesteuerten Reihenfolge bei der Fallbearbeitung auch realistische Formulare zur Anforderung von Laboruntersuchungen und Bildaufnahmen an. Zur Überprüfung und Eingabe der Lernergebnisse werden realitätsnah sogenannte Long-Menu-Fragen verwendet, bei denen Antworten zum Beispiel aus einer Liste aller ICD-10 klassifizierten Diagnosen entnommen werden müssen, was zu stärkerem Nachdenken über mögliche Antworten im Gegensatz zu Multiple-Choice-Fragen führt [Reichert et al. 2007].

Aktuelle Bestrebungen auf dem Gebiet der allgemeinen CBTs in der Medizin befassen sich mit der Evaluation und verstärkten Integration in die curriculare Ausbildung und mit erweiterten Möglichkeiten beim kooperativen und kollaborativen Lernen im *E-Learning 2.0* [Leven et al. 2006; Roth et al. 2007; Kleimann 2007].

## 2.7 Chirurgiesimulatoren

Systeme zum Erlernen und Trainieren von Operationstechniken und motorischen Fähigkeiten stellen eine weitere Kategorie an CBTs für die chirurgische Ausbildung dar. Chirurgiesimulatoren ermöglichen dabei über physikalische und geometrische Modelle sowie visuelles und haptisches Feedback die realitätsnahe Nachbildung von chirurgischen Eingriffen (siehe z.B. Abb. 2.21 links).

Schwerpunkte und Herausforderungen der aktuellen Forschung auf dem Gebiet der Chirurgiesimulation sind die Berechnung und Simulation von unterschiedlichen anatomischen bzw. pathologischen Varianten mit Weichgewebeverformung, haptischem Feedback, Kollisionserkennung, Gewebereaktionen und Blutungs- oder Raucheffekten sowie die Nachbildung unterschiedlicher chirurgischer Instrumente und Operationstechniken (vergl. [Çakmak und Kühnapfel 2000]). Auf dem Weg zu einem Chirurgiesimulator sind dabei anspruchsvolle und langfristige Entwicklungen auf jedem der einzelnen Gebiete erforderlich. Dies macht die Entwicklung von Chirurgiesimulatoren zu finanziell aufwändigen Mammutprojekten.

Das Trainieren von Operationsplanung ist nicht Gegenstand der Simulatoren. Ein verwandter wichtiger Aspekt im Rahmen der Chirurgesimulation ist jedoch die Erstellung von realistischen 3D-Geometriemodellen für das Training. Hierbei können grundlegend die Modelle erzeugt werden, indem (a) einzelne ideale Anatomiemodelle von Hand bzw. auf Basis des *Visible-Human* erstellt werden oder (b) reale Patientendaten akquiriert und aufbereitet werden. Ergänzend bestehen auch Bestrebungen, (c) parametrische Modelle von anatomischen und pathologischen Strukturen zu entwickeln, auf deren Basis dann unterschiedliche Fälle generiert werden können.

### 2.7.1 Karlsruhe Endoscopic Surgery Trainer

Der KARLSRUHE ENDOSCOPIC SURGERY TRAINER basiert auf der KISMET (Kinematic Simulation, Monitoring and Off-Line Programming Environment for Telerobotics) Software und wurde am Forschungszentrum Karlsruhe entwickelt. Auf dem System existieren verschiedene Anwendungen zur Simulation von endoskopischen minimal-invasiven Eingriffen (z.B. für die Gynäkologie siehe Abb. 2.21 links).

Realistische chirurgische Modelle werden beim KARLSRUHE ENDOSCOPIC SURGERY TRAINER von Hand mit dem KISMO (*Kismet Modeller*) [Çakmak und Kühnapfel 2000] modelliert (Abb. 2.21 rechts). Hierbei können ausgehend von drei Grundformen (Ebene, Zylinder, Kugel) Feder-Masse-Modelle beliebiger Anatomie erzeugt werden. Direkte Volumenvisualisierung, z.B. des *Visible-Female* Datensatzes, kann dabei als Modellier- und Texturierungsvorlage eingeblendet und verwendet werden.

### 2.7.2 Leberchirurgiesimulation

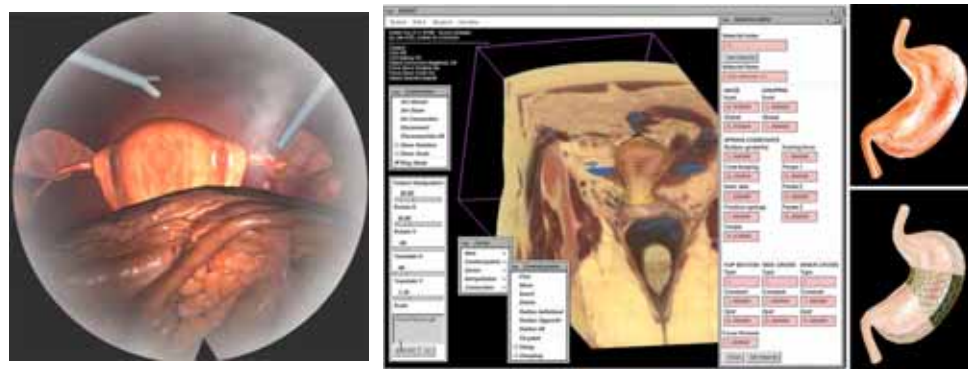
Für die Simulation von laparoskopischen Leberoperationen [Delingette und Ayache 2005] (Abb. 2.22) werden am französischen Institut *INRIA* Leberoberfläche, Gefäße und Tumore aus CT-Daten segmentiert und durch *Simplex Meshes* approximiert. Die *Simplex Meshes* [Delingette 1999] erlauben dabei die interaktive Verformung der Objekte. Die Approximation der segmentierten Strukturen erfolgt durch manuelles Modellieren und semi-automatische Anpassungsprozesse. Im Mittelpunkt der Anpassungen steht dabei die Erhöhung der Qualität der Dreiecksflächen und die Reduktion ihrer Anzahl in den generierten Oberflächenmodellen.

### 2.7.3 Hysteroscopy Simulator

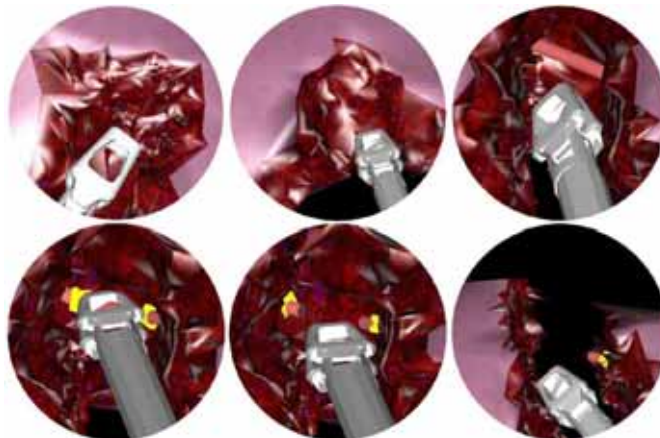
Der an der ETH Zürich entwickelte HYSTEROSCOPY SIMULATOR [ETH Zürich 2007] ermöglicht das Trainieren von endoskopischen Uterusuntersuchungen (Abb. 2.23 links). Zur Erzeugung variabler Trainingsszenarien können im HYSTEROSCOPY SIMULATOR über parametrisierte Modelle Organe mit variabler Geometrie erstellt werden.

Über *Active Shape Models* (ASMs) wird die anatomische Variabilität gesunder Organe repräsentiert [Sierra et al. 2004]. Die ASMs werden hierzu aus einer Reihe segmentierter Patientendaten gewonnen. Durch Anpassen der Ausprägung verschiedener typischer Variationen können dann beliebige Variationen der Organe künstlich erzeugt werden.

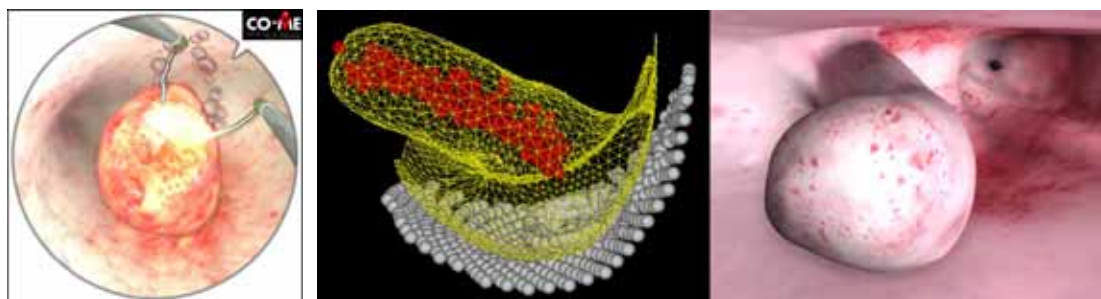
Pathologische Veränderungen (Polypen und Myome) im Uterus werden über parametrisierte Skelette oder unter Einbindung einer gegebenen Umgebungsanatomie durch simulierte Wachstumsprozesse (siehe Abb. 2.23) künstlich generiert [Sierra et al. 2003a].



**Abbildung 2.21:** Screenshots vom KISMET-System. Links: Simulation einer gynäkologischen laparoskopischen Operation mit Raucheffekten und Linsentrübung. Rechts: KISMO-Modellierungswerkzeug zum manuellen Erstellen von 3D-Modellen für den Chirurgesimulator. [Çakmak und Kühnapfel 2000]



**Abbildung 2.22:** Bildsequenz einer simulierten laparoskopischen Leberresektion mit Setzen von Clips und Durchtrennen der Pfortader im Leberchirurgiesimulator des INRIA. [Delingette und Ayache 2005]



**Abbildung 2.23:** Hysteroscopy Simulator: Links: Simulierte Entfernung eines Myoms im Uterus (modifiziert aus [ETH Zürich 2007]). Rechts: Partikelbasiertes Wachstumsmodell und gerenderte Oberfläche eines Polypen [Sierra et al. 2003b].

## 2.8 Zusammenfassung

Die chirurgische Ausbildung bzw. das chirurgische Wissen basiert grundlegend auf den drei Bereichen *theoretische Chirurgie*, *klinische Chirurgie* und *Operationstechniken* (vergl. Abschnitt 1.3). 3D-Anatomielernprogramme und die meisten allgemeinen CBTs in der Medizin sind ausschließlich für den Erwerb von Fakten- und theoretischem Chirurgiewissen konzipiert. Chirurgesimulatoren hingegen werden für das gefahrlose Trainieren von motorischen Fähigkeiten und Operationstechniken entwickelt. CBT-Systeme für die klinische Chirurgie sind jedoch rar. Während mit CAMPUS (siehe Abschnitt 2.6.3) CBT-Anwendungen für die klinische Chirurgie in Bezug auf pathologische Variationen, Differentialdiagnostik und Therapiealternativen möglich sind, ist das Erlernen und Trainieren der computerunterstützten Interventionsplanung nicht möglich.

Anwendung findet die Computerunterstützung bei den genannten Beispielen aus der Interventionsplanung mit dem Ziel, möglichst viel gesundes Gewebe zu schonen sowie Funktion (z.B. Beweglichkeit von Gelenken, Erhalt von ausreichend gesundem Lebergewebe) und Ästhetik (z.B. Symmetrie des Gesichts) wieder herzustellen bzw. bestmöglich garantieren zu können. Systeme zur computerunterstützten Interventionsplanung bauen dabei auf Ergebnissen der Bildanalyse (segmentierte Strukturen, quantifizierte Abstände, Ausdehnungen, Dichtekonzentrationen, ...) auf. Die Bildanalyseergebnisse werden verwendet, um unter anderem die patientenindividuelle Anatomie angemessen darstellen zu können, automatisch Therapievorschlüsse unterbreiten zu können, oder um zu erwartende Therapieergebnisse abzuschätzen bzw. zu simulieren.

Überblickend finden sich in allen genannten Anwendungen die genannten Ankerpunkte für die Computerunterstützung bei der Interventionsplanung (siehe Kap. 1): 1. Segmentierung und Quantifizierung, 2. Berechnung bzw. Simulation von Planungsergebnissen, 3. Visualisierung, und 4. Interaktionsmöglichkeiten. Eine Ausnahme bilden 5. die Ausbildung und das Training von computerunterstützten Therapieentscheidungs- und Interventionsplanungsprozessen, welche bisher (durch die sich selbst noch überwiegend im Entwicklungsstatus befindlichen Interventionsplanungssysteme) weniger Beachtung fanden.

### 2.8.1 Herausforderungen der computerunterstützten Interventionsplanung

Aus den Ankerpunkten für die computerunterstützte Interventionsplanung, ihrem Nutzen, wenn komplexe dreidimensionale Lagebeziehungen berücksichtigt, bewertet und/oder manipuliert werden müssen, und anhand der betrachteten Beispielanwendungen können Herausforderungen für die computerunterstützte Interventionsplanung abgeleitet werden. Dies sind:<sup>1</sup>

- *Identifikation und Abgrenzung von relevanten Strukturen (Segmentierung)*
- *Integration verschiedener Bildmodalitäten*
- *Rekonstruktion von patientenindividuellen anatomischen Oberflächenmodellen (für Visualisierung und Simulationen)*
- *Berechnung, Simulation und Bewertung zu erwartender Therapieergebnisse*

---

<sup>1</sup>Kursiv gedruckte Punkte gehen über den Rahmen der Themenstellung der vorliegenden Arbeit hinaus und werden bzw. sind von anderen Arbeiten zu behandeln.

- Ausbildung und Training neuer Interventionsplanungsmöglichkeiten
- *Übertragung von Planungsergebnissen auf die intraoperative Situation*
- *Belegen des therapeutischen Nutzens der Planung*

Spezielle Anforderungen auf der Ebene der Visualisierung und Mensch-Computer-Interaktion werden dabei gestellt an:

- *Techniken zur direkten oder indirekten Visualisierung der komplexen dreidimensionalen Patienten-anatomie*
- Techniken zur Erzeugung von patientenindividuellen anatomischen Oberflächenmodellen aus segmentierten Volumendaten
- Techniken zur effizienten Exploration und Navigation im 3D-Raum (Rotation)
- Techniken zur standardisierten Präsentation und dynamischen Visualisierung (standardisierte Animationen)

Die vorliegende Arbeit vertieft im Rahmen ihrer Zielstellung die Betrachtung von Techniken (1) zur Erzeugung patientenindividueller anatomischer Oberflächenmodelle (Kap. 3–7), (2) zur effizienten Exploration und Navigation im 3D-Raum (Kap. 8) und (3) zur standardisierten Präsentation und dynamischen Visualisierung (Kap. 9).

## 2.8.2 Herausforderungen der computerunterstützten Ausbildung

Relevante Herausforderungen bei der computerunterstützten Interventionsplanung (Abschnitt 2.8.1) bestehen auch für die betrachteten CBTs in der Medizin.

**Erzeugung anatomischer Oberflächenmodelle:** Sofern die CBTs dreidimensionale Anatomie verwenden (3D-Anatomiernprogramme, Chirurgiesimulatoren) müssen diese zuvor erzeugt werden. Während bei der Interventionsplanung genaue Anatomiemodelle des aktuellen Patienten erforderlich sind, wird für Ausbildung und Training meist auf „ideale“ Modelle zurückgegriffen. Für die Präsentation variabler Trainingsszenarien ist dieses Vorgehen ungeeignet. Entsprechend müssen 3D-Modelle parametrisch erzeugt werden, bzw. aus klinischen Daten (ähnlich der Interventionsplanung) extrahiert werden.

**Dynamische Präsentation (Animation):** Ein Vorteil von CBTs gegenüber konventionellen Lernmitteln stellen die vielfältigen Möglichkeiten zur Präsentation von multimedialen Inhalten (Ton, Videos, Animationen) dar. In 3D-Anatomiernprogrammen sowie allgemeinen CBT-Systemen in der Medizin kommen hierzu vorrangig vorberechnete Animationen und/oder Animationen bekannter 3D-Geometrie zum Einsatz. Zur Erstellung entsprechender Animationen für variable bzw. patientenspezifische Anatomiemodelle müssen Möglichkeiten zur effizienten Generierung standardisierter dynamischer Visualisierungen entsprechend variabler Daten gefunden werden.

**Effiziente 3D-Exploration und Navigation:** Die Präsentation von interaktiven 3D-Modellen erfordert auch in der Ausbildung effiziente Techniken zur Exploration und Navigation im virtuellen Raum. Zur Unterstützung des selbstgesteuerten tieferen Auseinandersetzens mit der 3D-Szene (siehe z.B. 3D PUZZLE Abschnitt 2.5) ist eine Beschränkung auf vorgeordnete Sichtrichtungen, wie z.B. im VOXELMAN und ANATOMYBROWSER (Abschnitt 2.5) nicht wünschenswert.

**Konzeptionelle Herausforderungen:** Für das computerunterstützte Lernen und Trainieren der Operationsplanung ergeben sich neben den technischen Herausforderungen für die Interventionsplanung auch folgende konzeptionelle Herausforderungen (vergl. [Lee und Owens 2000; Herczeg 2006]):

- Analyse und Spezifikation von Lernzielen und Inhalten
- Zielgruppenanalyse
- Lerndidaktische Konzeption und Organisation (Lernszenarios, Lernwege, Lernstufen, ...)
- Aufbau einer ausreichenden Fallbasis (mit charakteristischen Fällen für unterschiedliche Lerninhalte und Lernstufen)
- Interaktionsdesign: Entwicklung von Programmstruktur und Interaktionskonzept z.B. für die Navigation auf dem Lernweg und die Auswahl von Lernfällen
- Konzeption von Hilfe und Feedback
- Konzeption von Lernerfolgskontrolle und Evaluation

Auf die genannten Aspekte wird im Rahmen der Konzeption und Umsetzung des LIVER SURGERY TRAINER für Ausbildung und Training von Interventionsplanungsmöglichkeiten in der Leberchirurgie (Kap. 10) näher eingegangen.

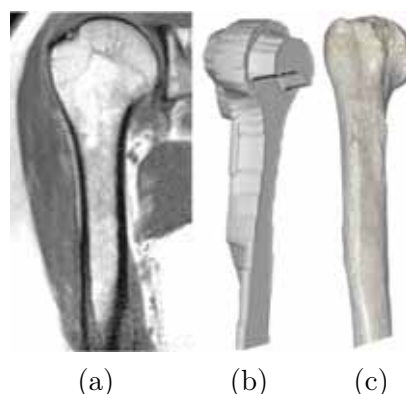
# Artefaktreduktion in medizinischen Ober- flächenmodellen: *Grundlagen*

---

Um Oberflächenmodelle von Strukturen aus radiologischen Bilddaten extrahieren zu können, müssen diese Strukturen eindeutig identifiziert bzw. abgegrenzt werden. Im einfachsten Fall kann dies anhand eines für eine Struktur spezifischen Iso-Wertes (z.B. bei Knochen  $>1000$  HU im CT) erfolgen. Meist ist dies jedoch nicht möglich und zu extrahierende Strukturen müssen zuvor mit speziellen Verfahren segmentiert werden. Als Ergebnis liefert dies im Allgemeinen eine binäre Segmentierungsmaske, die jedem Voxel des Bilddatensatzes ein Label (z.B. 0 = Hintergrund, 1 = Objekt) zuordnet. Anhand einer Segmentierungsmaske können dann Oberflächenmodelle extrahiert werden, die die Grenzfläche der segmentierten Strukturen repräsentieren.

Aus binären Segmentierungsmasken extrahierte Oberflächenmodelle weisen jedoch aufgrund der diskreten Auflösung der Bilddaten und der binären Zuordnung der Voxel Block- bzw. Terrassenartefakte auf. Abbildung 3.1 stellt ein Schichtbild aus einem MRT-Datensatz eines Oberarmknochens (a) mit einem daraus generierten Oberflächenmodell (b) und der Realaufnahme eines entsprechenden Knochens (c) beispielhaft gegenüber. Wie im Vergleich von Abb. 3.1 (b) und (c) demonstriert, kann ein generiertes Oberflächenmodell durch Block- und Terrassenartefakte durchaus stark vom realen Objekt abweichen.

Die Reduktion der Artefakte in Oberflächen aus binären Segmentierungsmasken ist Ziel dieses Teils der Arbeit. Im Folgenden wird einleitend dargelegt, wann und warum binäre Segmentierungsmasken erforderlich sind und welche Probleme sich bei der Oberflächenextraktion ergeben. Danach wird die Zielsetzung und Herangehensweise erläutert sowie Grundlagen zur verwendeten Notation, Krümmungsberechnung und zu Datenstrukturen dargelegt.



**Abbildung 3.1:** Gegenüberstellung von (a) einem MRT-Schichtbild eines Oberarmknochens, (b) mittels *Marching Cubes* extrahiertem Oberflächenmodell und (c) einer Realaufnahme. (Realaufnahme: [Wikimedia Foundation Inc. 2005])



### 3.1 Warum binäre Segmentierung?

In einfachen Fällen der Segmentierung können unterschiedliche Regionen (Strukturen) in den Bilddaten durch geeignete Schwell- bzw. Iso-Werte voneinander abgegrenzt werden (siehe Abb. 3.2 (a)). Können die Regionen, wie in Abbildung 3.2 (a), z.B. mit einem Iso-Wert von 70, eindeutig voneinander abgegrenzt werden, so würde das Speichern dieser Information zusammen mit den Bilddaten zur späteren Extraktion glatter Oberflächen ausreichen.

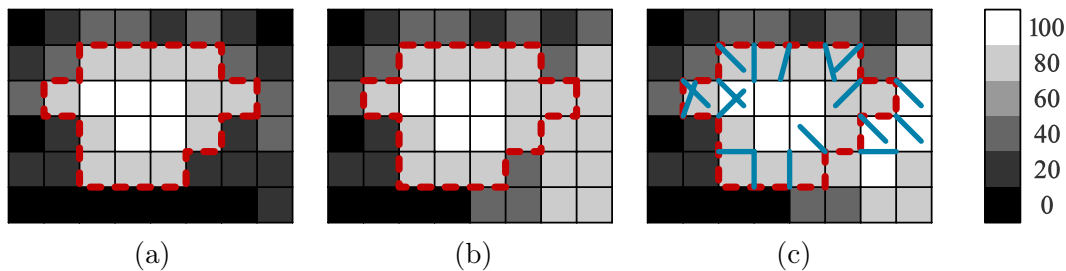
In Abbildung 3.2 (b) und (c) muss das Segmentierungsverfahren auch Voxel mit gleichen Werten und unterschiedlicher Gradientenrichtung abtrennen. Die Speicherung der Segmentierungsergebnisse über eine eindeutige Klassifizierung in Objekt- und Hintergrundvoxel stellt hier die Methode der Wahl dar.

In erweiterten Varianten können auch mehrere segmentierte Objekte gleichzeitig in einer *multi-label Maske* [Reitinger et al. 2005] gespeichert werden bzw. jedes Voxel gleichzeitig mehreren Strukturen angehören. Relevant ist dies zum Beispiel bei Strukturen innerhalb von Organen, da entsprechende Voxel dann zu dem Organ (z.B. der Leber), aber auch gleichzeitig zu inneren Strukturen (wie z.B. der Lebervene) gehören. Diese Mehrfachzuordnung der Voxel ändert jedoch nichts am binären Charakter der Segmentierung (Voxel gehört oder gehört nicht zum Objekt).

Ein Ansatz, um eine binäre Klassifikation durch die Segmentierung zu vermeiden, stellt die sogenannte *Soft Segmentation* von Prewer und Kitchen [2001] dar. Das Verfahren beruht auf *Weighted Linked Pyramids*, die zu den Segmentierungsverfahren mit regulären Bild-Pyramiden zählen [Marfil et al. 2006]. Da diese Verfahren unter anderem nicht geeignet sind elongierte Objekte zu segmentieren, kann die *Soft Segmentation* von Prewer und Kitchen [2001] nicht generell verwendet werden [Bister et al. 1990; Marfil et al. 2006].

Ein genereller Ansatz wird in [Hege et al. 1997; Stalling et al. 1998] verwendet. Hierbei werden zusätzlich zur binären Zuordnung eines Voxels zu einer Struktur Informationen über die Wahrscheinlichkeit der Zugehörigkeit abgespeichert. Die Bestimmung der Wahrscheinlichkeitswerte wird dabei aus einem einfachen Regionenwachstumsverfahren abgeleitet. Ist ein anderes Segmentierungsverfahren erforderlich, müssen die Werte interaktiv bestimmt oder über die Glättung der binären Segmentierungsmaske ermittelt werden. Entsprechende Glättungsverfahren werden in Abschnitt 4.1 separat betrachtet und ihre Möglichkeiten und Grenzen aufgezeigt.

Zusammenfassend ist die Speicherung der Segmentierungsergebnisse in binären Masken das Standardverfahren in der medizinischen Bildverarbeitung und Visualisierung. Die Genera-



**Abbildung 3.2:** Segmentierung in Bilddaten: (a) Regionen können mittels Schwellwert bzw. Iso-Wert (iso = 70) abgegrenzt werden, (b) Segmentierung muss Regionen gleicher Intensität trennen, (c) Segmentierung muss Regionen trennen, bei denen auch die Gradientenrichtung an der Regionengrenze wechselt.



lität und Speichereffizienz mögen Gründe hierfür sein. Probleme, die bei der Extraktion von Oberflächen aus entsprechenden binären Segmentierungsmasken entstehen, werden im folgenden Abschnitt kurz illustriert.

## 3.2 Probleme bei der Oberflächenextraktion

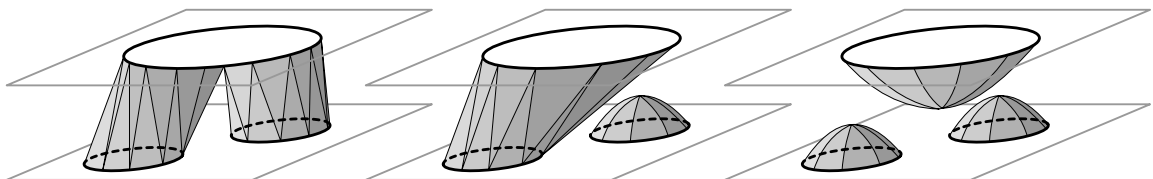
Die Extraktion von Oberflächenmodellen (boundary representations b-reps) aus schichtbasierten radiologischen Aufnahmen kann grundlegend für jede Schicht auf ein zweidimensionales Problem zurückgeführt werden. Hierbei sind in jeder Schicht die Konturen zwischen der zu extrahierenden Struktur zu finden. Anschließend sind die Konturen aus jeder Schicht passend mit den benachbarten Schichten zu verbinden, so dass eine dreidimensionale Oberfläche entsteht.

### 3.2.1 Konturkorrespondenzproblem

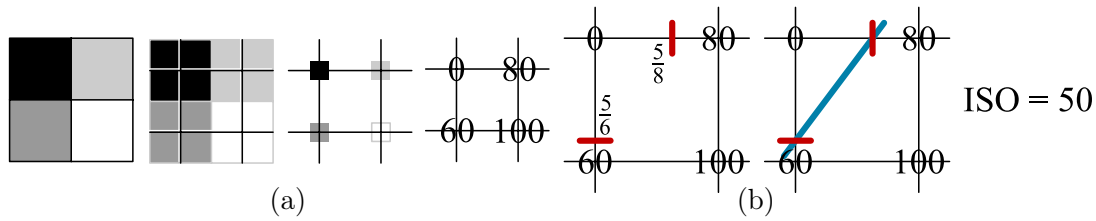
Die Herausforderung beim Verbinden der Konturen benachbarter Schichten stellt das Lösen des Korrespondenzproblems zwischen den Eckpunkten (vertices) der Konturen dar. Hier muss entschieden werden, welche Eckpunkte der Kontur auf einer Schicht mit welchen Eckpunkten einer Kontur auf einer anderen Schicht zu verbinden sind. Problematisch ist dies vor allem dadurch, dass sich die Konturen eines Objektes typischerweise von einer zur nächsten Schicht ändern. Zusätzlich können topologische Veränderungen (z.B.: Verzweigung) der Kontur von Schicht zu Schicht auftreten. Entsprechend muss, wie in Abbildung 3.3, eine Korrespondenz zwischen Eckpunkten mehrerer Konturen gefunden werden.

Ist der Abstand zwischen den Schichten so groß, dass sich die Form der Objekte zwischen den Schichten stark verändert, kann es vorkommen, dass sich die gefundenen Konturen in benachbarten Schichten nicht mehr überlappen. Dies kann dann bei der Oberflächenerzeugung zu abgelösten Elementen eines Objektes führen.

Zellbasierte Verfahren, wie beim Marching Cubes (MC) Algorithmus von Lorensen und Cline [1987] und allen abgeleiteten Varianten, vereinfachen das Korrespondenzproblem. Hierzu wird der 3D-Bildraum in kleine Zellen (Würfel bei MC) aufgeteilt, in denen dann separat das Problem gelöst und eine Oberfläche erzeugt wird. Abbildung 3.4 illustriert am Beispiel des MC-Algorithmus dieses Vorgehen im zweidimensionalen Raum bei der Extraktion einer Oberfläche anhand eines gegebenen Iso-Wertes. Zur einfachen Erzeugung der Oberfläche wird in den betrachteten Zellen durch Interpolation ermittelt, wo auf den Kanten der Zelle der Iso-Wert auftritt. An dieser interpolierten Position wird dann ein Eckpunkt für die Oberfläche erzeugt. Ein entsprechendes Vorgehen garantiert, dass überlappende Konturen in benachbarten Schichten eine plausible zusammenhängende Oberfläche ergeben.



**Abbildung 3.3:** Ausgewählte mögliche Lösungen des Korrespondenzproblems bei Topologieänderung (Verzweigung) der Konturen in benachbarten Schichten.

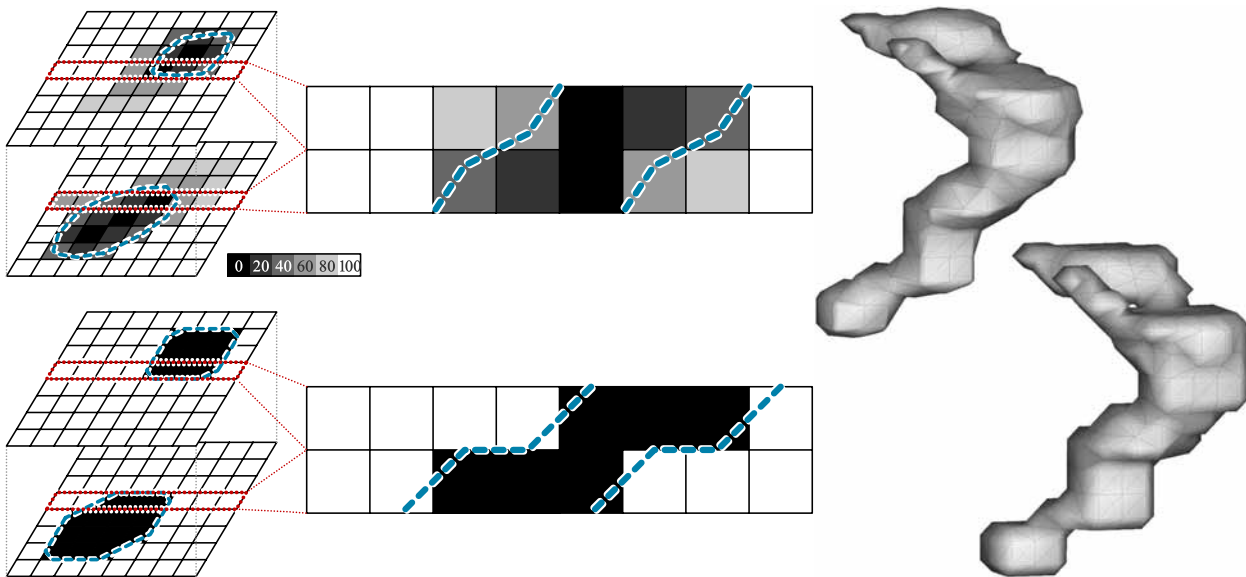


**Abbildung 3.4:** Marching Cubes Algorithmus in 2D: (a) vier benachbarte Voxel werden als eine Zelle aufgefasst; (b) Interpolation des Schnittpunktes der Iso-Fläche mit den Zellgrenzen und Verbindung bzw. Triangulierung der ermittelten Schnittpunkte über eine Lookup-Tabelle (nicht mit dargestellt).

### 3.2.2 Oberflächenextraktion aus binären Bilddaten

Bei binären Bilddaten enthalten die gegebenen Volumendaten jeweils nur einen Wert für alle Objektvoxel und einen anderen Wert für alle Hintergrundvoxel. Der Iso-Wert  $i_{iso}$  für die Oberflächenextraktion ergibt sich dann aus  $i_{iso} = (i_1 + i_2)/2$ , wobei  $i_1$  den Intensitätswert der Objektvoxel und  $i_2$  den der Hintergrundvoxel repräsentieren. Als Beispiel sei  $i_1 = 100$  und  $i_2 = 0$ , wobei sich dann  $i_{iso} = 50$  ergibt.

Entsprechend dieser Gegebenheiten verläuft die zu extrahierende Oberfläche immer genau in der Mitte zwischen einem Objektvoxel und einem Hintergrundvoxel. Bei der Extraktion der Oberfläche kann daher über die Interpolation des Iso-Wertverlaufs in einer Zelle im Vergleich zu nicht-binären Daten keine zusätzliche Information über den Verlauf der Objektgrenze gewonnen werden. In extrahierten Oberflächen äußert sich dies, indem Stufen zwischen einzelnen Ebenen sichtbar werden können, da weiche Übergänge fehlen. Abbildung 3.5 stellt vergleichend die Oberflächenextraktion aus einem nicht-binären Datensatz und dessen binärem Gegenstück dar.



**Abbildung 3.5:** Oberflächenextraktion aus einem nicht-binären (oben) und einem binarisierten Datensatz (unten). Links: Jeweils zwei Schichten aus den zwei Datensätzen und 2D-Darstellung der Oberflächengrenze. Rechts: Extrahierte Oberflächen.

### 3.3 Zielsetzung und Herangehensweise

Ein aus radiologischen Bilddaten extrahiertes Oberflächenmodell kann immer nur annähernd das Original bzw. die reale Anatomie des Patienten darstellen. Artefakte bzw. Diskontinuitäten, wie sie durch die Extraktion aus binären Segmentierungsmasken entstehen, können aber die Objektwahrnehmung übermäßig behindern.

Das visuelle System des Menschen ist sehr empfindlich gegenüber Diskontinuitäten in Oberflächen, da diese normalerweise wichtige Merkmale bei der Objekterkennung und Objektwahrnehmung darstellen. Zusätzlich werden bei merkmalsbetonenden bzw. illustrativen Darstellungstechniken (Silhouetten, Schraffur, ...) z.B. zur Darstellung von Kontextinformationen in medizinischen Visualisierungen diese Artefakte unvermeidlich mit betont. Entsprechend binden solche Artefakte unnötig die visuelle Aufmerksamkeit und entfremden oder verbergen gar die Form der dargestellten Strukturen. Gerade im Umfeld der chirurgischen Ausbildung und Interventionsplanung sollten die dargestellten Objekte aber einfach, effizient und genau von den Betrachtern erkannt werden können.

Bei der Simulation von physikalischen oder physiologischen Prozessen (z.B. bei der Strömungssimulation in Blutgefäßen) können darüber hinaus entsprechende Diskontinuitäten die Simulation verfälschen oder gar unmöglich machen.

Ziel dieses Teils der Arbeit ist es deshalb, Möglichkeiten zur Reduktion von Artefakten in Oberflächenmodellen aus binären Volumendaten zu untersuchen und geeignete Methoden zur effizienten Erzeugung von hochwertigen (artefaktreduzierten) Oberflächenmodellen zu entwickeln.

Dazu werden in den folgenden Abschnitten Grundlagen zur Verarbeitung von Oberflächenmodellen (Polygonnetzen) vertieft. Daraufhin wird in Kapitel 4 die Oberflächenextraktionspipeline analysiert. Dabei werden die auf den Stufen der Pipeline existierenden Strategien zur Reduktion der Artefakte identifiziert und es wird der Stand der Technik dargelegt.

Der Fokus wird dann auf die Glättung extrahierter Polygonnetze gelegt, da diese zur Reduktion der Artefakte in der medizinischen Visualisierung weit verbreitet sind und ihre Eignung weitgehend unklar ist. Ziel ist hierbei die Erarbeitung von Vorschlägen und Richtlinien für die Glättung von medizinischen Oberflächenmodellen. In Kapitel 5 werden dazu typische Oberflächenmodelle der medizinischen Visualisierung und ihre Artefakte untersucht sowie Anforderungen an ihre Glättung erarbeitet. Darauf aufbauend wird ein Vergleich von Glättungsalgorithmen konzipiert und durchgeführt, der dann die Ableitung von Vorschlägen und Richtlinien ermöglicht.

Die gewonnenen Kenntnisse aus dem Vergleich der Glättungsverfahren werden dann in Kapitel 6 und 7 genutzt, um Lösungen für identifizierte offene Probleme zu entwickeln.

### 3.4 Verwendete Notation

Ein Oberflächennetz  $\mathcal{M} = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$  besteht aus einer Menge von Eckpunkten (*vertices*)  $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ , Kanten (*edges*)  $\mathcal{E}$  und Flächen (*faces*)  $\mathcal{F}$ . Dabei stellen  $v_1, v_2, \dots, v_n$  die  $n$  Eckpunkte des Oberflächennetzes dar. Zwei Eckpunkte  $(v_i, v_j)$  können jeweils eine Kante  $\vec{e}_{ij} = (v_j - v_i) \in \mathcal{E}$  bilden, wobei  $v_i, v_j \in \mathcal{V} \wedge v_i \neq v_j$  gilt.

#### 3.4.1 Nachbarschaften

Die Menge aller topologischen Nachbarn  $\mathcal{U} \subseteq \mathcal{V}$  eines Eckpunktes  $v$  wird als  $\mathcal{U}_v^g$  bezeichnet, wobei  $g$  den Grad der topologischen Nachbarschaft angibt. Alle Nachbarpunkte werden dabei als  $(u_1, \dots, u_{j-1}, u_j, u_{j+1}, \dots, u_m) \in \mathcal{U}_{v_i}^g$  repräsentiert. Hierbei gilt für alle  $u_j \in \mathcal{U}_{v_i}^g$ , dass  $u_j$  von  $v_i$  über  $g$  oder weniger Kanten zu erreichen ist. Jedes  $j$  mit  $1 \leq j \leq m$  gibt dabei einen Index für alle Nachbarpunkte an, wobei  $m = \|\mathcal{U}_{v_i}^g\|$  die Anzahl aller Punkte der  $g$ -ten Nachbarschaft von  $v_i$  repräsentiert. Abbildung 3.6 illustriert die Nachbarschaft ersten ( $\mathcal{U}_{v_i}^1$ ) und zweiten Grades ( $\mathcal{U}_{v_i}^2$ ).



**Abbildung 3.6:** Topologische Nachbarschaft in Polygonnetzen.

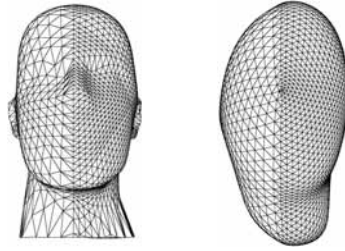
Neben der topologischen Nachbarschaft können bei der Verarbeitung von Polygonnetzen auch andere Nachbarschaftsmetriken herangezogen werden. Üblich ist dabei die Verwendung der euklidischen oder geodätischen Distanz der Eckpunkte. Hierbei ist ein Eckpunkt  $u_j$  dann ein Nachbar von  $v_i$ , sofern  $\mathcal{D}(v_i, u_j) < d$ , wobei  $\mathcal{D}(v_i, u_j)$  den Abstand zweier Eckpunkte über das verwendete Distanzmaß und  $d$  den Radius der Nachbarschaft angibt.

Bei der Glättung von Polygonnetzen finden hauptsächlich topologische Nachbarschaftsbeziehungen Anwendung. Dies hat den Vorteil, dass diese über alle Iterationen hinweg konstant bleiben (Glättung verändert die Topologie nicht), während abstandsasierte Nachbarschaften in jeder Iteration neu berechnet werden müssten. Nachteilig kann sich die Verwendung der topologischen Nachbarn jedoch auswirken, wenn diese nicht annähernd gleichmäßig und mit gleichem Abstand verteilt sind (siehe Abb. 3.7). Unter Verwendung von unterschiedlichen Gewichtungen der Nachbarn im Glättungsprozess kann dieser Nachteil jedoch begrenzt werden, so dass sich grundlegend auf die topologische Nachbarschaft ( $\mathcal{U}_v^g$ ) beschränkt werden kann.

#### 3.4.2 Explizite vs. Implizite Integration

Die Glättung eines Oberflächennetzes  $\mathcal{M}$  z.B. mit einem Low-Pass-Filter kann über die Verwendung partieller Differentialgleichungen wie folgt beschrieben werden:

$$\frac{\delta \mathcal{M}}{\delta t} = \nabla \mathcal{M} \quad (3.1)$$



**Abbildung 3.7:** Verzerrungen bei ungleichmäßiger topologischer Nachbarschaft. (modifiziert aus: [Desbrun et al. 1999])

wobei  $\nabla$  den Filter repräsentiert. Analytische Methoden zur Lösung von Gleichung 3.1 im 3D-Raum sind sehr kompliziert. Deshalb erfolgt die Lösung meist über den Finite-Differenzen-Ansatz unter Verwendung der folgenden iterativen Gleichung:

$$\mathcal{M}_{\text{explizit}}^{t+1} = \mathcal{M}^t + \lambda \nabla \mathcal{M}^t \quad (3.2)$$

wobei  $\lambda$  einen Zeitschritt bzw. allgemeiner einen Wichtungsfaktor pro Iteration darstellt. Gleichung 3.2 repräsentiert damit die grundlegende Form der expliziten Filterung/Glättung eines Oberflächennetzes.

Da die explizite Form auf Zeitschritte  $\lambda \leq 1$  beschränkt ist, existieren zur schnelleren Konvergenz des Verfahrens Ansätze zur impliziten Integration [Desbrun et al. 1999]. In der verwendeten Notation [Chen et al. 2004] ergibt sich die implizite Integration nach [Desbrun et al. 1999] wie folgt:

$$(I - \lambda dt L) \mathcal{M}^{t+1} = \mathcal{M}^t \quad \Rightarrow \quad \mathcal{M}_{\text{implizit}}^{t+1} = \mathcal{M}^t + \lambda \nabla \mathcal{M}^{t+1} \quad (3.3)$$

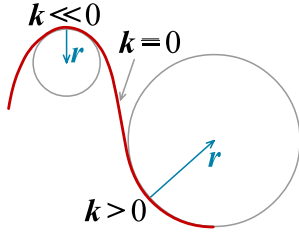
wobei Gleichung 3.3 dann aufwändiger über präkonditionierte Bi-Konjugierte-Gradienten-Verfahren (PBCG) (lineares Gleichungssystem gegenüber einer expliziten linearen Gleichung) zu lösen ist [Desbrun et al. 1999]. Folglich reichen weniger Iterationen aus, um eine signifikante Glättung zu erreichen, jedoch wird die Berechnung pro Iteration zeitintensiver, so dass Laufzeitvorteile erst im Vergleich zu mehreren hundert Iterationen bei expliziter Lösung ins Gewicht fallen.

Für eine bessere Übersichtlichkeit und Vergleichbarkeit beschränkt sich deshalb die vorliegende Arbeit auf die detaillierte Betrachtung der expliziten Integrationslösung. Bei zitierten Publikationen, die sich auf eine implizite Integration berufen, wird dies in dieser Arbeit aber angemerkt.

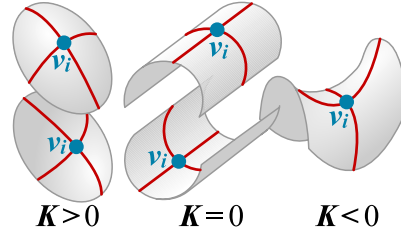
## 3.5 Bestimmung von Krümmungsmaßen

Grundlegend ist die Krümmung für analytisch beschreibbare Oberflächen definiert und beschreibt dabei die Abweichung der unmittelbaren Umgebung eines Oberflächenpunktes  $v$  von einer Ebene. Der Krümmungsbetrag ist dabei reziprok zum Krümmungsradius (siehe Abb. 3.8 im  $\mathbb{R}^1$ ) definiert. Auf Oberflächen im  $\mathbb{R}^3$  ist der Krümmungsradius richtungsabhängig und kann für jede beliebige Richtung auf der Tangentialebene in  $P$  angegeben werden. Zur Angabe der Krümmung sind dabei unterschiedliche Krümmungsmaße gebräuchlich. Wichtigste Krümmungsmaße sind:

$k_1, k_2$ : orthogonale *Hauptkrümmungen*, wobei  $k_1$  meist als *Maximale Krümmung* mit  $|k_1| \geq |k_2|$  definiert wird.



**Abbildung 3.8:** Krümmungskreise und Betrag der Krümmung  $k$  im  $\mathbb{R}^1$  mit  $k > 0$  in konvexen und  $k < 0$  in konkaven Bereichen.



**Abbildung 3.9:** Betrag der *Gaußschen Krümmung* zur Klassifikation von elliptischen  $K > 0$ , parabolischen  $K = 0$  und hyperbolischen  $K < 0$  Oberflächenregionen.

$H$ : *Mittlere Krümmung* (Mean Curvature), mit  $H = \frac{k_1 + k_2}{2}$

$K$ : *Gaußsche Krümmung* (Gaussian Curvature), mit  $K = k_1 k_2$

Bei analytisch beschreibbaren Oberflächen stellen die Hauptkrümmungen  $k_1$  und  $k_2$  die Eigenwerte der Hesse-Matrix aus den partiellen zweiten Ableitungen dar. Die zu den Eigenwerten  $k_1$  und  $k_2$  gehörenden Eigenvektoren  $\vec{k}_1$  und  $\vec{k}_2$  können dabei ermittelt werden, um die Hauptkrümmungsrichtungen zu bestimmen. Bei einer parametrisch gegebenen quadratischen Oberfläche der Form

$$z = ax^2 + bxy + cy^2 + dx + ey + f$$

im  $\mathbb{R}^3$  ergeben sich daraus  $k_1$  und  $k_2$  wie folgt:

$$k_{1,2} = a + c \pm \sqrt{(a - c)^2 + b^2} \quad (3.4)$$

Sind *Mittlere Krümmung*  $H$  und *Gaußsche Krümmung*  $K$  gegeben, so können  $k_1$  und  $k_2$  auch über das Lösen der folgenden Gleichung bestimmt werden:

$$k^2 - 2Hk + K = 0 \quad \Rightarrow \quad k_{1,2} = H \pm \sqrt{H^2 - K} \quad (3.5)$$

Das Vorzeichen der Hauptkrümmungen  $k_1$  und  $k_2$  gibt Aufschluss über ausgewölbte (konvexe) bzw. eingewölbte (konkave) Bereiche der Oberfläche. In konvexen Regionen ist das Vorzeichen positiv, während es in konkaven Regionen negativ ist (siehe Abb. 3.8). Anhand der kombinierten Vorzeichen von  $k_1$  und  $k_2$  kann am Vorzeichen der *Gaußschen Krümmung* zwischen elliptischen  $K > 0$ , parabolischen  $K = 0$  und hyperbolischen  $K < 0$  Oberflächenregionen (siehe Abb. 3.9) unterschieden werden.

### Krümmungsberechnung auf diskreten Oberflächen

Mathematisch betrachtet haben diskrete Oberflächen (Polygonnetze) keine Krümmung, da all ihre Polygonflächen flach sind und die Netze an Kanten und Eckpunkten nicht zweimal differenzierbar sind. Unter der Annahme, dass die diskrete Oberfläche eine lineare Approximation einer unbekannten „glatten“ Oberfläche darstellt, kann versucht werden, die Krümmung dieser unbekannten Oberfläche zu bestimmen.

Grundlegend kann dazu zwischen zwei Vorgehensweisen unterschieden werden:

- *Surface Fitting* - Anschmiegen parametrischer Oberflächen
- *Discrete Differential Geometry Operators* - für Eckpunkte und Kanten in Dreiecksnetzen

**Surface Fitting:** Beim *Surface Fitting* werden parametrische, analytisch beschreibbare Oberflächen bestmöglich an eine lokale Umgebung um einen Punkt  $v$  des Polygonnetzes angeschmiegt. Im einfachsten Fall wird dazu eine quadratische Funktion verwendet, aus deren Parametern nach Gleichung 3.4  $k_1$  und  $k_2$  direkt berechnet werden können. Bei höhergradigen Funktionen und anderen Formen der Parametrisierung können die Krümmungswerte über die Weingartenabbildung bestimmt werden. In [Schulz 2005] wird ein guter Überblick über die verschiedenen Methoden und den mathematischen Hintergrund gegeben. Aufgrund der komplexen Natur der Krümmungsberechnung über *Surface Fitting* findet diese keine direkte Anwendung in Verfahren zur Glättung von Polygonnetzen.

**Discrete Differential Geometry Operators:** Im Gegensatz zu analytischen Oberflächen ist die Bestimmung von Normalen und Krümmungsmaßen bei diskreten polygonalen Oberflächennetzen Gegenstand aktueller Forschungen [Desbrun et al. 2006]. Operatoren der diskreten Differentialgeometrie bauen vereinfacht auf der Annahme simpler parametrischer Flächen (Kugel an Eckpunkten und Zylinder an Kanten) auf [Dyn et al. 2001]. Aus dieser Annahme leiten sich diskrete Operatoren für die Berechnung der Krümmungswerte ab.

Mittlere Krümmung  $H$  und Gaußsche Krümmung  $K$  ergeben sich dabei wie folgt:

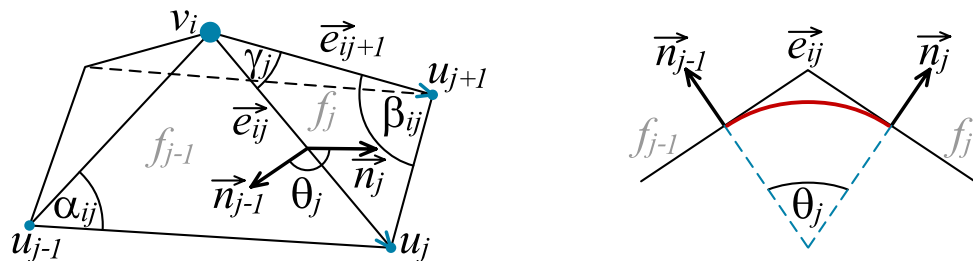
$$H = \frac{1}{4A} \sum_{j \in U_{v_i}^1} \|\vec{e}_{ij}\| \theta_j \quad (3.6)$$

$$K = \frac{1}{A} \left( 2\pi - \sum_{j \in U_{v_i}^1} \gamma_j \right) \quad (3.7)$$

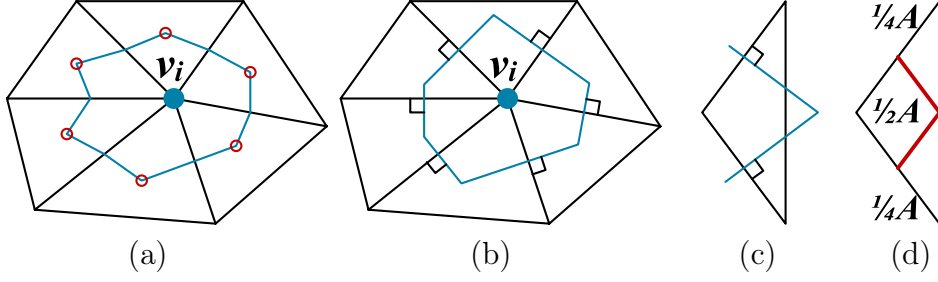
wobei  $\theta_j$  den Winkel zwischen den Normalen der zwei Flächen, welche die Kante  $\vec{e}_{ij}$  gemeinsam haben angibt und  $\gamma_j$  den Winkel zwischen den Kanten  $\vec{e}_{ij}$  und  $\vec{e}_{ij+1}$  an  $v_i$  angibt (siehe Abb. 3.10).

Eine anschauliche Herleitung der Formeln kann unter anderem in [Taubin 1995a; Dyn et al. 2001] und [Meyer et al. 2002] gefunden werden. Zur Normalisierung wird in den Formeln 3.6 und 3.7 unter Annahme einer gleichmäßigen Verteilung der Krümmung um den Eckpunkt  $v_i$  durch den Flächeninhalt  $A$  dividiert.

Für den eingesetzten Flächeninhalt  $A$  sind zwei unterschiedliche Varianten gebräuchlich. Am häufigsten wird der baryzentrische Flächeninhalt  $A_{barycentric}$  verwendet. Dieser ergibt sich aus der Fläche um den Punkt  $v_i$ , begrenzt durch die Mittelpunkte und Baryzenter,



**Abbildung 3.10:** Punkt  $v_i$  und zugehörige Variablen für die diskrete Krümmungsberechnung (links) und Querschnitt durch den angeschmiegt Zylinder an Kante  $\vec{e}_{ij}$  zwischen Dreieck  $f_{j-1}$  und  $f_j$  (rechts).



**Abbildung 3.11:** Flächeninhalte  $A_{barycentric}$  (a) und  $A_{voronoi}$  (b) für die Berechnung der Krümmung auf diskreten Oberflächen. Bei Dreiecken mit stumpfen Winkeln überragt die Voronoi-Region das Dreieck (c). In diesen Fällen wird  $A_{mixed}$  durch die Flächeninhalte  $\frac{1}{2}A$  bzw.  $\frac{1}{4}A$  (d) angenommen.

der ihn umgebenden Kanten und Dreiecke (siehe Abb.3.11 a).  $A_{barycentric}$  kann über die Flächeninhalte  $A(f_j)$  aller an  $v_i$  angrenzenden Dreiecke  $f_j$  wie folgt ermittelt werden:

$$A_{barycentric} = \frac{1}{3} \sum_{j \in U_{v_i}^1} A(f_j) \quad (3.8)$$

Um den Fehler der Approximation geringer zu halten, wird in der Literatur auch der Flächeninhalt der Voronoi-Region  $A_{voronoi}$  um den Punkt  $v_i$  (siehe Abb.3.11 b) verwendet.  $A_{voronoi}$  kann dabei grundlegend über

$$A_{voronoi} = \frac{1}{8} \sum_{j \in U_{v_i}^1} (\cot \alpha_{ij} + \cot \beta_{ij}) \|\vec{e}_{ij}\|^2 \quad (3.9)$$

berechnet werden. Die Winkel  $\alpha_{ij}$  und  $\beta_{ij}$  sind dabei (siehe Abb.3.10) die in den angrenzenden Dreiecken zu Kante  $\vec{e}_{ij}$  gegenüberliegenden Winkel. Probleme mit der Berechnung von  $A_{voronoi}$  ergeben sich in stumpfwinkligen Dreiecken, da hier die Voronoi-Region das betrachtete Dreieck überragt (siehe Abb.3.11 c). Meyer et al. [2002] schlagen in diesem Fall die Verwendung von  $A_{mixed}$  vor. Hierbei wird der Flächeninhalt für alle Dreiecke, die keinen stumpfen Winkel enthalten, über  $A_{voronoi}$  berechnet und der Flächeninhalt bei stumpfen Dreiecken als  $\frac{1}{2}A(f_i)$  (wenn der stumpfe Winkel in  $v_i$  liegt) bzw. als  $\frac{1}{4}A(f_i)$  (sonst) angenommen (siehe Abb.3.11 d). Das resultierende  $A_{mixed}$  für beliebige Punkte  $v_i$  summiert sich damit über das gesamte Oberflächennetz  $\mathcal{M}$  weiterhin zum gesamten Oberflächeninhalt  $A(\mathcal{M})$  auf, so dass weiter Gleichung 3.6 und 3.7 gelten. Mit den so berechenbaren Werten für die *Mittlere Krümmung*  $H$  und die *Gaußsche Krümmung*  $K$  können die Hauptkrümmungen  $k_1$  und  $k_2$  wie in Gleichung 3.5 dargelegt bestimmt werden.

Die soweit vorgestellten Operatoren stellen die Grundlage für aktuelle Glättungsverfahren dar. Es sei jedoch auf erweiterte Verfahren verwiesen, welche die begrenzte Betrachtung der direkten topologischen Nachbarn aufheben. Stokely und Wu [1992] verwenden einen geschlossenen Ring beliebiger Größe um einen Punkt und die darin eingeschlossene Oberfläche zur Approximation der Krümmung. Hameiri und Shimshoni [2002] und Page et al. [2002] schlagen eine geodätische Nachbarschaft mit einer Gewichtung der inversen geodätischen Entfernung als Modifikation von [Taubin 1995a] vor. Kalogerakis et al. [2007] verwenden sogar ein auf M-Estimation basierendes statistisches Modell zur Krümmungsapproximation in adaptiv angepassten Nachbarschaften.



## 3.6 Datenstrukturen für polygonale Oberflächennetze

Polygonnetze erlauben grundsätzlich die Anfrage, welche Flächen, Kanten und Eckpunkte zu jeder Fläche, Kante und jedem Eckpunkt des Netzes benachbart sind. Folglich ergeben sich neun unterschiedliche Nachbarschaftsbeziehungen. Welche dieser Nachbarschaften mit welchem Aufwand bestimmt werden können, hängt dabei von der internen Repräsentation der polygonalen Oberflächennetze ab.

Im Folgenden werden verbreitete Datenstrukturen für die Speicherung und Verarbeitung von polygonalen Oberflächennetzen besprochen. Der Fokus liegt dabei auf effizienten Zugriffsmöglichkeiten für Operationen, wie die Bestimmung topologischer Nachbarn und die Berechnung von Krümmungswerten, die für die Glättung von Oberflächennetzen relevant sind.

### 3.6.1 Explizite Repräsentation

In der expliziten Repräsentation wird jedes Polygon bzw. jede Fläche  $f$  über eine Liste von Koordinaten für jeden der  $n$  Eckpunkte von  $f$  repräsentiert [Foley et al. 1996]:

$$f = ((x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n))$$

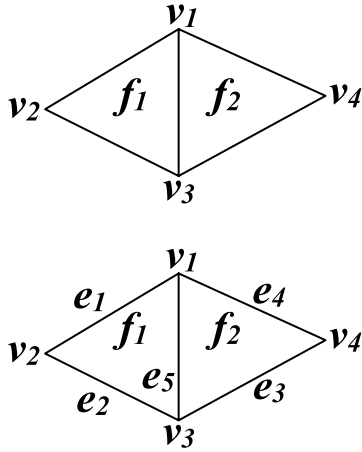
Die Eckpunkte werden dabei geordnet, meist entgegengesetzt dem Uhrzeigersinn angegeben. Die Kanten des Polygons ergeben sich jeweils zwischen einem Eckpunkt und dem darauf folgenden und zwischen dem letzten und ersten.

Für ein einzelnes Polygon ist diese Repräsentation effizient. Für ein Polygonnetz wird beim Speichern jedoch Platz durch das wiederholte Abspeichern der Eckpunktkoordinaten, die mehrere Polygone im Netz gemeinsam haben, verschwendet. Nachteilig ist, dass keine expliziten Informationen über gemeinsame Eckpunkte und Kanten im Polygonnetz existieren.

Die einzige gespeicherte Nachbarschaftsbeziehung repräsentiert, welche Eckpunkte (Eckpunktkoordinaten) zu einer Fläche gehören. Für die Bestimmung der topologischen Nachbarschaft eines Eckpunktes bedeutet dies, dass zunächst ein Vergleich des Koordinatentripels dieses Eckpunktes mit allen anderen Polygonen erfolgen muss, um alle Polygone zu finden, die diesen Eckpunkt gemeinsam haben. Im besten Fall ist das über die Sortierung der Punkte nach ihren Koordinaten mit  $N \log_2 N$  Aufwand möglich. Daraufhin können dann, aufgrund der Reihenfolge der gespeicherten Eckpunkte die topologischen Nachbarn erster Ordnung bestimmt werden, wobei auch hier wieder die Koordinatentripel verglichen werden müssen, um eine Mehrfachaufnahme von Nachbarpunkten zu vermeiden. Darüber hinaus besteht die größte Gefahr beim Vergleich der Koordinatentripel darin, dass sich durch Rechenungenauigkeiten die Koordinaten gemeinsamer Eckpunkte leicht unterscheiden.

### 3.6.2 Indizierte Eckpunkt- und Kantenlisten

Bei der Repräsentation von Polygonnetzen über indizierte Eckpunktlisten (*Indexed Face Sets*) werden die Koordinaten jedes Eckpunktes im Netz nur einmal in einer Liste  $\mathcal{V} = ((x_1, y_1, z_1), \dots, (x_n, y_n, z_n))$  abgespeichert. Die Flächen bzw. Polygone des Netzes werden dann über Zeiger in diese Eckpunktliste, wie in Abbildung 3.12 (oben) definiert. Diese Repräsentation hat gegenüber der expliziten Polygonrepräsentation den Vorteil, dass jeder Eckpunkt nur einmal gespeichert werden muss und dass die Koordinaten eines Eckpunktes nun sehr einfach verändert werden können. Nun ist die Information, welche Eckpunkte zu einem Polygon gehören, explizit ersichtlich und weitere Nachbarschaften können über eine globale



$$\begin{aligned}\mathcal{V} &= \{v_1, v_2, v_3, v_4\} = \{(x_1, y_1, z_1), \dots, (x_4, y_4, z_4)\} \\ f_1 &= \{v_1, v_2, v_3\} \\ f_2 &= \{v_1, v_3, v_4\}\end{aligned}$$

$$\begin{aligned}\mathcal{V} &= \{v_1, v_2, v_3, v_4\} = \{(x_1, y_1, z_1), \dots, (x_4, y_4, z_4)\} \\ e_1 &= \{v_1, v_2, f_1, \emptyset\} & f_1 &= \{e_1, e_2, e_5\} \\ e_2 &= \{v_2, v_3, f_1, \emptyset\} & f_2 &= \{e_5, e_3, e_4\} \\ e_3 &= \{v_3, v_4, f_2, \emptyset\} \\ e_4 &= \{v_4, v_1, f_2, \emptyset\} \\ e_5 &= \{v_1, v_3, f_1, f_2\}\end{aligned}$$

**Abbildung 3.12:** Repräsentation eines Polygonnetzes über eine indizierte Eckpunktliste (oben) und über eine indizierte Kantenliste (unten).

Suche bestimmt werden. Dadurch ist es z.B. jedoch immer noch sehr aufwändig, Polygone zu finden, die sich eine Kante teilen. Werden die Kanten im Netz zusätzlich explizit repräsentiert, so lässt sich diese Suche vereinfachen.

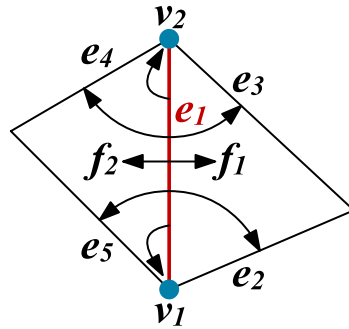
Zum Repräsentieren von Polygonen über Indizes einer Kantenliste werden die Polygone über Zeiger auf ihre Kanten repräsentiert. Dazu werden die Kanten zusätzlich zu der Eckpunktliste in einer Kantenliste abgespeichert. Jede Kante zeigt dann auf ihre Eckpunkte in der Eckpunktliste, wie in Abbildung 3.12 (unten) illustriert.

Über die somit vorhandene Information, welche Kanten zu einem Polygon gehören, können nun einfacher alle Polygone gefunden werden, die sich eine Kante teilen. Dies ist jedoch nicht in konstanter Zeit möglich.

### 3.6.3 Winged-Edge Repräsentation

In keiner der zuvor genannten Repräsentationen ist es ohne Umstände möglich zu bestimmen, welche Kanten in einem Eckpunkt zusammenlaufen. Bei den indizierten Kantenlisten müssten hierzu alle Kanten inspiziert werden. Eine verbreitete Datenstruktur, die dafür explizite Informationen zur Verfügung stellt, ist die von Baumgart [1972, 1975] entwickelte *Winged-Edge* Repräsentation. Wie in Abbildung 3.13 dargestellt, wird jede Kante in der *Winged-Edge* Datenstruktur durch Zeiger auf ihre zwei Eckpunkte und ihre zwei angrenzenden Flächen sowie durch Zeiger auf die Vorgänger- und Nachfolgerkante in den zwei angrenzenden Flächen repräsentiert. Darüber hinaus hat jeder Eckpunkt einen Zeiger auf eine der aus ihm hervorgehenden Kanten und jede Fläche einen Zeiger auf eine ihrer Kanten.

Aus den möglichen neun Nachbarschaftsbeziehungen erlaubt die *Winged-Edge* Datenstruktur in konstanter Zeit die Bestimmung von Eckpunkten und Flächen, die mit einer Kante assoziiert sind. Durch das explizite Speichern weiterer Nachbarschaftsbeziehungen (z.B. Zeiger von jedem Eckpunkt auf jede in ihm zusammenlaufende Kante) kann die Bestimmung weiterer Informationen sowie die Verarbeitung der Oberflächennetze (z.B. bei der Glättung und Krümmungsberechnung) weiter beschleunigt werden.



**Abbildung 3.13:** *Winged-Edge* Datenstruktur der Kante  $e_1$ . Zusätzlich haben  $v_1$ ,  $v_2$ ,  $f_1$  und  $f_2$  einen Zeiger auf eine ihrer Kanten (nicht mit dargestellt).

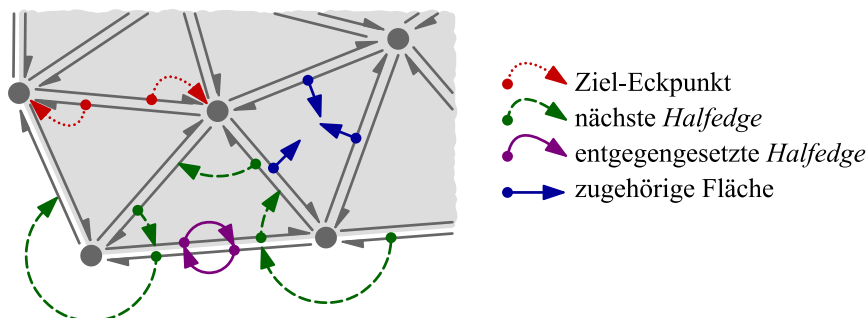
### 3.6.4 Halfedge Repräsentation

Erweiterte Varianten der kantenbasierten *Winged-Edge* Datenstruktur repräsentieren eine Kante jeweils über zwei sogenannte „*Halfedges*“, wobei die *Halfedges* einer Polygonkante jeweils in die entgegengesetzte Richtung zeigen und jede eine der zwei an dieser Kante zusammentreffenden Flächen entgegen dem Uhrzeigersinn umgrenzt. Abbildung 3.14 illustriert die *Halfedge* Datenstruktur mit allen gespeicherten Zeigern auf die angrenzende Fläche, den nächsten Eckpunkt sowie auf die nächste und die entgegengläufige *Halfedge*. Für die Bestimmung der topologischen Nachbarschaft erster Ordnung eines Eckpunktes muss dann, wie in Abbildung 3.15 dargestellt, lediglich wiederholt drei Zeigern der Datenstruktur gefolgt werden, was in konstanter Zeit möglich ist.

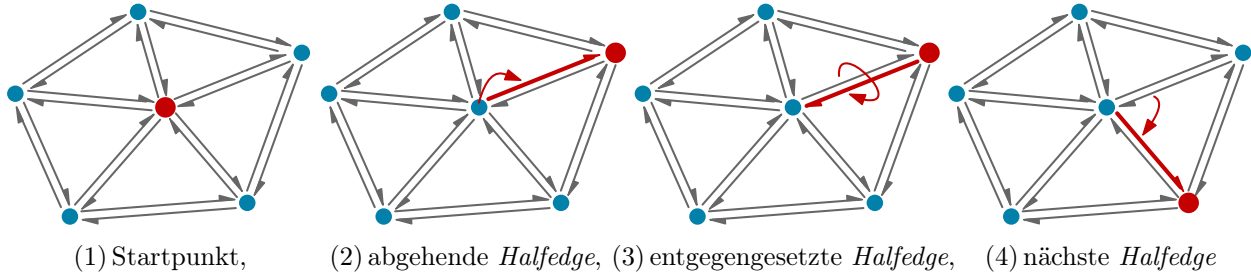
Aufgrund der gegebenen Flexibilität und gleichzeitigen Effizienz finden *Halfedge* Datenstrukturen unter anderem in Polygonnetz-Bibliotheken wie CGAL [CGAL 2007] und OPENMESH [Botsch et al. 2002; OpenMesh 2007] Anwendung. Einen aktuellen Überblick über Datenstrukturen und Anwendungen geben Botsch et al. [2007].

### 3.6.5 Effiziente Repräsentation für die diskrete Krümmungsberechnung

Aus den Vorbetrachtungen der Krümmungsberechnung über diskrete Operatoren (siehe Abschnitt 3.5) lassen sich die Mindestanforderungen an eine spezialisierte, laufzeiteffiziente Datenstruktur aufstellen:



**Abbildung 3.14:** *Halfedge* Datenstruktur und die pro *Halfedge* gespeicherten Zeiger. Jede Fläche und jeder Eckpunkt haben zusätzlich noch einen Zeiger auf eine ihrer *Halfedges* (nicht mit dargestellt).

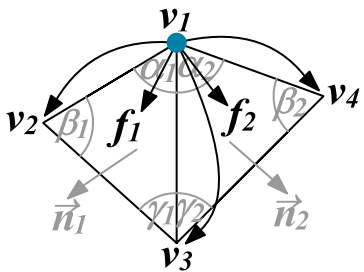


**Abbildung 3.15:** Nötige Schritte zum Traversieren der topologischen Nachbarn eines Eckpunktes in der *Halfedge* Datenstruktur. Die Schritte (3) und (4) sind dabei für alle Nachbarn zu wiederholen. (modifiziert aus [OpenMesh 2007])

- nur Dreiecke sind als Polygone bzw. Flächen im Netz zulässig
- effizienter Zugriff auf die topologischen Nachbarn erster Ordnung von jedem Eckpunkt
- effizienter Zugriff auf die Flächen, die in jedem Eckpunkt zusammenlaufen

Da die Operatoren für die diskrete Differentialgeometrie und Krümmungsberechnung nur auf Dreiecksnetzen definiert sind, kann eine entsprechend rein auf Dreiecke spezialisierte Datenstruktur verwendet werden. Die topologischen Nachbarn erster Ordnung sind erforderlich, um alle nötigen Informationen (Kantenlängen und Winkel) für die Berechnungen in Formel 3.6 und 3.7 zu ermitteln. Ein effizienter Zugriff auf die Flächen, die in jedem Eckpunkt zusammenlaufen, ist von Vorteil, um die Flächennormalen und Winkel nicht mehrmals neu berechnen zu müssen.

Folglich ist, wie in Abbildung 3.16 illustriert, für eine effiziente minimale Repräsentation der Polygonnetze nur eine Liste aller Eckpunkte und jeweils eine Liste mit Zeigern auf alle Nachbarnpunkte und auf alle anliegenden Flächen von jedem Eckpunkt nötig. Dies stellt eine vorwiegend eckpunktbasierte Polygonnetzrepräsentation dar. Kanten und Flächen des Netzes ergeben sich implizit zwischen den gegebenen Nachbarnpunkten und durch die Beschränkung auf reine Dreiecksnetze. Pro Fläche reicht es aus, den ersten Eckpunkt, die Dreieckswinkel (beim ersten Eckpunkt beginnend) und die Normale zu speichern. Auch wenn diese Repräsentation für die Krümmungsberechnung und darauf aufbauende Operationen wie die Glättung effizient ist, so ist sie z.B. für das Rendern des Polygonnetzes oder für topologieverändernde Operationen eher ungeeignet.



$$\begin{aligned}
 \mathcal{V} &= \{v_1, v_2, v_3, v_4\} = \{(x_1, y_1, z_1), \dots, (x_4, y_4, z_4)\} \\
 \mathcal{U}_{v_1}^1 &= \{v_2, v_3, v_4\} & \mathcal{F}_{v_1} &= \{f_1, f_2\} & f_1 &= \{v_1, \alpha_1, \beta_1, \gamma_1, \vec{n}_1\} \\
 \mathcal{U}_{v_2}^1 &= \{v_1, v_3\} & \mathcal{F}_{v_2} &= \{f_1\} & f_2 &= \{v_1, \alpha_2, \beta_2, \gamma_2, \vec{n}_2\} \\
 \mathcal{U}_{v_3}^1 &= \{v_4, v_1, v_2\} & \mathcal{F}_{v_3} &= \{f_2, f_1\} \\
 \mathcal{U}_{v_4}^1 &= \{v_1, v_3\} & \mathcal{F}_{v_4} &= \{f_2\}
 \end{aligned}$$

**Abbildung 3.16:** Effiziente Polygonnetzrepräsentation für die Krümmungsberechnung mittels diskreter Operatoren.

# Strategien zur Artefaktreduktion

Abbildung 4.1 stellt die Pipeline bei der Erzeugung von Oberflächenmodellen aus segmentierten Volumendaten dar. Ausgehend von der binären Segmentierungsmaske auf der Bildebene werden über Algorithmen zur Oberflächenextraktion Polygonnetze erstellt. Die extrahierten Oberflächenmodelle bilden damit die Oberflächennetzebene, auf der dann Visualisierungs- und Interaktionstechniken ansetzen.

Um Artefakte zu reduzieren und damit die Qualität von extrahierten Oberflächenmodellen aus binären Volumendaten zu verbessern, existieren im Rahmen der Extraktionspipeline drei grundlegende Strategien: (1) *Glättung der binären Segmentierungsmaske*, (2) *Artefaktreduktion bei der Extraktion von Oberflächenmodellen* und (3) *Glättung der extrahierten Oberflächenmodelle* (siehe Abb. 4.1).

Im Folgenden wird der Stand der Technik in den verschiedenen Bereichen besprochen. Dabei wird bereits auf die Eignung existierender Techniken in den unterschiedlichen Schritten der Oberflächenextraktionspipeline eingegangen.

## 4.1 Glättung von binären Segmentierungsmasken

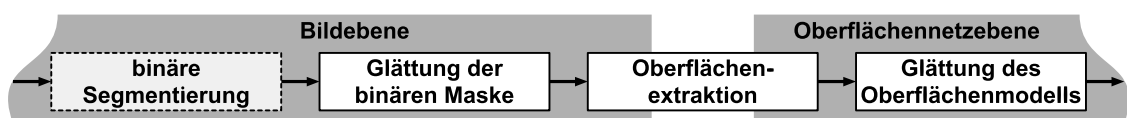
Die erste Möglichkeit zur Reduktion von Artefakten stellt die direkte Filterung und Manipulation der Bilddaten dar. Existierende Ansätze verwenden dazu einfache Tiefpass-Filter, erzeugen *fuzzy boundaries*, interpolieren Sub-Voxel oder glätten mit Level-Sets und einem aus der Polygonnetzglättung abgeleitetem Glättungsverfahren.

### 4.1.1 Tiefpass-Filterung

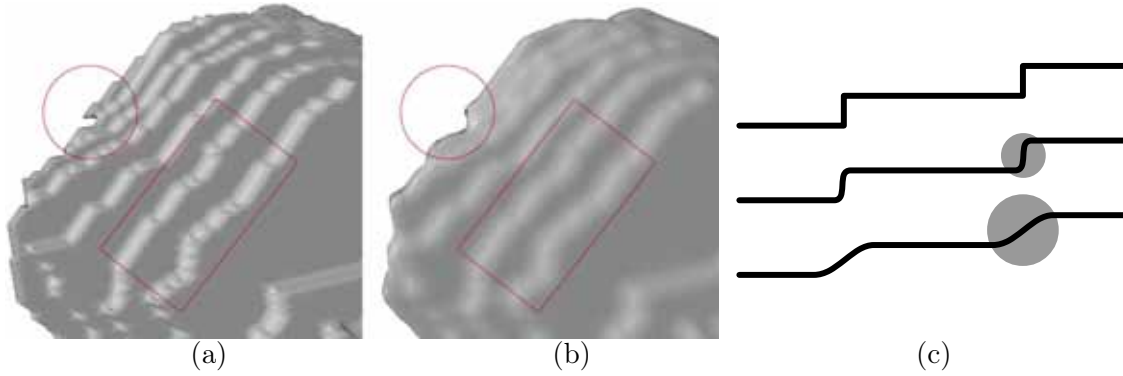
Da die Unstetigkeiten in der binären Segmentierungsmaske im Frequenzraum ein unbegrenztes Frequenzspektrum darstellen, kann als einfachste Möglichkeit eine Tiefpass-Filterung zur Unterdrückung hoher Frequenzen bei der Reduktion von Block- und Terrassenartefakten in der binären Maske angewandt werden [Šramek und Kaufman 1998].

Abbildung 4.2 (a) und (b) zeigen die extrahierten Oberflächenmodelle aus einer binären Segmentierungsmaske vor und nach der Tiefpass-Filterung. Artefakte, die größer als der verwendete Filterkernel sind, können dabei nicht entfernt werden (siehe Viereck). Abbildung 4.2 (c) illustriert den Zusammenhang zwischen Filtergröße und erreichter Glättung.

Die Filtergröße ist dabei aber auch dafür verantwortlich, dass Details (siehe Kreis in Abb. 4.2 (a) und (b)) „weggeglättet“ bzw. entfernt werden können. Dies äußert sich in der Bildebene, indem Voxel umkippen. Das bedeutet, dass Voxel, die zuvor einen Wert  $i > i_{iso}$  hatten, nach der Glättung einen Wert  $i < i_{iso}$  besitzen und umgekehrt. Eine mögliche Lösung besteht darin, sicherzustellen, dass alle Voxel entsprechend ihrem initialen Wert immer



**Abbildung 4.1:** Pipeline zur Erzeugung von Oberflächenmodellen aus Volumendaten. Grob kann zwischen Teilschritten auf der Bildebene und/oder Oberflächennetzebene unterschieden werden.



**Abbildung 4.2:** Tiefpass-Filterung der Segmentierungsmaske: (a) extrahierte Oberfläche aus einer binären Segmentierungsmaske, (b) extrahierte Oberfläche nach einer 5x5x5 Gaußfilterung der binären Maske. (c) Abhängigkeit von Kernelgröße und Glättung von Terrassenartefakten.

einen Wert größer bzw. kleiner dem Iso-Wert  $i_{iso}$  annehmen. Bei einer größeren Anzahl an Iterationen führt dies jedoch wieder zu Artefakten. Werden die Voxelwerte auf  $\geq i_{iso} + \delta$  bzw.  $\leq i_{iso} - \delta$  mit einem beliebigen  $\delta$  beschränkt, so nehmen nach mehreren Glättungsiterationen die Voxel an der Objektgrenze die Werte  $i_{iso} + \delta$  bzw.  $i_{iso} - \delta$  an, was wiederum eine binäre Klassifikation ergibt.

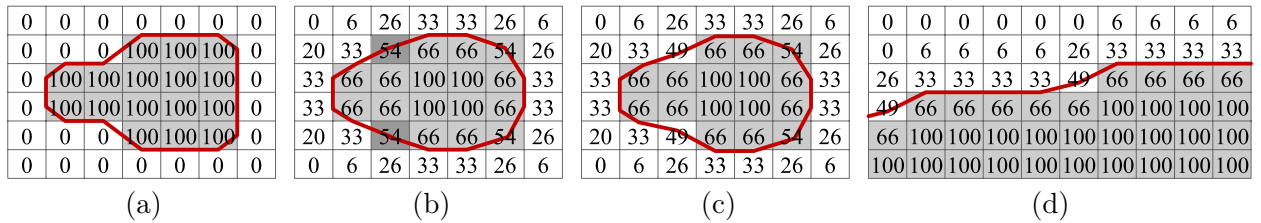
### 4.1.2 Fuzzy Boundaries

Für die direkte Volumenvisualisierung wurden deshalb Strategien zur Erzeugung von *fuzzy boundaries* entwickelt [Lakare und Kaufman 2003; Neubauer et al. 2004; Neubauer 2005]. Hierbei werden in einem kleinen Bereich um die Objektgrenze die Voxelwerte idealerweise linear vom Hintergrund zum Inneren des Objektes hin erhöht.

Lakare und Kaufman [2003] verwenden die Originaldaten, um den Werteverlauf an den Grenzen der Segmentierungsmaske als *fuzzy boundaries* repräsentieren zu können. Um dadurch die Glättung einer binären Segmentierung zu erreichen, sind die Originaldaten erforderlich und es muss einen geeigneten Grauwertverlauf an der Objektgrenze in den Originaldaten geben. Ein geeigneter Grauwertverlauf ist dabei ein kontinuierliches Abnehmen bzw. Zunehmen des Grauwertes vom Inneren des Objektes zu den umliegenden Strukturen (siehe z.B. Abb. 3.2 (a) S. 28). Ist dies aber ähnlich wie in Abbildung 3.2 (b) und (c) nicht der Fall, kann das Verfahren keine überzeugenden Ergebnisse liefern.

Der Ansatz von Neubauer et al. [2004] wendet morphologische Operatoren in einem definierten Gebiet um die Segmentierungsgrenze an, um abstandskorreliert die Voxelwerte in der Grenzregion abstufen zu können. Hierbei wird zusätzlich darauf geachtet, dass die ursprüngliche Objekt-Hintergrund-Klassifikation der Voxel beibehalten wird. Wie Abbildung 4.3 (a)-(c) illustriert, ist das Verfahren in der Lage, Artefakte aus binären Segmentierungsmasken kleiner Objekte gut zu reduzieren. Größere Terrassenartefakte und Plateaus können damit jedoch nicht reduziert werden (siehe Abb. 4.3 (d)).

Zusätzlich bleibt anzumerken, dass *fuzzy boundaries* für die direkte Volumenvisualisierung geeignet sind, da die Grenzen der Objekte im Volumerendering dadurch insgesamt weicher werden. Bei der Extraktion von Oberflächen aus binären Daten ermöglichen die Verfahren jedoch keine ausreichende Reduktion der Artefakte.

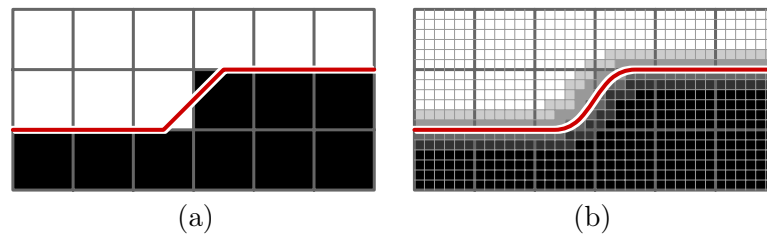


**Abbildung 4.3:** Glättung binärer Maske nach [Neubauer et al. 2004]: (a) Originalmaske mit Oberfläche, (b) nach Glättung mit morphologischen Operatoren, (c) Korrektur der Objekt-Hintergrund-Klassifikation, (d) größere Objekte mit Plateaus können nicht ausreichend geglättet werden.

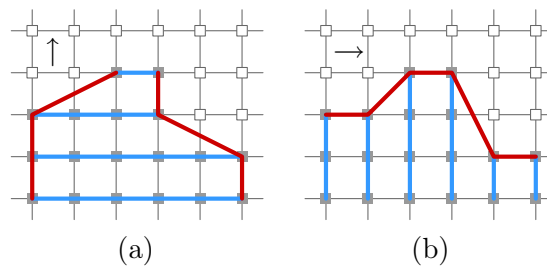
### 4.1.3 Sub-Voxel Interpolation

Andere Methoden interpolieren Zwischenschichten bzw. *Sub-Voxel* zur Glättung der binären Maske. Diese Methoden können, wie in Abbildung 4.4 dargestellt, grundlegend nur Blockartefakte aber keine Terrassenartefakte reduzieren.

Deshalb verwenden Albu et al. [2003] zur Berechnung der Zwischenschichten im *Sub-Voxel*-Bereich morphologische Operatoren, um eine Oberflächenkontur aus einer Originalschicht über mehrere Sub-Voxelschichten in eine Kontur einer anderen Originalschicht übergehen zu lassen. Das morphologische Überführen von einer Kontur in einer Schicht in die Kontur in der benachbarten Schicht macht das Verfahren jedoch richtungsabhängig. Abbildung 4.5 zeigt die unterschiedlichen Ergebnisse beim morphologischen Überführen der Konturen in horizontale und vertikale Richtung in 2D. Durch diese richtungsabhängige Glättung der Maske können Artefakte nur in der gewählten Richtung reduziert werden. Albu et al. [2003] führen deshalb nach der Oberflächenextraktion zusätzlich eine Glättung der Oberflächennetze durch.



**Abbildung 4.4:** Interpolation von Sub-Voxeln zur Glättung von binären Masken: (a) binäre Maske und (b) berechnete Sub-Voxel mit jeweils resultierendem Oberflächenverlauf.



**Abbildung 4.5:** Richtungsabhängiges morphologisches Überführen von Objektkonturen in einer Schicht in eine benachbarte Schicht nach Albu et al. [2003]. (a) Die Konturen liegen in horizontalen Schichten und werden vertikal ineinander überführt; (b) Die Konturen liegen in vertikalen Schichten und werden horizontal ineinander überführt.

Bei Sub-Voxelverfahren allgemein ist zusätzlich zu berücksichtigen, dass sich bei der Berechnung der Zwischenschichten und *Sub-Voxel* die Datenmenge drastisch erhöhen kann.

#### 4.1.4 Level-Sets

Eine weitere Methode zur Artefaktreduktion der binären Segmentierungsmaske stellen die von Whitaker [2000] auf die Bildebene übertragenen *constrained elastic surface nets* (CESN, siehe Abschnitt 4.2.4) dar. Dieser *Level-Sets*-Ansatz ist jedoch im Vergleich zum originalen Netzglättungsalgorithmus sehr rechenaufwändig. Sinnvoll ist dieser Ansatz jedoch dann, wenn eine direkte Volumenvisualisierung der binären Segmentierungsmaske erwünscht ist, da dann der vergleichbare Weg mit CESN über die Extraktion, Glättung und anschließende 3D-Rasterung (Voxelization) einer Oberfläche ineffizient wird [Whitaker 2000].

## 4.2 Artefaktreduktion bei der Oberflächenextraktion

Das Standardverfahren zur Extraktion von Oberflächenmodellen aus Volumendaten stellt *Marching Cubes* (MC) [Lorensen und Cline 1987] dar. Hierbei werden jeweils vier benachbarte Voxel aus einer Schicht und die dazu benachbarten vier Voxel aus einer angrenzenden Schicht zu Quadern bzw. allgemeiner zu Zellen zusammengefasst (siehe Abb. 3.4 S. 30). Diese können dann unabhängig voneinander betrachtet und ein Oberflächenmodell extrahiert werden. Da hierbei zur Bestimmung der zu extrahierenden Oberfläche aber nur eine lineare Interpolation auf den Zellgrenzen verwendet wird, enthalten extrahierte Oberflächenmodelle Diskontinuitäten und sind mathematisch nur  $C^0$ -stetig (siehe Abb. 4.6 (b)).

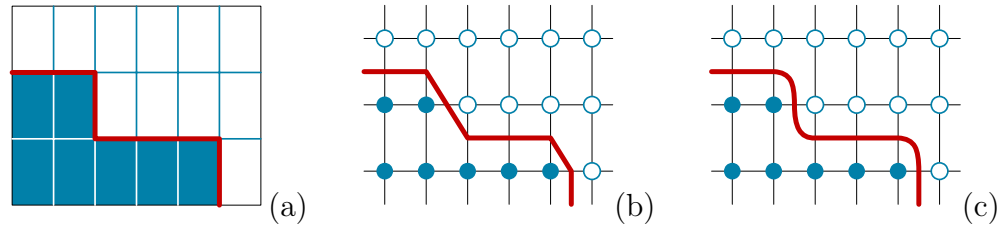
Da der originale MC-Algorithmus alle Zellen eines Datensatzes betrachtet, wird meist viel Rechenzeit damit vergeudet, Zellen zu untersuchen, die nicht zur Oberfläche beitragen. Zahlreiche Varianten zur Beschleunigung des Verfahrens verwenden deshalb hierarchische Datenstrukturen, um möglichst viele Zellen, die nicht zur Oberfläche beitragen, von der Betrachtung ausschließen zu können [Wilhelms und Gelder 1992; Itoh und Koyamada 1995; Cignoni et al. 1997].

Auch die topologische Konsistenz und Korrektheit ist im originalen MC-Algorithmus nicht gewährleistet. So können durch Mehrdeutigkeiten in der Triangulierung einzelner Fälle in der Lookup-Tabelle des MC-Algorithmus Löcher in extrahierten Oberflächen auftreten. Entsprechend existieren Erweiterungen, die diese Löcher vermeiden und/oder die korrekte Topologie in solchen Fällen berechnen [Dürst 1988; Nielson und Hamann 1991; Montani et al. 1994; Chernyaev 1995]. Die korrekte Topologie kann dabei über trilineare Interpolation in den topologisch mehrdeutigen Zellen ermittelt werden [Chernyaev 1995]. Bei binären Daten führt eine trilineare Interpolation in den mehrdeutigen Fällen dazu, dass für jeden Fall immer dieselbe Topologie verwendet wird und somit eine starre Lookup-Tabelle für alle Standardfälle ausreicht. Dies führt, unter Betrachtung der Originaldaten, nicht zwingend zu topologisch korrekten, aber immer zu topologisch konsistenten Oberflächen.

### 4.2.1 Trilineare Interpolation

Die trilineare Interpolation wird neben der Topologiebestimmung in den Zellen auch dazu verwendet, die Genauigkeit und Glattheit der extrahierten Oberflächen zu verbessern [Hamann et al. 1997; Allamandri et al. 1998; Cignoni et al. 2000; Theisel 2002; Lopes und Brodlie 2003]. Dabei wird die Genauigkeit und Glattheit gesteigert, indem zusätzlich zur





**Abbildung 4.6:** Oberflächenextraktion aus binären Segmentierungsmasken: (a) binäre Segmentierungsmaske, (b) MC extrahierte Oberfläche, (c) extrahierte Oberfläche mit tri-linearer Interpolation im Zellinneren nach [Allamandri et al. 1998].

Interpolation auf den Zellgrenzen auch innerhalb der Zellen über trilineare Interpolation Punkte erzeugt werden [Allamandri et al. 1998; Cignoni et al. 2000]. Entsprechend wird die extrahierte Oberfläche innerhalb der Zellen glatter. Jedoch ist es auf diese Weise nicht möglich, Terrassenartefakte, wie sie aus binären Segmentierungsmasken entstehen, zu reduzieren (siehe Abb. 4.6 (c)).

### 4.2.2 Sub-Voxel Interpolation

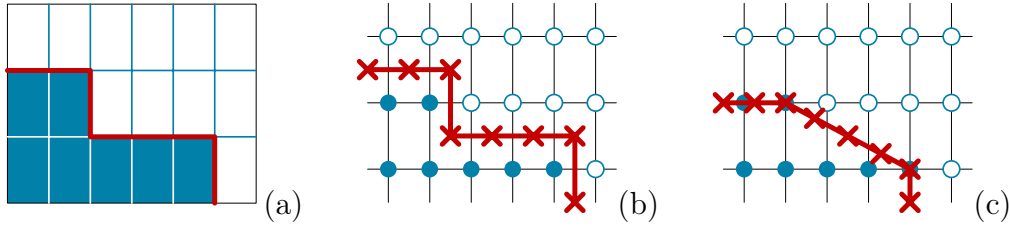
Weitere Extraktionsmethoden beruhen direkt auf der Interpolation von Zwischenschichten bzw. *Sub-Voxeln* und unterliegen damit den in Abschnitt 4.1.3, Abbildung 4.4 und 4.5 bereits beschriebenen Problemen. Auch ein fortgeschrittener Sub-Voxel Ansatz, wie das in Abschnitt 4.1.3 beschriebene Verfahren von Albu et al. [2003], kommt dabei nicht ohne eine zusätzliche Glättung der extrahierten Oberflächennetze aus.

### 4.2.3 Approximation mit impliziten Oberflächen

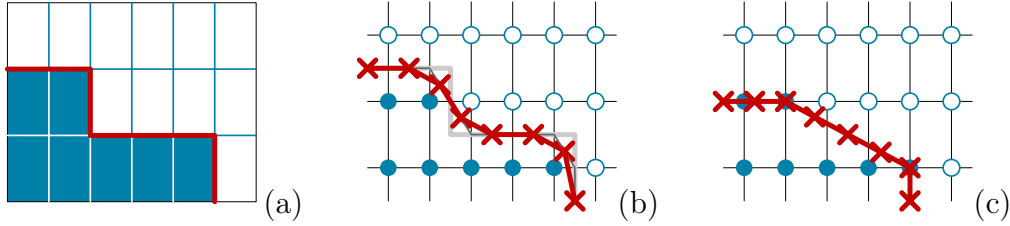
Zur approximativen oder modellbasierten Bestimmung von Oberflächenmodellen aus binären Segmentierungsmasken finden Techniken Anwendung, die 3D-Primitive wie z.B. Zylinder und Kegelstümpfe an die Segmentierung anpassen [Masutani et al. 1996; Hahn et al. 2001] bzw. implizite Oberflächen erzeugen [Grimm und Hughes 1995; Oeltze und Preim 2005; Braude 2005; Braude et al. 2007]. Verfahren, die dabei einen modellbasierten Ansatz verfolgen, eignen sich nur, solange die Modellannahmen für das zu extrahierende Oberflächenmodell zutreffen und sind somit nur in Spezialfällen einsetzbar. Bei Masutani et al. [1996]; Hahn et al. [2001]; Oeltze und Preim [2005] ist dies die Visualisierung von gesunden Blutgefäßen, wobei hier der Fokus auf der korrekten Darstellung der Topologie der Gefäße, aber nicht der genauen Geometrie (z.B. der exakten Ausdehnungen) liegt. Modellfreie Verfahren, die implizite Oberflächen erzeugen, um artefaktfreie Oberflächenmodelle extrahieren zu können (siehe [Grimm und Hughes 1995; Braude 2005; Braude et al. 2007]), leiden unter einer über zumeist sehr viele Parameter schlecht steuerbaren Genauigkeit der erzeugten Oberflächen.

### 4.2.4 Kombinierte Oberflächenextraktion und Glättung

Einen weiteren Ansatz zur Artefaktreduktion stellen Oberflächenextraktionstechniken dar, die sich aus einer Kombination aus spezieller Extraktionsmethode, angepasster nachfolgender Glättung und eventuell weiterer Verarbeitung zusammensetzen. Zu dieser Kategorie gehören die *Constrained Elastic Surface Nets* (CESN) [Gibson 1998] und *Dual Marching Cubes* (DualMC) [Nielson 2004].



**Abbildung 4.7:** CESN für binäre Segmentierungsmasken: (a) binäre Segmentierungsmaske, (b) initial extrahierte Oberfläche mittels CESN, (c) Glättung der Oberfläche beschränkt auf Originalvoxel mit CESN.



**Abbildung 4.8:** DualMC für binäre Segmentierungsmasken: (a) binäre Segmentierungsmaske, (b) initial extrahierte Oberfläche mittels DualMC, (c) Glättung der Oberfläche beschränkt auf Originalvoxel über die iterative Erzeugung von *Duals*.

Bei den *Constrained Elastic Surface Nets* CESN [Gibson 1998] werden bei der Extraktion der Oberfläche ohne jegliche Interpolation die Zellmittelpunkte aller Grenzzellen als Oberflächenpunkte extrahiert und verbunden (Abb. 4.7 (b)). Diese zunächst kantige Oberfläche wird daraufhin über die iterative Minimierung der Kantenenergie, die der Laplace-Glättung entspricht, geglättet [Gibson 1998]. Um ein Schrumpfen der Oberfläche durch das Glätten zu minimieren, wird dabei darauf geachtet, dass die erzeugten Oberflächenpunkte nicht die Zelle verlassen, zu der sie gehören. Abbildung 4.7 illustriert die initial extrahierte Oberfläche und das Ergebnis der Glättung mit Beschränkung auf die Originalvoxel.

Ähnlich wird beim DualMC-Ansatz von Nielson [2004] vorgegangen, bei dem die initial extrahierte Oberfläche aber bereits das *Dual* der initial bei CESN erzeugten Oberfläche ist (Abb. 4.8 (b)). Daraufhin werden iterativ *Duals* der aktuellen Oberfläche erzeugt, um eine Glättung zu erreichen. Da dies auch zum Schrumpfen führen würde, wird auch hierbei darauf geachtet, dass die extrahierten Oberflächenpunkte nicht ihre Originalzelle verlassen. Die erreichbaren Ergebnisse entsprechen dabei denen der mit CESN erzeugten Oberflächen (vergl. Abb. 4.7 (c) und 4.8 (c)).

Diese kombinierten Verfahren erfordern eine spezielle Extraktionsmethode und stellen danach grundlegend Verfahren zur Glättung von Oberflächenmodellen dar.

### 4.3 Glättung von extrahierten Oberflächenmodellen

Zur Glättung von Oberflächenmodellen stehen zahlreiche publizierte Verfahren, von einfachen *Laplace*-Filtern bis hin zu erweiterten *Mean Curvature Flow* oder *Anisotropic Diffusion* Filtern, zur Verfügung (siehe u.a. [Taubin 1995b; Desbrun et al. 1999; Bajaj und Xu 2003; Hildebrandt und Polthier 2004]).

Hauptanwendungsziel existierender Polygonnetzglättungsverfahren ist die Reduktion von Rauschen in modellierten oder gescannten Oberflächenmodellen (siehe u.a. [Belyaev und

Ohtake 2003; Desbrun et al. 1999; Vollmer et al. 1999; Yagou et al. 2002b]). Erweiterte Verfahren bemühen sich neben der effizienten Rauschreduktion zusätzlich, das Originalvolumen und scharfe Kanten der Modelle zu erhalten, da diese unter der Glättung tendenziell schrumpfen bzw. verwischen.

### 4.3.1 Laplace-Glättung

Der Laplace-Filter, auch *Laplacian Flow* oder *Relaxationsfilter* genannt, ist der einfachste Filter zur Glättung von Oberflächenmodellen. Er erzeugt effizient glatte Oberflächenmodelle bereits nach wenigen Iterationen.

Der Laplace-Filter iteriert über alle Oberflächenpunkte (*vertices*) und verschiebt diese ins geometrische Zentrum seiner Nachbarn. Im Allgemeinen wird hierzu die topologische Nachbarschaft 1. Grades (siehe Abschnitt 3.4.1) verwendet. Die Prozedur kann eine beliebige Anzahl  $\iota$  von Iterationen wiederholt werden, bis die gewünschte Glättung erreicht ist.

Da dieses Vorgehen Oberflächenmodelle stark schrumpfen lässt, wird in der Regel ein Wichtungsfaktor  $\lambda$  verwendet. Dieser Faktor reguliert den Einfluss der Nachbarn auf die neue Position des aktuell betrachteten Oberflächenpunktes bei jeder Iteration. Die neue geglättete Position  $v_i'$  des  $i$ -ten Eckpunktes aller Eckpunkte  $\mathcal{V}$  des Oberflächennetzes ergibt sich dabei aus seiner alten Position  $v_i$  und den Positionen seiner direkten Nachbarn  $\mathcal{U}_{v_i}^1 = \{u_1, \dots, u_m\}$ , wie in Formel 4.1 und Abbildung 4.9.

$$v_i' = v_i + \frac{\lambda}{m} \sum_{j=1}^m (u_j - v_i) \quad v_i, u_j \in \mathcal{V}, v_i \neq u_j, \forall u_j \in \mathcal{U}_{v_i}^1 \quad m = \|\mathcal{U}_{v_i}^1\| \quad (4.1)$$

In dieser Form repräsentiert der Laplace-Filter mit seinen zwei Parametern (Anzahl der durchgeführten Iterationen  $\iota$ , Wichtungsfaktor  $\lambda$ ) einen der verbreitetsten Algorithmen zur Glättung von Oberflächenmodellen.

Wie Abbildung 4.9 andeutet, schrumpfen Oberflächen unter Verwendung des Laplace-Filters stark, wobei selbst die Wahl eines geringen Wichtungsfaktors (z.B.  $\lambda = 0,01$ ) dies nur verzögern, aber nicht verhindern kann. Zusätzlich kommt es durch den einfachen Ansatz, die Eckpunkte ins Zentrum ihrer Nachbarn zu verschieben, zu tangentialen Verschiebungen der Eckpunkte. Dies äußert sich, indem bei einer ungleichmäßig verteilten Nachbarschaft die Eckpunkte während der Glättung über die Oberfläche wandern und dessen Gestalt verzerren können (siehe Abb. 4.10 (b)). Um die unterschiedliche Auswirkung von ungleichmäßig entfernten Nachbarn zu reduzieren, kann der Einfluss jedes Nachbarnpunktes umgekehrt proportional zu seiner Entfernung gewichtet werden. Mit  $\|\vec{e}_{ij}\|$  als Kantenlänge zwischen  $v_i$  und seinem Nachbarn  $u_j$  ergibt sich eine abstandsabhängige bzw. *Scale-Dependent* Variante des Laplace-Filter wie folgt:

$$v_i' = v_i + \frac{\lambda}{E_i} \sum_{j=1}^m \frac{u_j - v_i}{\|\vec{e}_{ij}\|} \quad \text{mit } E_i = \sum_{j=1}^m \frac{1}{\|\vec{e}_{ij}\|} \quad (4.2)$$

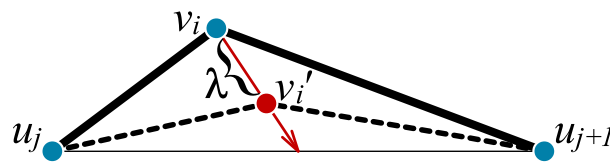
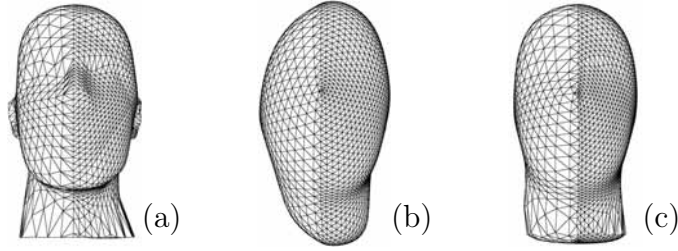


Abbildung 4.9: Laplace-Glättung eines Oberflächenpunktes.



**Abbildung 4.10:** Anwendung des Laplace-Filters auf ein Oberflächenmodell: (a) Oberfläche mit unregelmäßiger Auflösung, (b) verzerrte Oberfläche nach einfacher Laplace-Filterung, während (c) nach gleichstarker *Scale-Dependent* Laplace-Glättung kaum Verzerrung auftritt. (modifiziert aus: [Desbrun et al. 1999])

Über den *Scale-Dependent* Laplace-Filter kann, wie in Abbildung 4.10 (c) dargestellt, die tangential Driftbewegung und eine Verzerrung der Oberfläche verringert werden. Eine tangential Bewegung der Eckpunkte, z.B. in ebenen Regionen, sowie das Schrumpfen der geglätteten Objekte kann dadurch nicht verhindert werden.

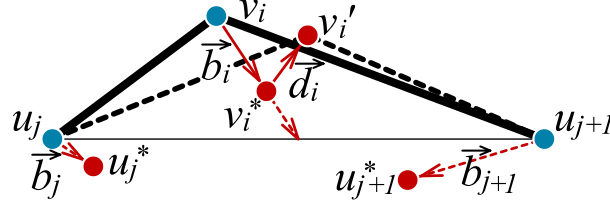
### 4.3.2 Laplace+ $\mathcal{HC}$ -Glättung

Der Laplace+ $\mathcal{HC}^1$  Algorithmus von Vollmer et al. [1999] reduziert das Schrumpfen der Oberflächennetze bei der Laplace-Filterung. Die Idee bei der  $\mathcal{HC}$ -Korrektur besteht dabei darin, die bei der Laplace-Filterung verschobenen Punkte in Richtung der ursprünglichen Oberfläche zurück zu verschieben. Die Richtung und der Betrag für dieses Zurückverschieben wird dabei von der originalen und vorherigen Position eines Knotens sowie aus der gemittelten Verschiebung seiner direkten Nachbarn abgeleitet. Ein alternierendes Ausführen von Laplace-Glättung und  $\mathcal{HC}$ -Korrektur bewirkt dabei eine Glättung des Oberflächenmodells bei gleichzeitiger Reduktion des Schrumpfens.

Da der Laplace+ $\mathcal{HC}$ -Filter auf dem Laplace-Filter aufbaut, besitzt der Laplace+ $\mathcal{HC}$ -Filter die gleichen zwei Grundparameter (Wichtungsfaktor  $\lambda$  und Anzahl der Iterationen  $\iota$ ) plus zwei zusätzliche Parameter für die  $\mathcal{HC}$ -Korrektur. Der erste der beiden zusätzlichen Parameter ( $\alpha$ ) gibt an, wie stark in Richtung der originalen ( $o_i$ ) bzw. letzten Position ( $v_i$ ) des Knotens zurück verschoben wird, und der zweite ( $\beta$ ), wie stark dabei der Verschiebungsvektor  $\vec{b}_i$  des aktuellen Knotens und der der Nachbarn  $\vec{b}_j$  eingehen. Formel 4.3 stellt dar, wie erst die mittels Laplace geglättete Position  $v_i^*$  für jeden Knoten  $v_i$  berechnet und dann unter der gegebenen Wichtung durch  $\alpha$  und  $\beta$  die  $\mathcal{HC}$ -korrigierte Position  $v_i'$  berechnet wird. Abbildung 4.11 stellt den Zusammenhang aus Formel 4.3 grafisch dar.

$$\begin{aligned}
 v_i^* &= v_i + \frac{\lambda}{m} \sum_{j=1}^m (u_j - v_i) & \forall u_j \in \mathcal{U}_{v_i}^1 \quad m = \|\mathcal{U}_{v_i}^1\| & \quad (4.3) \\
 \vec{b}_i &= v_i^* - (\alpha o_i + (1 - \alpha)v_i) \\
 \vec{d}_i &= -\beta \vec{b}_i - \frac{1 - \beta}{m} \sum_{j=1}^m \vec{b}_j \\
 v_i' &= v_i^* + \vec{d}_i
 \end{aligned}$$

<sup>1</sup>„ $\mathcal{HC}$  stands for Humphrey’s Classes and has no deeper meaning.“ [Vollmer et al. 1999]

Abbildung 4.11: Laplace+ $\mathcal{HC}$ -Glättung eines Oberflächenpunktes.

### 4.3.3 Taubin's $\lambda$ - $\mu$ -Glättung

Der  $\lambda$ - $\mu$ -Filter (auch als LowPass-Filter bezeichnet) repräsentiert eine Kombination aus einem positiv und einem negativ gewichteten Laplace-Filter, die alternierend ausgeführt werden [Taubin 1995b]. Dies entspricht, ähnlich dem Laplace+ $\mathcal{HC}$ -Filter, einem Glättungsschritt gefolgt von einem Rückverschiebungsschritt, der zu einem besseren Erhalt des Volumens im Gegensatz zur einfachen Laplace-Glättung führt.

Die geglättete Position  $v_i'$  eines Oberflächenpunktes ergibt sich dabei aus seiner Laplace gefilterten Position  $v_i^*$  und einer negativ gewichteten Laplace-Filterung von  $v_i^*$  und den Laplace gefilterten Positionen seiner Nachbarn  $u_j^*$ :

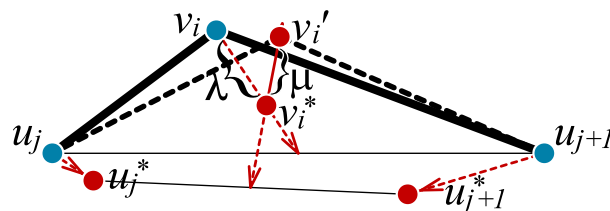
$$\begin{aligned} v_i^* &= v_i + \frac{\lambda}{m} \sum_{j=1}^m (u_j - v_i) \quad \forall u_j \in \mathcal{U}_{v_i}^1 \quad m = \|\mathcal{U}_{v_i}^1\| \\ v_i' &= v_i^* + \frac{\mu}{m} \sum_{j=1}^m (u_j^* - v_i^*) \end{aligned} \quad (4.4)$$

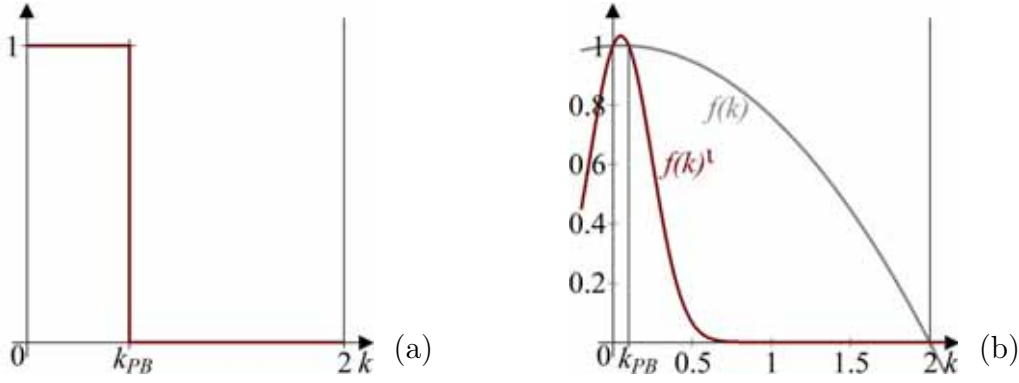
Abbildung 4.12 veranschaulicht das Vorgehen des  $\lambda$ - $\mu$ -Filters. Im zweiten negativ gewichteten Laplace-Schritt wird der für den einfachen Laplace-Schritt verwendete Wichtungsfaktor  $\lambda$  durch einen negativen Faktor  $\mu$  ersetzt. Hierbei wird  $\mu$  so gewählt, dass  $\|\mu\|$  leicht größer als  $\lambda$  ist. Um  $\lambda$  und  $\mu$  geeignet zu bestimmen, hilft es, die Filterfunktion der gewünschten Verstärkung bzw. Abschwächung bestimmter Frequenzen im Oberflächennetz zu betrachten.

Zur Glättung von Oberflächennetzen sollte dabei zur Annäherung an einen idealen Lowpass-Filter (Abb. 4.13 (a)) eine Filterfunktion  $f(k)$  so gewählt sein, dass bei  $\iota$ -maliger iterativer Anwendung  $f(k)^\iota$  für hohe Frequenzen  $f(k_i)^\iota \approx 0$  und für niedrige Frequenzen  $f(k_i)^\iota \approx 1$  sei. In Abhängigkeit von  $\lambda$  und  $\mu$  wird dazu von Taubin [1995b, 2000] die folgende Filterfunktion (siehe auch Abb. 4.13) empfohlen:

$$f(k) = (1 - \lambda k)(1 - \mu k) \quad \forall k \in [0; 2] \quad (4.5)$$

Mit  $f(0) = 1$  und der beschriebenen Beziehung  $\lambda + \mu < 0$  existiert ein positiver Wert für  $k$ , welcher als  $k_{PB}$  (Pass-Band-Frequenz) bezeichnet wird, für den gilt  $f(k_{PB}) = 1$  (siehe

Abbildung 4.12: Taubin's  $\lambda$ - $\mu$ -Glättung eines Oberflächenpunktes.



**Abbildung 4.13:** Idealer Lowpass-Filter mit Pass-Band-Frequenz  $k_{PB}$  (a) und angenäherte Filterfunktion (b) für Taubin's  $\lambda$ - $\mu$ -Glättung mit  $f(k) = (1 - \lambda k)(1 - \mu k)$  und  $f(k)^\iota$  mit  $\iota = 50$  und  $k_{PB} = 0,1$  (b).

Abb. 4.13). Der Wert von  $k_{PB}$  ergibt sich damit durch Einsetzen von  $f(k_{PB}) = 1$  in Formel 4.5 aus:

$$k_{PB} = \frac{1}{\lambda} + \frac{1}{\mu} > 0 \quad (4.6)$$

Der Graph der Funktion zeigt, wie in Abbildung 4.13 (b) dargestellt, den typischen Verlauf eines Lowpass-Filters, bei dem  $f(k) \approx 1$  für  $0 \leq k \leq k_{PB}$  und  $f(k)$  für steigendes  $k$  stark abnimmt. Dabei nähert sich  $f(k)^\iota$  bei steigender Iterationszahl  $\iota$  einem idealen Lowpass- bzw. Pass-Band Filter mit seiner Grenze an  $k_{PB}$  an.

Durch adäquates Wählen von  $k_{PB}$ ,  $\lambda$  und  $\mu$  können dann stabile und schnelle Filter erzeugt werden [Taubin et al. 1996]. Taubin [1995b]; Taubin et al. [1996]; Taubin [2000] verwenden dabei im allgemeinen  $k_{PB} = 0,1$ ,  $\lambda = 0,6307$  und das sich über Formel 4.6 ergebende  $\mu = -0,6732$ . Ein hoher  $\lambda$ -Wert sorgt dabei für eine schnelle Annäherung der Filterfunktion an einen idealen Tiefpassfilter. Eine starke Glättung kann darüber hinaus durch einen niedrigen  $k_{PB}$ -Wert<sup>2</sup> erreicht werden.

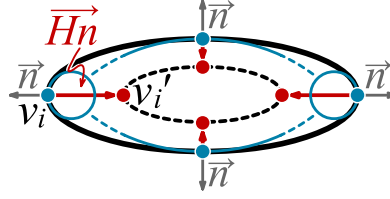
#### 4.3.4 Mean Curvature Flow - Glättung

Um die Glättung unabhängig von der Triangulierung der Oberfläche zu machen, wird von Desbrun et al. [1999] die Glättung mittels *Mean Curvature Flow* eingeführt. Während bei den auf dem Laplace-Filter beruhenden Verfahren Eigenschaften der Triangulierung des Netzes mit geglättet werden (siehe Abb. 4.10 (b)), wird bei der *Mean Curvature Flow* Glättung die Oberfläche nur in Richtung der lokalen Normale mit der Geschwindigkeit der *Mittleren Krümmung*  $H$  (siehe Abschnitt 3.5) verschoben.

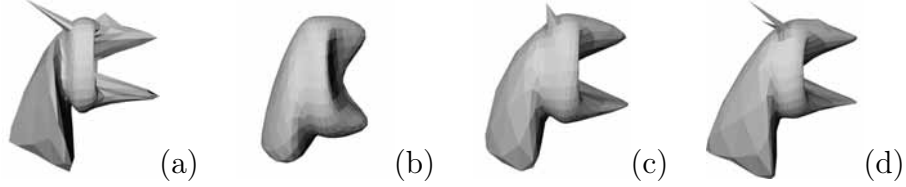
Über die Definition der *Mean Curvature Flow* Normale  $\vec{Hn} = \frac{\nabla A}{2A}$  kann die folgende diskrete Repräsentation (vergleiche auch Abschnitt 3.5) gefunden werden [Desbrun et al. 1999]:

$$\vec{Hn} = \frac{1}{4A} \sum_{j=1}^m (\cot \alpha_{ij} + \cot \beta_{ij})(u_j - v_i) \quad \forall u_j \in \mathcal{U}_{v_i}^1 \quad m = \|\mathcal{U}_{v_i}^1\| \quad (4.7)$$

<sup>2</sup>Kleiner  $k_{PB}$ -Wert  $\Rightarrow$  starke Glättung, da alle höheren Frequenzen (die einem Wert über  $k_{PB}$  entsprechen) reduziert werden.



**Abbildung 4.14:** *Mean Curvature Flow* Glättung eines Oberflächenpunktes.



**Abbildung 4.15:** Starke Glättung eines Modells: (a) Original, (b) Standard-Laplace-Glättung, (c) Scale-Dependent Laplace-Glättung und (d) *Mean Curvature Flow*-Glättung mit bestem Formerhalt. [Desbrun et al. 1999]

Formel 4.7 eignet sich dabei für die implizite Integration. Eine Beschreibung von  $\overrightarrow{Hn}$  für die explizite Integration kann dabei über die Normalisierung erhalten werden (explizite vs. implizite Integration siehe Abschnitt 3.4.2):

$$\overrightarrow{Hn}_{Normalized} = \frac{1}{G} \sum_{j=1}^m (\cot \alpha_{ij} + \cot \beta_{ij})(u_j - v_i) \quad \forall u_j \in \mathcal{U}_{v_i}^1 \quad m = \|\mathcal{U}_{v_i}^1\| \quad (4.8)$$

$$\text{mit } G = \sum_{j=1}^m (\cot \alpha_{ij} + \cot \beta_{ij})$$

Die mittels *Mean Curvature Flow* geglättete Position  $v_i'$  des Oberflächenpunktes  $v_i$  (siehe Abb. 4.14) ergibt sich unter Einbeziehung des Wichtungsfaktors  $\lambda$  über:

$$\begin{aligned} v_i' &= v_i + \lambda \overrightarrow{Hn}_{Normalized} \\ &= v_i + \frac{\lambda}{G} \sum_{j=1}^m (\cot \alpha_{ij} + \cot \beta_{ij})(u_j - v_i) \end{aligned} \quad (4.9)$$

Wie in Abbildung 4.15 dargestellt, kann im Vergleich zur Standard-Laplace- und Scale-Dependent Laplace-Glättung die Form des Objektes besser erhalten werden. Jedoch ist die Glättung mittels *Mean Curvature Flow* nicht volumenerhaltend (auch wenn dies durch Skalieren der Ergebnisse in [Desbrun et al. 1999] nahe gelegt wird) und führt ohne geeignete Korrektur zum Schrumpfen der Objekte (siehe Abb. 4.14).

### 4.3.5 Mean- und Median-Glättung der Normalen

Ein anderer Ansatz zum Glätten von Oberflächennetzen glättet die Normalen der Oberfläche und passt dann das Oberflächennetz an die geglätteten Normalen an [Yagou et al. 2002a, b]. Hierzu wird im ersten Schritt die geglättete Normale  $\overrightarrow{m}_i$  für jedes Dreieck  $f_i$  des Oberflächennetzes aus den Normalen  $\overrightarrow{n}_j$  der direkt benachbarten Dreiecke  $f_j$  berechnet. Im zweiten



Schritt wird dann jeder Eckpunkt  $v_i$  des Oberflächennetzes entsprechend den geglätteten Normalen wie folgt angepasst:

$$v_i' = v_i + \frac{\lambda}{\sum_{j=1}^m A(f_j)} \sum_{j=1}^m A(f_j) \vec{w}_j \quad \forall f_j \in \mathcal{U}_{v_i}^1 \quad m = \|\mathcal{U}_{v_i}^1\| \quad (4.10)$$

mit  $\vec{w}_j = \left( \overrightarrow{(c_j - v_i)} \cdot \vec{m}_j \right) \vec{m}_j$

wobei die Summen über alle Dreiecke  $f_j$  berechnet werden, die in  $v_i$  zusammenlaufen,  $A(f_j)$  der Flächeninhalt des entsprechenden Dreiecks ist und  $\vec{w}_j$  die Projektion der Strecke von  $v_i$  zum Zentrum  $c_j$  jedes benachbarten Dreiecks  $f_j$  auf dessen geglättete Normale  $\vec{m}_j$  darstellt. Entsprechend ergibt sich ein an die geglätteten Normalen angepasstes Oberflächenmodell. Abbildung 4.16 illustriert die beschriebenen Zusammenhänge.

Um eine geglättete Normale  $\vec{m}$  zu berechnen, kann der Durchschnitt der Nachbarnormalen (siehe Abb. 4.16) über Mittelung und Normalisierung bestimmt werden:

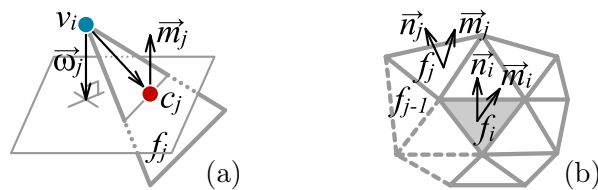
$$\begin{aligned} \vec{m}_i^* &= \frac{1}{\sum_{j=1}^m A(f_j)} \sum_{j=1}^m A(f_j) \vec{n}_j \quad \forall f_j \in \mathcal{U}_{v_i}^1 \quad m = \|\mathcal{U}_{v_i}^1\| \\ \vec{m}_i &= \frac{\vec{m}_i^*}{\|\vec{m}_i^*\|} \end{aligned} \quad (4.11)$$

Mit dieser Mittelung erhält man einen *Mean*-Filter zur Glättung von Oberflächenmodellen.

Über eine geeignete Sortierung der Nachbarnormalen kann auch eine *Median*-Filterung realisiert werden. Yagou et al. [2002a] verwenden für die Sortierung den Winkel  $\phi_j = \angle(\vec{n}_i, \vec{n}_j)$  zwischen der Normale  $\vec{n}_i$  des aktuellen Dreiecks  $f_i$  und der Normale  $\vec{n}_j$  jedes benachbarten Dreiecks  $f_j$ . Entsprechend der klassischen Median-Filterung wird die Normale in der Mitte der nach  $\phi$  sortierten Normalen als „geglättete“ Normale  $\vec{m}$  für das aktuelle Dreieck verwendet. Entsprechend kann diese Methode als *Angle Median*-Filterung bezeichnet werden. In [Yagou et al. 2002b] wird zusätzlich eine *Curvature Median*-Filterung diskutiert, wobei die richtungsabhängige Krümmung (*directional curvature*) hier über den Winkel  $\phi$  und den Abstand der Zentren der Dreiecke als  $k_i = \frac{\phi_j}{\|c_j - c_i\|}$  approximiert wird.

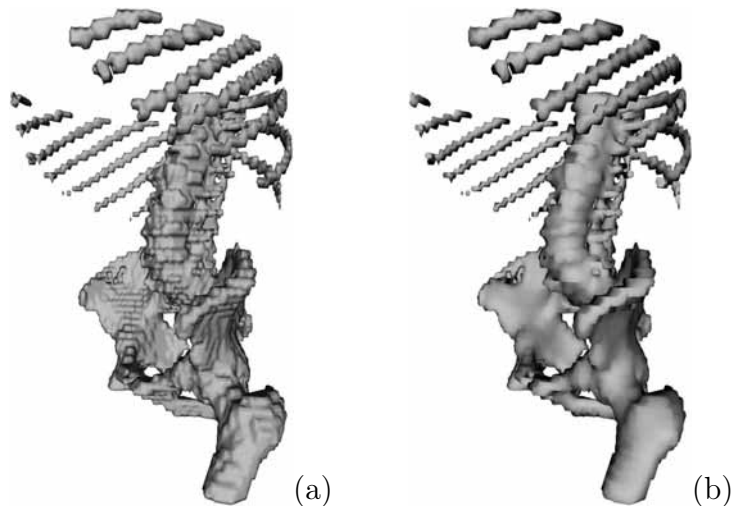
Der Vorteil der Normalen-Glättung mit *Mean* und *Median*-Filter liegt nach Yagou et al. [2002b] darin, dass diese kein Schrumpfen verursachen. Diese Aussage ist aber vor dem Hintergrund, dass Yagou et al. [2002a, b] ihre Ergebnisobjekte auf das richtige Volumen skalieren, zu relativieren.

Ein Vorteil der *Median*-Filterung gegenüber der *Mean*-Filterung und allen anderen bisher beschriebenen Filtern ist der Erhalt von markanten Merkmalen und Kanten der Oberflächennetze. Die *Angle Median*-Filterung erhält dabei im Vergleich zur *Curvature Median*-Filterung besser scharfe Kanten [Yagou et al. 2002b]. Yagou et al. [2003] zeigen in einer



**Abbildung 4.16:** *Mean*- und *Median*-Glättung nach Yagou et al. [2002b]: (a) visuelle Repräsentation von Formel 4.10 für ein Nachbardreieck  $f_j$ , (b) Nachbarschaftsbeziehung für die Berechnung der mittleren Normalen für Formel 4.11.





**Abbildung 4.17:** Rein visuelle Glättung der Normalen: (a) Originalmodell mit Normalen senkrecht auf den Flächenelementen, (b) Modell nach fünf Iterationen *Mean*-Filterung der Normalen, welches geringes Rauschen visuell gut geglättet aber Block- und Terrassenartefakte nicht.

erweiterten Variante der Verfahren, dass sich mit der gewichteten Kombination aus *Median*- und *Mean*-Filterung die Stärke der zu erhaltenden Kanten regeln lässt.

**Visuelle Glättung:** Eine vereinfachte verwandte Variante ist die reine visuelle Glättung von Oberflächennetzen durch Glättung der Eckpunkt- bzw. Flächennormalen. Dabei werden die Normalen wie zuvor beschrieben gemittelt. Das Oberflächennetz wird jedoch nicht an die gemittelten Normalen angepasst, sondern diese werden einfach als neue Normalen für die Flächen verwendet. Abbildung 4.17 zeigt ein Modell mit entsprechend geglätteten Normalen. Visuell ist damit bei verrauschten Oberflächen schnell eine signifikante Glättung ohne Veränderung der Geometrie möglich. Bei Block- und Terrassenartefakten oder auch starkem Rauschen sind diese Diskontinuitäten der Geometrie aber vor allem an den Rändern der Objekte bzw. an der Objektsilhouette unverändert stark wahrzunehmen. Zusätzlich kann es zu sehr unnatürlich beleuchteten Regionen und damit zu neuen Beleuchtungsartefakten führen, die vor allem beim interaktiven Umgang mit den Modellen unangenehm auffallen (siehe z.B. Rippen in Abb. 4.17). Entsprechend ist die rein visuelle Glättung zur Reduktion von Block- und Terrassenartefakten ungeeignet. Zur Verbesserung der visuellen Ergebnisse bei gering verrauschten Modellen kann sie jedoch sehr effektiv sein.

#### 4.3.6 Merkmalserhaltende Glättung

Der allgemeine Ansatz zur merkmalerhaltenden (*feature-preserving*) Glättung stellt die *Anisotrope Diffusion* dar. Hierbei wird die Diffusion (Glättung) über zu erhaltende Merkmale unterbunden, wodurch sich eine ungleichmäßig ausgebreitete (anisotrope) Glättung in unterschiedliche Raumrichtungen ergibt und scharfe Kanten nicht bzw. nur in Richtung der Kante geglättet werden [Hildebrandt und Polthier 2004].

Als übliches Maß für die Detektion von markanten Merkmalen bzw. scharfen, zu erhaltenden Kanten werden die Hauptkrümmungsrichtungen ( $\vec{k}_1$  und  $\vec{k}_2$  siehe Abschnitt 3.5) herangezogen. Anhand dieser wird bevorzugt in Richtung der minimalen Krümmung geglättet und die maximale Krümmung erhalten (siehe zum Beispiel [Clarenz et al. 2000; Bajaj und Xu 2003; Chen et al. 2004]). Die Standard-Form für die iterative Integration aus Formel 3.1

(S. 32) erweitert sich dabei um eine Wichtungsfunktion  $\omega(k_1, k_2)$ , die von den Hauptkrümmungen  $k_1$  und  $k_2$  abhängt:

$$\mathcal{M}^{t+1} = \mathcal{M}^t + \lambda \omega(k_1, k_2) \nabla \mathcal{M}^t \quad (4.12)$$

Die Gewichte von  $\omega(k_1, k_2)$  werden dabei fallbasiert für Oberflächenpunkte auf sogenannten Ecken, Kanten und einheitlichen Flächen gewählt, welche über Betrachtung von  $k_1$  und  $k_2$  in Relation zu einem gegebenen Schwellwert  $\kappa$  detektiert werden können. Chen et al. [2004] verwenden z.B. folgende Funktion (siehe aber auch [Meyer et al. 2002] für eine ähnliche):

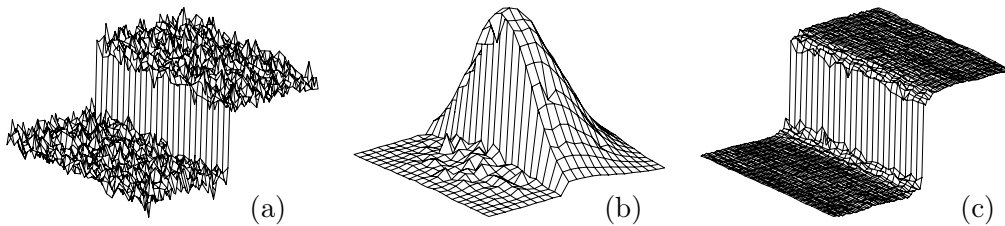
$$\omega_\kappa(k_1, k_2) = \begin{cases} 1 & \text{if } |k_1| \leq \kappa \text{ und } |k_2| \leq \kappa \\ 0 & \text{if } |k_1| \geq |k_2| \geq 2\kappa \\ e^{-(k_2/k_1-1)^2} - e^{-1} & \text{if } |k_1| > \kappa \text{ und } |k_2| \leq \kappa \\ e^{-((2\kappa-k_2)/k_1-1)^2} - e^{-1} & \text{if } |k_1| > \kappa \text{ und } \kappa < |k_2| \leq 2\kappa \end{cases} \quad (4.13)$$

mit  $|k_1| \geq |k_2|$

Entsprechend Formel 4.13 werden Oberflächenbereiche mit einer Krümmung kleiner dem gegebenen Schwellwert geglättet (einheitliche Fläche). Regionen mit einer Krümmung über dem Zweifachen des Schwellwertes (scharfe Ecken) sowie Regionen, bei denen die minimale Krümmung 0 ist (scharfe Kanten), werden von der Glättung ausgeschlossen. In Regionen, die weder eine scharfe Ecke oder Kante, noch eine einheitliche Fläche sind, findet eine sanft ansteigende Glättung von eher scharfkantigen Regionen zu eher ebenen statt. Mit einer geeigneten Wahl des Schwellwertes  $\kappa$  können entsprechend Kanten und Ecken in den Modellen erhalten werden.

Erweiterte Verfahren verwenden nicht die einfache Krümmung in einen Oberflächenpunkt, um die Glättung zu steuern, sondern betrachten die Änderung der Krümmung in einer größeren Umgebung um diesen Punkt. Lee et al. [2005] definiert in diesem Zusammenhang die absolute Differenz der Gauss-gewichteten *Mittleren Krümmung* in einer Nachbarschaft um einen Oberflächenpunkt als „*Mesh Saliency*“, welche kürzlich von Zhihong et al. [2006] zur merkmalerhaltenden Glättung genutzt wurde.

Einen weiteren Ansatz zur merkmalerhaltenden Glättung stellen „*Bilateral Mesh Filtering*“ Verfahren dar, welche aus der Bildverarbeitung abgeleitet wurden [Tomasi und Manduchi 1998; Jones et al. 2003; Fleishman et al. 2003]. Hierbei werden anisotrope Gewichte für benachbarte Pixel im Bild bzw. Eckpunkte auf Oberflächennetzen entsprechend ihrer Entfernung und Ähnlichkeit gewählt (Abb. 4.18). Einen ähnlichen Ansatz stellt „*Smoothing by Example*“ von Yoshizawa et al. [2006] dar, bei dem über ein Ähnlichkeitsmaß der Nachbarschaften aller Eckpunkte eines Netzes deren Gewichtung für die Glättung benachbarter Punkte bestimmt wird.



**Abbildung 4.18:** *Bilateral Filtering*: (a) verrauschte Eingangsdaten, (b) auf der Ähnlichkeitsberechnung basierende Gewichte für die Nachbarschaft um einen Punkt auf dem oberen Rand der Kante aus (a) und (c) Ergebnis nach *Bilateral Filtering* von (a). [Tomasi und Manduchi 1998]

Einfachere Verfahren zur Erhaltung von markanten Merkmalen betrachten den Winkel zwischen den Normalen benachbarter Flächen. Ist der Winkel über einem gegebenen Schwellwert, so werden Nachbarpunkte über diese Flächen hinweg nicht zur Glättung des aktuellen Oberflächenpunktes hinzugezogen. Entsprechende Varianten des Laplace- und  $\lambda$ - $\mu$ -Filters finden im VTK im *vtkSmoothPolyDataFilter* bzw. *vtkWindowedSincPolyDataFilter* Anwendung [Kitware Inc. 2007].

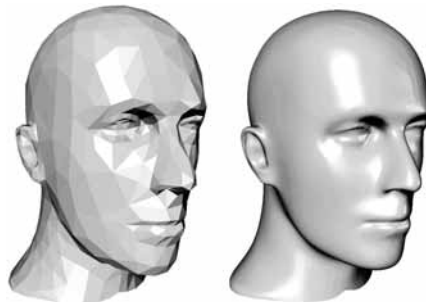
Grundlegend stellen merkmalerhaltende Glättungsverfahren unabhängig von der gewählten Methode zur Bestimmung individueller Wichtungen der Nachbarpunkte nur Varianten der zuvor beschriebenen Glättungsverfahren mit ungleichmäßig verteilten Wichtungen dar. Alle Verfahren, die dabei auf dem Mitteln bzw. anisotrop gewichtetem Mitteln von Nachbarpunkten beruhen, führen zum Schrumpfen der geglätteten Objekte [Fleishman et al. 2003]. Das Schrumpfen fällt jedoch, durch die vermiedene Glättung von markanten Merkmalen, meist geringer aus. Um das Schrumpfen zusätzlich zu verringern, kann jedoch auch hier auf ein Vorgehen wie in Taubin's  $\lambda$ - $\mu$ -Verfahren (Abschnitt 4.3.3) oder auf das Skalieren auf das Originalvolumen von Desbrun et al. [1999] zurückgegriffen werden [Fleishman et al. 2003].

Für die Glättung bzw. Reduktion von Block und Terrassenartefakten aus binären Volumendaten eignen sich die Verfahren jedoch nicht. Da die auftretenden Artefakte Diskontinuitäten mit signifikanten Krümmungswerten darstellen, würden diese als markante Merkmale identifiziert und erhalten werden. Entsprechend bieten bei der Reduktion von Artefakten aus binären Volumendaten anisotrope Glättungsverfahren gegenüber isotropen Verfahren keinen zusätzlichen Genauigkeits- und Glättungsgewinn.

### 4.3.7 Subdivision Surfaces

Neben der Glättung von Oberflächenmodellen bzw. Polygonnetzen durch das Verändern/Glätten der Position vorhandener Eckpunkte ist auch eine Glättung durch das Hinzufügen neuer Eckpunkte (*Subdivision Surfaces*) möglich.

Das Hinzufügen neuer Eckpunkte über *Subdivision*-Verfahren (siehe z.B. [Kobbelt 1997; Pakdel und Samavati 2007]) wird verwendet, um ein grobes Netz durch zusätzliche Eckpunkte und Flächen feiner aufzulösen und gleichzeitig glatter repräsentieren zu können (siehe Abb. 4.19). Zur Glättung und Reduktion von Artefakten in bereits sehr detaillierten Polygonnetzen, wie sie in dieser Arbeit behandelt werden, führt die weitere Verfeinerung des Oberflächennetzes zu einer übermäßigen Dreiecksanzahl und zu keiner Reduktion von Block- und Terrassenartefakten.



**Abbildung 4.19:** Glatte Repräsentation von grob aufgelösten Oberflächen über *Subdivision Surfaces*: (links) originales Oberflächennetz, (rechts) stark verfeinertes Netz. (Rendering beider Modelle mittels *flat shading*.) [Kobbelt 1997]

### 4.3.8 Vergleich von Glättungsverfahren

Die Vielfalt an Glättungsverfahren und ihrer Parametrisierungsmöglichkeiten erfordern gezielte Vergleiche der Techniken. Bisherige Vergleichsarbeiten gehen dabei, durch die Hauptanwendung zur Rauschreduktion, von idealen Modellen aus, die künstlich verrauscht und dann möglichst gut wieder geglättet werden (siehe u.a. [Vollmer et al. 1999; Belyaev und Ohtake 2003; Yagou et al. 2003; Chen et al. 2004]). Die meisten Arbeiten präsentieren dabei ausschließlich die mit dem jeweiligen Verfahren bestmöglichen Ergebnisse für das künstlich verrauschte Objekt [Desbrun et al. 1999; Vollmer et al. 1999; Belyaev und Ohtake 2003; Yagou et al. 2002b]. Um zu den bestmöglichen Ergebnissen zu kommen, werden für das jeweilige Glättungsverfahren und das jeweils betrachtete Objekt empirisch geeignete Parametereinstellungen vorgenommen. Wie jedoch entsprechend gute Parametrisierungen des jeweiligen Verfahrens für andere Objekte zu finden sind, ist unklar.

Als Bewertungskriterium für die Güte der Rauschreduktion durch das jeweilige Glättungsverfahren wird meist der Abstand des Glättungsergebnisses zum originalen unverrauschten Objekt verwendet. Belyaev und Ohtake [2003] berechnen hierzu zu jedem Punkt des geglätteten Netzes den am nächsten gelegenen Punkt auf dem Originalmodell. Der euklidische Abstand dieser Punkte und die Abweichung der Normalen an den Punkten werden dann als Fehlermaße angegeben. Yoshizawa et al. [2006] schlagen in Anlehnung an die Subtraktionsbilder in der Bildverarbeitung als Fehlermaß den euklidischen Abstand  $|v_i - v_i'|$  zwischen verrauschter  $v_i$  und geglätteter Position  $v_i'$  vor. Dieses Vorgehen ist auch dann geeignet, wenn kein ideales Originalmodell vorhanden ist und gibt darüber hinaus deutlicher Aufschluss darüber, welche Details der Oberfläche geglättet wurden (siehe Abb. 4.20).

Andere Arbeiten [Taubin 1995b; Desbrun et al. 1999; Bajaj und Xu 2003; Fleishman et al. 2003] beschränken sich darauf, rein visuelle Ergebnisse zu präsentieren, ohne diese quantitativ zu untersuchen. Algorithmenvergleichende Arbeiten wie [Belyaev und Ohtake 2003] und [Yagou et al. 2002b] vergleichen in der Regel nur verschiedene Algorithmen in Bezug auf die Änderung eines Fehlermaßes zum „perfekten“ Originalmodell bei festen Parametern und steigender Iterationsanzahl. Aussagen zur geeigneten Parameterwahl sind dabei meist vage und beschränken sich auf empirische Werte für die jeweils betrachteten Objekte.

Keine der vergleichenden Studien betrachtet medizinische Modelle aus radiologischen Bilddaten. Nur Taubin [1995b] präsentiert für seinen  $\lambda$ - $\mu$ -Filter Glättungsergebnisse medizinischer Modelle. Da die Reduktion von Block- und Terrassenartefakten sich im Gegensatz zur Reduktion von Rauschen und dem Erhalt von Merkmalen stark unterscheidet, können keine Aussagen aus bisherigen Vergleichsstudien einfach auf die Glättung von medizinischen Modellen übertragen werden. Aussagen und Arbeiten zur volumen- und formerhaltenden Glättung sind hingegen relevant für die Glättung medizinischer Modelle. Diese wurden bis-



**Abbildung 4.20:** Positionsunterschiede nach Glättung: (a) verrauschtes und (b) mittels *Bilateral Filtering* geglättetes Modell (Färbung nach *Mittlerer Krümmung*) sowie (c) Positionsunterschiede der Oberflächenpunkte zwischen verrauschtem und geglättetem Oberflächenmodell. [Yoshizawa et al. 2006]

her aber meist auch nur im Zusammenhang mit Merkmalserhaltung und nicht-medizinischen Modellen erarbeitet. Entsprechend ist die Eignung der unterschiedlichsten Algorithmen für die Glättung von Oberflächenmodellen für medizinische Oberflächenmodelle unklar.

## 4.4 Zusammenfassung und Diskussion

Zur Reduktion von Artefakten in Oberflächennetzen aus binären Volumendaten dienen grundlegend drei unterschiedliche Strategien: (a) Filterung und Glättung der binären Volumendaten (Bildebene), (b) Reduktion der Artefakte bei der Oberflächenextraktion (Bild- und Oberflächennetzebene) sowie (c) die Filterung und Glättung der extrahierten Oberflächen (Oberflächennetzebene).

Existierende Verfahren zur Glättung der binären Volumendaten eignen sich dabei vornehmlich für die direkte Volumenvisualisierung. Dabei erzielen zum Beispiel *fuzzy boundaries* eine ausreichende Glättung. Für die anschließende Extraktion von Oberflächen reicht die Glättung von Terrassenartefakten jedoch nicht aus. Grundsätzlich leiden die bildbasierten Ansätze unter einer hohen Laufzeit, da meist viele Voxelwerte pro Iteration berechnet und verändert werden müssen. Zusätzlich leidet die Qualität von extrahierten Oberflächen darunter, dass während der Oberflächenextraktion erneut Artefakte auftreten können.

Existierende Oberflächenextraktionsverfahren, die bei Volumendaten mit annähernd kontinuierlichem Wertebereich die Oberflächenqualität in Bezug auf die Glattheit verbessern können (z.B. trilineare und Sub-Voxel Interpolation), vermögen jedoch in binären Volumendaten nichts auszurichten. Gängige Verfahren, die eine Artefaktreduktion erlauben, approximieren die zu extrahierende Oberfläche mit Hilfe von impliziten Oberflächen oder verbinden eine spezialisierte Oberflächenextraktionsmethode mit einer entsprechend abgestimmten und im Anschluss ausgeführten Oberflächenglättung.

Zur Artefaktreduktion und Glättung von Oberflächenmodellen stehen zahlreiche Verfahren zur Verfügung. Verfahren, die markante Merkmale und Kanten in den Modellen erhalten bzw. zu erhalten versuchen, sind dabei für die Reduktion von Block und Terrassenartefakten nicht geeignet. So bieten sich hauptsächlich isotrope Glättungsstrategien an. Entsprechend finden solche Verfahren verbreitet Anwendung bei der Erzeugung artefaktfreier bzw. artefaktreduzierter Oberflächen aus binären Volumendaten.

Da das Hauptaugenmerk der Forschung bei Oberflächenglättungsverfahren auf der Reduktion von Rauschen und dem Erhalt von Kanten in eher technischen Modellen liegt, ist die konkrete Eignung der Verfahren zur Reduktion von Artefakten in Oberflächennetzen aus binär segmentierten radiologischen Bilddaten jedoch unklar. Speziell der für beliebige Objekte mögliche Volumenerhalt, die Genauigkeit und die zu erreichende Artefaktreduktion sind dabei grundlegend unbekannt.



# Evaluation von Glättungsverfahren für medizinische Oberflächenmodelle

---

Nach Betrachtung der möglichen Strategien zur Artefaktreduktion in Oberflächenmodellen aus binären Volumendaten werden in diesem Kapitel Glättungsverfahren für Oberflächenmodelle anatomischer Strukturen näher untersucht.

Unter Berücksichtigung spezifischer Anforderungen und Schwierigkeiten im Gegensatz zur Rauschreduktion in eher technischen Modellen ist bisher unklar, welche Algorithmen zur Glättung von medizinischen Oberflächenmodellen geeignet sind. Neben der Wahl eines Glättungsverfahrens müssen auch geeignete Parameter für die Verfahren bestimmt werden. Die erreichbare Glättung sowie auftretende Fehlerwerte sind dabei schwer abzuschätzen. Entsprechend ist derzeit kein elaboriertes und zielgerichtetes Vorgehen bei der Glättung medizinischer Oberflächenmodelle, unter Berücksichtigung von Volumen-, Ausdehnungs- und Abstandserhalt möglich. Um dies zu ermöglichen, werden in diesem Kapitel isotrope Glättungsalgorithmen zur Glättung medizinischer Oberflächenmodelle untersucht. Hierzu werden zunächst die zu glättenden medizinischen Oberflächenmodelle analysiert und klassifiziert (Abschnitt 5.1). Daraufhin werden die Anforderungen an die Glättung medizinischer Oberflächenmodelle erarbeitet (Abschnitt 5.2).

Aufbauend auf Objekt- und Anforderungsanalyse können dann die verschiedenen Verfahren und Methoden zur Oberflächennetzglättung evaluiert werden (Abschnitt 5.3–5.6), um anschließend Glättungsempfehlungen ableiten zu können (Abschnitt 5.7). Zum Vergleich und zur Bewertung der Glättungsverfahren wird ein Konzept entwickelt, welches auch im Alltag der medizinischen Visualisierung eingesetzt werden kann (Abschnitt 5.4). In diesem Konzept wird vom initial extrahierten Oberflächenmodell als „Original“ ausgegangen und Vergleichs- und Bewertungsangaben in Bezug zu diesem unvollkommenen und artefaktbehafteten Original untersucht. Dieses Vorgehen orientiert sich an der medizinischen Praxis, in der nicht auf ein „perfektes“ Oberflächenmodell der patientenindividuellen Anatomie zurückgegriffen werden kann. Dadurch können die Glättungsergebnisse und Erfahrungen aus dieser Studie nicht nur dazu verwendet werden, Glättungsempfehlungen abzuleiten, sondern können auch direkt in den üblichen Visualisierungsprozess zur Qualitätssicherung integriert werden.

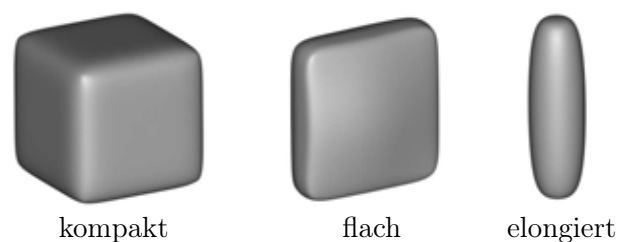
## 5.1 Objektanalyse und -klassifizierung

Dreidimensionale Oberflächenmodelle anatomischer Strukturen weisen im Normalfall kaum rechtwinklige oder scharfe gerade Kanten auf. Sie repräsentieren meist organische Strukturen mit leicht verlaufenden, oft objektcharakteristischen Furchen, Abzweigungen oder Ein- und Auswölbungen. Dies steht im Gegensatz zu den Modellen, auf die sich die kantenerhaltenden Glättungsverfahren (z.B. für CAD/CAM-Anwendungen) konzentrieren. Weiterhin weisen Oberflächenmodelle anatomischer Strukturen eine große Vielfalt unterschiedlicher Objekteigenschaften und -charakteristika auf.

Da Objekte unterschiedlicher Form sich unterschiedlich bei der Glättung verhalten und entsprechend verschiedene Glättungsverfahren auch nur für bestimmte Objekte geeignet sind,

scheint es sinnvoll, anhand maßgeblicher Objekteigenschaften Objektklassen abzuleiten, für die dann ähnliche Bedingungen gelten. Entsprechend wird in dieser Arbeit zwischen Oberflächenmodellen kompakter, flacher und elongierter Objekte unterschieden (Abb. 5.1). Darüber hinaus bietet sich eine Unterscheidung der Objekte entsprechend dem Verhältnis von Dreiecksgröße zu Featuregröße in den Oberflächenmodellen an. In diesem Zusammenhang wird nachfolgend vereinfachend von *high-res.*-Objekten gesprochen, sofern die Objektdetails/features um ein Vielfaches größer als die Dreiecke sind. Von *low-res.*-Objekten wird hingegen gesprochen, wenn die Dreiecksgröße ähnlich der Featuregröße ist (Featuregröße am Rande der Auflösungsgröße).

Im folgenden Abschnitt werden die unterschiedlichen Objektklassen näher beschrieben und charakteristische Eigenschaften und Probleme von typischen Vertretern dieser Klassen analysiert.



**Abbildung 5.1:** Klassifizierung der Oberflächenmodelle in kompakte, flache, und elongierte Objekte.

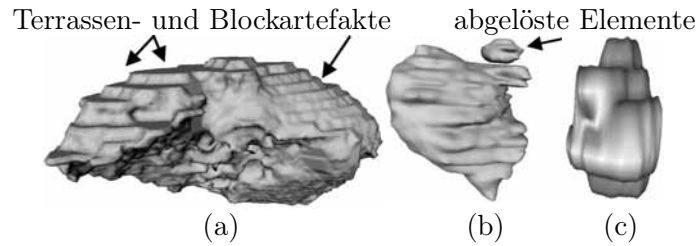
### 5.1.1 Kompakte Objekte

Kompakte Objekte zeichnen sich durch ihr großes Volumen bei geringer Oberfläche aus. Typische medizinische Beispiele sind Leber, Milz und Nieren sowie kleinere *low-res.* Strukturen, wie Lymphknoten, Zysten und einige Drüsen. Von medizinischer Relevanz bei kompakten *low-res.* Objekten (Tumore, krankhafte Lymphknoten) sind die genaue räumliche Lage, das Volumen und die Ausdehnung. Im Kontrast dazu sind bei kompakten *high-res.* Oberflächenmodellen (z.B.: von Organen) kleinere Abweichungen der Volumina und das Schrumpfen kleinerer Objektelemente tolerierbar.

Bei kompakten Objekten kommt es oft vor, dass sich zwischen zwei aufeinander folgenden Schichten im Bildstapel starke Veränderungen in der Objektgröße ergeben. An diesen Stellen kommt es dann bei der Oberflächenrekonstruktion zu Terrassenartefakten in den Oberflächenmodellen. Bedingt durch die Oberflächenextraktion sind diese Terrassen durch eine große Anzahl koplanarer Dreiecke charakterisiert, welche den Glättungsprozess erschweren können. Vor allem bei stark anisotropen Daten, wie CTs vom Hals mit einem Verhältnis von 10:1 ( $0,3 \times 0,3 \times 3\text{mm}$ ) ist dieser Effekt am stärksten.

Besonderheiten bei kompakten Objekten können abgelöste Objektteile sein. Diese können Teil der abgebildeten Strukturen (z.B. Satelliten eines Tumors) sein, aber auch Artefakte aus der Bildaufnahme und Segmentierung (siehe u.a. [Bade et al. 2006a]). Abbildung 5.2 zeigt typische kompakte anatomische Strukturen mit den beschriebenen Problemen. Kompakte *low-res.* Objekte (siehe Lymphknoten in Abb. 5.2 (c)) weisen im Gegensatz zu den kompakten *high-res.* Objekten stärkere Krümmungswerte und Unterschiede zwischen benachbarten Oberflächendreiecken auf, was bei der Glättung bereits bei geringen Iterations- und Wichtungsfaktoren zu starken Veränderungen des Oberflächenmodells führen kann.

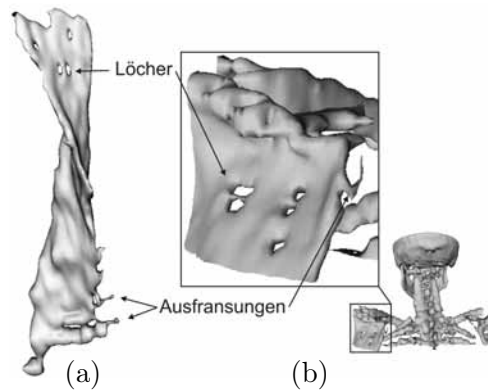




**Abbildung 5.2:** Beispiele für kompakte Objekte und ihre Besonderheiten: (a) Oberflächenmodell einer Leber mit Block- und Terrassenartefakten; (b) Tumormodell mit abgelösten Elementen (Satellit); (c) Modell eines Lymphknotens (vergrößert).

### 5.1.2 Flache Objekte

Objekte werden als flach klassifiziert, wenn ihre Ausdehnung in eine Richtung wesentlich kleiner als in die anderen Richtungen ist. Typische Beispiele sind der Halsmuskel *M. sternocleidomastoideus* und flache Knochen, wie Schulterblätter, Brustbein und Teile des Beckens. Durch die geringe Ausdehnung dieser Strukturen in eine Richtung kann es durch den Partialvolumeneffekt und die Segmentierung häufig zu Löchern, Ausfransungen und abgelösten Elementen kommen. Bedingt durch erhöhte Krümmungswerte und große Unterschiede zwischen benachbarten Oberflächendreiecken an diesen Problemstellen kann die Glättung zu starken Deformationen dieser Objektteile führen. Entsprechend allen Oberflächenmodellen aus binären Segmentierungsmasken leidet auch die Darstellung flacher Objekte unter Block- und Terrassenartefakten. Abbildung 5.3 illustriert die aufgezeigten Probleme bei flachen Strukturen.

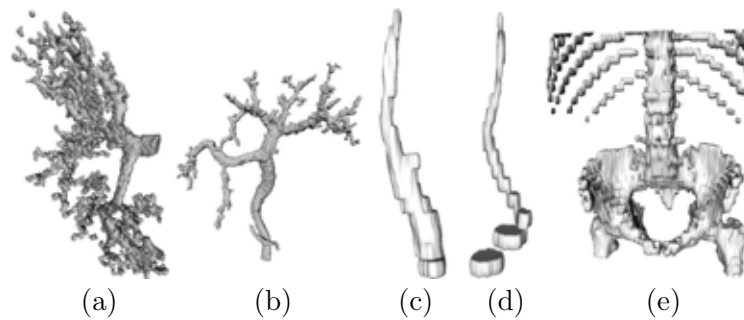


**Abbildung 5.3:** Beispiele für flache Objekte und ihre Besonderheiten: (a) Halsmuskel (*M. sternocleidomastoideus*) und (b) Schulterblatt, beide mit Löchern und Ausfransungen.

### 5.1.3 Elongierte Objekte

Strukturen, die sich hauptsächlich in eine Richtung entlang einer (gekrümmten) Achse erstrecken, werden als elongierte Objekte klassifiziert. Blutgefäße, Bronchialwege sowie einige Knochen stellen typische medizinische Beispiele für elongierte Objekte dar (siehe Abb. 5.4).

In der Regel handelt es sich in dieser Objektkategorie um kleine filigrane Strukturen wie Blutgefäße. Der durchgehende Verlauf dieser Strukturen ist dabei meist entscheidend für ihre Funktion (z.B. als Pipeline). Durch ihr üblicherweise schlechtes Verhältnis von Dreiecksgröße zu Featuregröße (*low-res.*), leidet die Darstellung elongierter Objekte stark unter Block-



**Abbildung 5.4:** Beispiele für elongierte Objekte und ihre Besonderheiten: (a) Stark verzweigter Lebergefäßast mit abgelösten Elementen, (b) Lebergefäßast mit wenigen Verzweigungen, (c) Halsschlagader (*A. carotis*), (d) Drosselvene (*V. jugularis*), und (e) Rippenknochen im Kontext von Wirbelsäule und Becken mit starken Blockartefakten und abgelösten Elementen.

und Terrassenartefakten. Block- und Terrassenartefakte können hierbei zu visuell auffälligen und diagnostisch irreführenden Einschnürungen führen. Weiterhin ist es sehr wahrscheinlich, dass einige Teile solch dünner und filigraner Objekte nicht korrekt in Bildaufnahme und Segmentierung erfasst werden. Entsprechend können resultierende Oberflächenmodelle eine erhöhte Anzahl an abgelösten Elementen enthalten.

Elongierte Objekte weisen meist mehrfach Verzweigungen auf, deren Vorhandensein, Anzahl und Verzweigungsverhalten von Relevanz ist. Glättungsalgorithmen können diese Verzweigungen verfälschen sowie dünne Objektteile schrumpfen und kollabieren lassen, was zu weiteren Verzerrungen und abgelösten Elementen führen kann.

## 5.2 Anforderungen an die Glättung

Aus der Analyse medizinischer Oberflächenmodelle, typischer Probleme und unter Berücksichtigung der medizinischen Anwendung werden nachstehend Anforderungen an die Glättung von medizinischen Oberflächenmodellen abgeleitet.

In erster Linie soll durch eine Glättung der Oberflächenmodelle (*I*) eine Verbesserung der Oberfläche in Richtung Original erzielt werden. Um dies zu erreichen, sind bei allen Strukturen *I-a* Blockartefakte und (vorwiegend bei kompakten Objekten) *I-b* Terrassenartefakte zu reduzieren. Speziell bei flachen Objekten wäre *I-c* das Schließen von Löchern und *I-d* das Ausgleichen von Ausfransungen wünschenswert. Bei elongierten Objekten wäre das Wiederherstellen eines zusammenhängenden und kontinuierlichen Verlaufs durch *I-e* Verbinden abgelöster Elemente und durch *I-f* Rekonstruktion eines kontinuierlichen Durchmesserlaufs denkbar. Jedoch sollten die Anforderung *I-c* bis *I-f* eher in der Bildgebung und Segmentierung behoben werden und sollten im Nachhinein nur durch spezielle Rekonstruktionsmechanismen (siehe z.B. [Selle et al. 2002; Tyrrell et al. 2007]) und nicht durch allgemeine Oberflächenglättungsverfahren behandelt werden. Jedoch dürfen *I-g* entsprechend vorhandene Artefakte nicht durch die Glättung verstärkt bzw. neue hervorgerufen werden.

Je nach Objektklasse und dem Verhältnis von Dreiecksgröße zu Feautregröße von Oberflächenmodellen sind unterschiedlich große Abweichungen zwischen geglätteten und originalen Oberflächenmodellen akzeptabel. Um ein Überschreiten bestimmter Fehlergrenzen zu vermeiden, sind (*II*) Genauigkeits- und Abbruchkriterien für die Glättung erforderlich. Diese sollten garantieren, dass *II-a* die objektspezifische Form, *II-b* das Volumen und *II-c* räumliche Ausdehnungen und Abstände erhalten bleiben.

**Tabelle 5.1:** Anforderungen an die Glättung von medizinischen Modellen.

<b><i>I     Glättung:</i></b>	
I-a	Reduktion von Blockartefakten
I-b	Reduktion von Terrassenartefakten
I-c ... I-f	Korrektur topologischer Fehler (Löcher, Ausfransungen, ...)
I-g	kein Verstärken existierender bzw. Erzeugen neuer Artefakte
<b><i>II    Genauigkeit:</i></b>	
II-a	Erhalt der objektspezifischen Form
II-b	Erhalt des Volumens
II-c	Erhalt der räumlichen Ausdehnungen und Abstände
<b><i>III   praktische Anwendbarkeit:</i></b>	
III-a	leichte bzw. automatische Parametrisierung
III-b	effizientes Laufzeitverhalten
<b><i>IV   Dreieckselemente:</i></b>	
IV-a	Verbesserung der Dreiecksqualität
IV-b	Reduktion der Dreiecksanzahl

Mit Blick auf (*III*) die praktische Anwendbarkeit der Oberflächennetzglättung in medizinischen Anwendungen sind *III-a* leicht bzw. automatisch parametrisierbare Glättungsverfahren wünschenswert. Für die eventuell nötige manuelle Parametrisierung oder Anpassung der Verfahren bei auftretenden Problemen ist *III-b* ein effizientes Laufzeitverhalten erforderlich.

Weitere Ziele bei der Behandlung von Oberflächennetzen können *IV-a* die Verbesserung der Dreiecksqualität und *IV-b* die Reduktion der Dreiecksanzahl sein. Die Glättung von Polygonnetzen kann bei beiden Zielen unterstützend eingesetzt werden, jedoch unterscheiden sich beide Ziele zum Teil stark von der Glättung. Während bei der Glättung das Erscheinungsbild der Oberflächen verändert werden soll, soll dies bei der Dreiecksreduktion und bei der Verbesserung der Dreiecksqualität gerade vermieden werden.

Tabelle 5.1 stellt alle Anforderungen noch einmal im Überblick dar, wobei alle für die folgende Evaluierung relevanten Anforderungen schwarz gedruckt sind.

## 5.3 Kriterien für die Bewertung der Glättungsergebnisse

Um die Erfüllung der Anforderungen an die Glättung von medizinischen Oberflächenmodellen (siehe Abschnitt 5.2) im Glättungsprozess überprüfen und bewerten zu können, werden nachstehend Kriterien zusammengetragen und analysiert, die eine Bewertung der Qualität und Genauigkeit der Glättungsergebnisse in Bezug zu den originalen (artefaktbehafteten) Oberflächenmodellen erlauben.

### 5.3.1 Bewertung der Glattheit

Zur Bewertung der erzielten Glattheit ( $\uparrow$  I) eignen sich: die visuelle Inspektion, die Berechnung von Krümmungswerten sowie die Messung des Flächeninhalts.

Zur visuellen Inspektion werden Glättungsergebnisse verschiedener Verfahren und unterschiedlicher Parametrisierungen dieser gegenübergestellt und ihre Güte visuell qualitativ beurteilt.

Quantitativ kann die erreichte Glattheit über die lokale Krümmung bestimmt werden. Bei den Krümmungswerten hat sich zur Beurteilung der erreichten Glattheit (bei nicht kantenerhaltenden Verfahren) die *Maximale Krümmung* als am aussagekräftigsten erwiesen, da sich diese im Gegensatz zur *Mittleren Krümmung* merklich unter der Glättung verringert [Haase 2005]. Zur quantitativen Beurteilung der Glattheit erweist sich auch die Berechnung des Flächeninhalts als effektive Methode. Bei Volumenerhalt schrumpft der Flächeninhalt beim Glätten um das reduzierte Rauschen und ist somit ein Maß für die erreichte Glättung.

### 5.3.2 Bewertung der Genauigkeit

Als Kriterien zur Beurteilung der Abweichung bzw. des Fehlers des geglätteten Modells zum Original ( $\uparrow$  II) können Messungen des Volumens, der Hauptachsenausdehnungen und der Distanz zwischen geglättetem und Originalmodell verwendet werden.

Das Volumen  $V(\mathcal{M})$  eines beliebigen Dreiecksnetzes  $\mathcal{M}$  kann dabei effizient über die vorzeichenbehaftete Summe der Volumina aller Tetraeder, welche jedes Dreieck des Oberflächenmodells mit dem Koordinatenursprung aufspannen, berechnet werden [Desbrun et al. 1999]:

$$V(\mathcal{M}) = \frac{1}{6} \sum_{i=1}^{|\mathcal{F}|} g(f_i) \overrightarrow{n(f_i)} \quad (5.1)$$

$$\text{mit } g(f_i) = \frac{v_{1i} + v_{2i} + v_{3i}}{3} \text{ und } \overrightarrow{n(f_i)} = \overrightarrow{v_{2i} - v_{1i}} \times \overrightarrow{v_{3i} - v_{1i}}$$

wobei  $v_{1i}$ ,  $v_{2i}$  und  $v_{3i}$  die Eckpunkte des  $i$ -ten Dreiecks  $f_i$  sind und die Summe über die Anzahl  $|\mathcal{F}|$  aller Dreiecke im Oberflächennetz geht. Durch Berechnung des Volumens  $V(\mathcal{M})$  des ungeglätteten und des Volumens  $V(\mathcal{M}')$  des geglätteten Modells kann der Erhalt des Volumens quantifiziert werden.

Neben dem globalen Maß des Volumens bietet sich die Berechnung der symmetrischen Hausdorff-Distanz an, über die lokal für jeden Oberflächenpunkt der Fehler zum Ausgangsmodell bestimmt werden kann. Hierzu wird der minimale Abstand jedes Punktes im geglätteten Modell  $v' \in \mathcal{M}'$  zu allen Dreiecken im Ausgangsmodell  $f \in \mathcal{M}$  und umgekehrt berechnet. Cignoni et al. [1998] und Aspert et al. [2002] beschreiben effiziente Methoden zur Berechnung der symmetrischen Hausdorff-Distanz zwischen Oberflächenmodellen, welche jeweils in den publizierten Systemen METRO<sup>1</sup> und MESH<sup>2</sup> zur Verfügung stehen.

Zusätzlich kann auch durch Sichtkontrollen und entsprechend angemessene Visualisierungen der Abstand zum Original beurteilt werden.

### 5.3.3 Bewertung der praktischen Anwendbarkeit

Als zusätzliches Kriterium für die praktische Angemessenheit bzw. Anwendbarkeit ( $\uparrow$  III) der Glättungsverfahren werden ihr Berechnungsaufwand und die daraus resultierende Laufzeit verwendet. Auch die Komplexität der geeigneten Parametrisierung der Glättungsverfahren für unterschiedliche Modelle mit unterschiedlichen Eigenschaften ist ein wesentliches Kriterium im praktischen Einsatz.

<sup>1</sup><http://vcg.sourceforge.net/tiki-index.php?page=Metro>

<sup>2</sup><http://mesh.epfl.ch>

## 5.4 Evaluation: Konzept und Durchführung

Bisherige Gegenüberstellungen von Glättungsmethoden vergleichen meist nur unterschiedliche Glättungsalgorithmen mit fest eingestellten, empirisch ermittelten Parametern, wobei deren Wahl nicht weiter betrachtet wird. Im Kontrast dazu ist ein Vergleich von Glättungsverfahren mit veränderlichen Parametern viel komplexer, aber notwendig für einen angemessenen Vergleich von erreichbaren Glättungsergebnissen. Die zu vergleichenden Glättungsverfahren und die jeweilige veränderliche Parametrisierung werden im folgenden Abschnitt vorgestellt.

### 5.4.1 Glättungsverfahren und Parametrisierung

Wie bereits in Abschnitt 4.3.6 erwähnt, eignen sich merkmalerhaltende Glättungsverfahren nicht, um Block- und Terrassenartefakte zu reduzieren. Ohne die merkmalerhaltende Gewichtung bei der Glättung sind die merkmalerhaltenden Verfahren jedoch identisch mit den in Abschnitt 4.3 vorgestellten isotropen Glättungsverfahren. Für die Evaluation werden entsprechend alle hier besprochenen isotropen Glättungsverfahren herangezogen.

Um den Vergleich der Techniken auch mit variablen Parametern überschaubar zu halten, soll jedes Verfahren mit fünf unterschiedlichen Iterationszahlen  $\iota$  und sechs repräsentativen Wichtungskombinationen untersucht werden. Für die getesteten Iterationszahlen wird eine typische Verteilung mit  $\iota = 5, 10, 20$  und  $50$  Iterationen gewählt. Für die Schrittweite bzw. Wichtung der Glättung pro Iteration werden die Werte  $\lambda = 0,05; 0,1; 0,3; 0,5; 0,7$  und  $0,9$  gewählt, die gut übliche Werte aus den betrachteten Veröffentlichungen abdecken.

Als Nachbarschaft für die Glättung wird die topologische Nachbarschaft 1. Grades (siehe Abschnitt 3.4.1) gewählt. Die Beschränkung erscheint hier sinnvoll, da die Glättung mittels *Mean Curvature Flow* nicht für eine größere Nachbarschaft definiert ist und weil die Ergebnisse der höheren Nachbarschaft über eine entsprechende Erhöhung der Iterationsanzahl auch annähernd mit der Nachbarschaft 1. Grades erreicht werden können. Untersuchungen der Nachbarschaft 2. Grades sind in [Bade et al. 2006a] zu finden und fließen in die finalen Glättungsempfehlungen mit ein.

Neben dieser globalen Parametrisierung besitzen die zu untersuchenden Glättungsverfahren unterschiedliche zusätzliche Parameter. Die Parametrisierung der Techniken wird deshalb nachstehend einzeln besprochen.

**Laplace-Glättung:** Der Laplace-Filter bietet neben der Iterationsanzahl  $\iota$  und dem Wichtungsfaktor  $\lambda$  keine weiteren Parameter und kann somit mit den gegebenen Parameterwerten evaluiert werden. Geringe  $\lambda$ -Werte und eine geringe Anzahl  $\iota$  an Iterationen führt dabei zu geringer Glättung, während höhere Werte eine stärkere Glättung erzielen.

**Laplace+ $\mathcal{HC}$ -Glättung:** Die Laplace+ $\mathcal{HC}$ -Glättung hat zwei zusätzliche Parameter ( $\alpha$  und  $\beta$ ). Um für die Untersuchung des Filters keinen vierdimensionalen Suchraum aufspannen zu müssen, werden die zwei zusätzlichen Parameter auf übliche Werte festgesetzt. Vollmer et al. [1999] konnten zeigen, dass das Laplace+ $\mathcal{HC}$ -Verfahren konvergiert, wenn  $\alpha > 0$  und  $\beta > 0,5$  gewählt wird. Jedoch gilt nur für  $\alpha = 0$  und  $\beta = 0,5$ , dass hohe Frequenzen entfernt und niedrige Frequenzen erhalten bleiben [Vollmer et al. 1999]. Entsprechend wird für die Untersuchung  $\alpha = 0$  gesetzt, so dass alle Punkte der Oberfläche nur in Richtung ihrer vorherigen Position verschoben werden und es wird  $\beta = 0,5$  gesetzt, so dass der

Verschiebungsvektor an einem Oberflächenpunkt und der gemittelte Verschiebungsvektor seiner Nachbarn zu gleichen Anteilen in die Laplace+ $\mathcal{HC}$ -Glättung eingehen. Entsprechend ergibt sich für die verwendete Parametrisierung der Laplace+ $\mathcal{HC}$ -Glättung aus Formel 4.3 die aggregierte Form:

$$v_i' = v_i^* - 0,5 \left( (v_i^* - v_i) + \frac{1}{m} \sum_{j=1}^m (u_j^* - u_j) \right) \quad (5.2)$$

**Taubin's  $\lambda$ - $\mu$ -Glättung:** Im Gegensatz zum Laplace-Filter besitzt der  $\lambda$ - $\mu$ -Filter den zusätzlichen Parameter  $\mu$  und die sich aus der Kombination von  $\lambda$  und  $\mu$  ergebende Pass-Band-Frequenz (repräsentiert als  $k_{PB}$ ). Wie bereits in Abschnitt 4.3.3 erläutert, führen hohe absolute Werte für  $\lambda$  und  $\mu$  und ein kleines  $k_{PB}$  dabei zu einer schnellen und viele Frequenzen umfassenden Glättung. Für den Vergleich mit den anderen Glättungsverfahren wurden deshalb geeignete Parameterkombinationen ausgewählt. Beginnend bei einer langsamen ( $\lambda = 0,05$ ) und geringen ( $k_{PB} = 1$ ) Glättung bis hin zu einer schnell konvergierenden ( $\lambda = 0,9$ ) und umfassenden Glättung ( $k_{PB} = 0,05$ ) wird das Parameterspektrum repräsentativ abgedeckt.

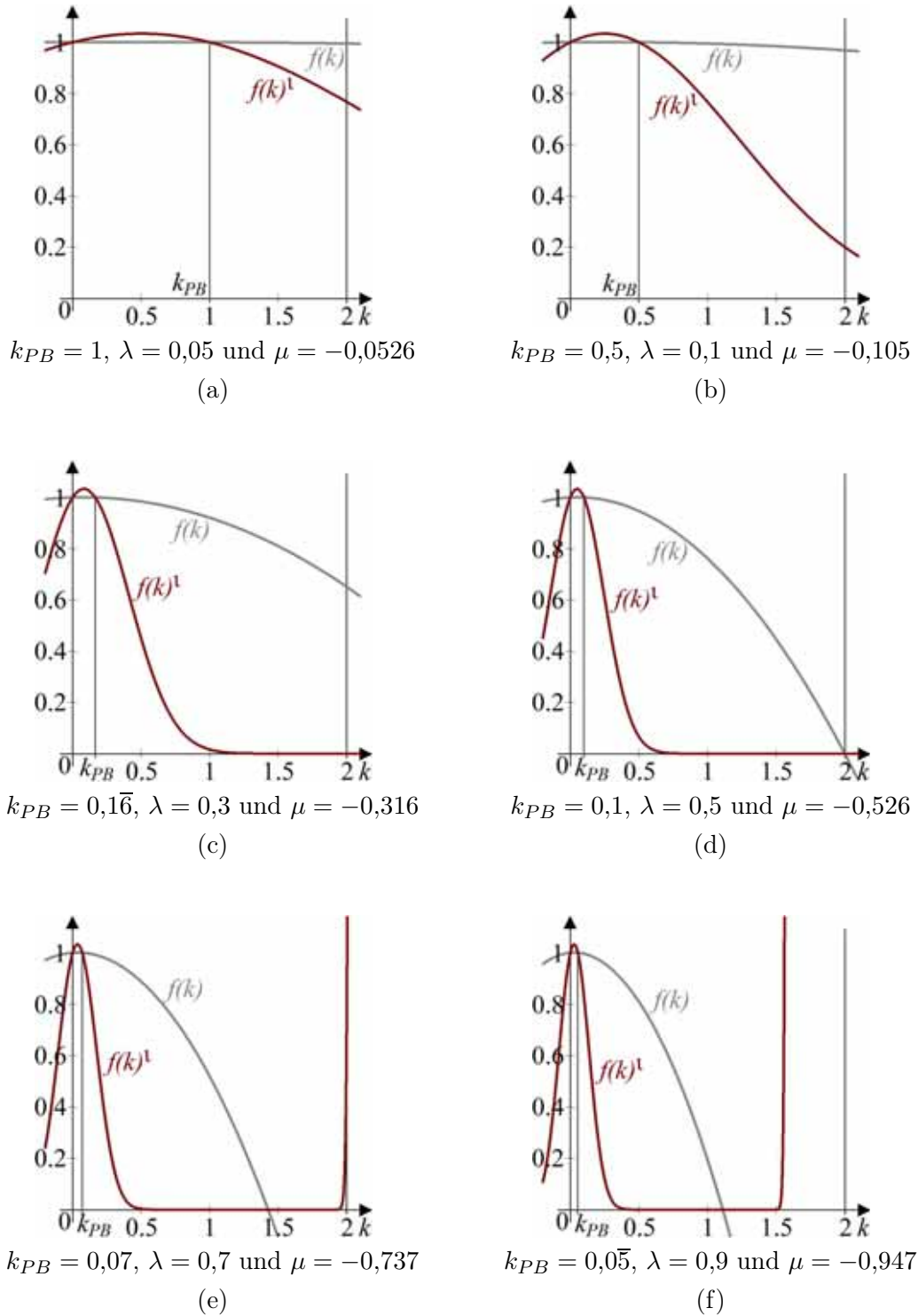
Da in [Taubin 1995b; Taubin et al. 1996; Taubin 2000]  $k_{PB} = 0,1$ ,  $\lambda = 0,6307$  und  $\mu = -0,6732$  für alle Beispiele verwendet wird, soll dies auch in der Evaluation verwendet werden. Jedoch werden etwas kleinere Werte  $\lambda = 0,5$  und  $\mu = -0,526$  gewählt, welche die resultierende Filterfunktion etwas langsamer an die Pass-Band-Frequenz  $k_{PB}$  annähern lassen. Dies erscheint sinnvoll, da Taubin's  $\lambda$ - $\mu$ -Glättung das Schrumpfen nicht gänzlich verhindern kann und ein etwas langsames Annähern an die Pass-Band-Frequenz einen besseren Volumenerhalt bei moderatem Mehraufwand an Iterationen verspricht.

Abbildung 5.5 stellt die ausgewählten Parameterkombinationen und resultierenden Filterfunktionen grafisch dar. Eine Besonderheit repräsentiert dabei die maximale Parameterkombination (siehe Abb. 5.5 (f)), da diese im Definitionsbereich  $k = [0, 2]$  der Funktion  $f(k)$  nach einigen Iterationen wieder dramatisch ansteigt. Die in Abb. 5.5 (e) gezeigte Filterfunktion stellt einen ähnlichen Fall dar, der aber bei einer Anzahl von  $\iota = 50$  Iterationen einen Grenzfall darstellt. Entsprechend führen hohe  $\lambda$ -Werte zu einem schnelleren Annähern an die Pass-Band-Frequenz, wodurch aber auch instabile Filter entstehen können [Taubin et al. 1996]. Sinnvoll ist die Wahl dieser Parameterkombination in der Evaluierung, damit auch Auswirkungen der möglichen Instabilität von Taubin's  $\lambda$ - $\mu$ -Glättung überprüft werden können.

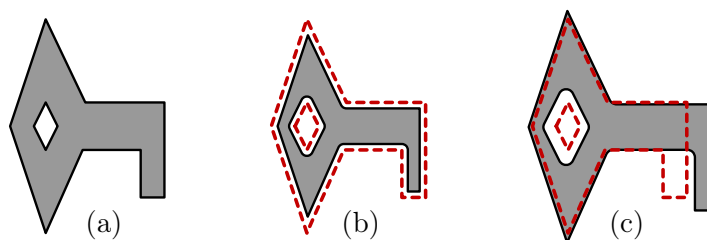
**Mean Curvature Flow Glättung:** Die *Mean Curvature Flow* Glättung wird, wie die einfache Laplace-Glättung, nur über den Wichtungsfaktor  $\lambda$  und die Anzahl der durchgeführten Iterationen  $\iota$  parametrisiert (siehe Abschnitt 4.3.4). Desbrun et al. [1999] verwenden jedoch eine Volumenskalierung, um dem Schrumpfen der geglätteten Oberflächenmodelle entgegenzuwirken. Hierzu wird das Volumen vor der Glättung  $V(\mathcal{M})$  und nach der Glättung  $V(\mathcal{M}')$  über Formel 5.1 berechnet. Danach werden die 3D-Koordinaten jedes Punktes  $v_i'$  des geglätteten Oberflächennetzes entsprechend dem Verhältnis der Volumina wie folgt im Raum skaliert [Desbrun et al. 1999]:

$$v_i^\# = v_i' \cdot \left( \frac{V(\mathcal{M})}{V(\mathcal{M}')} \right)^{\frac{1}{3}} \quad (5.3)$$

Durch die Skalierung wird das exakte Ausgangsvolumen wieder hergestellt. Dabei werden alle Frequenzen verstärkt, was in Kombination mit der Glättung das Reduzieren hoher Frequenzen weiterhin unbehindert ermöglicht [Desbrun et al. 1999].



**Abbildung 5.5:** Ausgewählte Parameterkombinationen für Taubin's  $\lambda$ - $\mu$ -Glättung und resultierende Filterfunktionen  $f(k)$  nach einer und  $f(k)^\iota$  nach  $\iota = 50$  Iterationen. Mit  $k \in [0, 2]$  als Repräsentation des Frequenzspektrums und  $k_{PB}$  als Pass-Band-Frequenz (siehe Abschnitt 4.3.3).



**Abbildung 5.6:** Probleme beim Volumenerhalt durch Volumenskalierung nach Desbrun et al. [1999]: (a) Ausgangsobjekt, (b) geschrumpftes Objekt nach Glättung und (c) auf Ausgangsvolumen skaliertes Objekt.

Die entsprechend skalierten Oberflächenmodelle verändern jedoch ihre absolute Position, wenn auch die Oberflächenpunkte ihre relative Position beibehalten. Zusätzlich schrumpfen Oberflächenmodelle durch eine Glättung lokal unterschiedlich, wohingegen das Skalieren ein global einheitlicher Prozess ist. Bei der Glättung einzelner kompakter Objekte führt dies meist nur zur geringen Beeinträchtigung der *relativen* Ergebnisse. Bei allen komplexeren Objekten, wie in Abbildung 5.6 dargestellt, kann es jedoch durch das Skalieren zu erheblichen Veränderungen und Differenzen in der Position und Form einzelner Objektteile kommen. Zusätzlich werden Objekte einer Szene, die unterschiedlich beim Glätten schrumpfen, unterschiedlich stark skaliert und im Raum relativ zueinander verschoben. Entsprechend ist es für Szenen, die aus mehreren Objekten bestehen (z.B. einzelne Knochen, Organe, Blutgefäße, ...) unmöglich, auf diese Weise skalierte Objekte wieder exakt zueinander zu positionieren.

Diesen Beobachtungen folgend ist die von Desbrun et al. [1999] eingeführte und auch in anderen Publikationen verwendete Volumenskalierung (siehe [Fleishman et al. 2003]) ungeeignet für medizinische Oberflächenmodelle in der computerunterstützten Chirurgie. Folglich findet die Volumenskalierung keine Anwendung in der vorliegenden Arbeit.

**Mean- und Median-Glättung:** Die in Abschnitt 4.3.5 vorgestellte *Mean*- und *Median*-Glättung besitzen, wie der Laplace-Filter nur die Iterationsanzahl  $\iota$  und den Wichtungsfaktor  $\lambda$  als Parameter. Entsprechend werden zu Vergleichszwecken die *Mean*- und *Median*-Glättung mit den gleichen fünf Iterationsstufen und sechs  $\lambda$ -Werten untersucht, wie die anderen vier Verfahren.







## 5.4.2 Auswahl der Oberflächenmodelle

Zur Evaluation der Glättungsverfahren werden diese an sechs Referenzmodellen untersucht. Als Referenzmodelle werden dazu jeweils zwei typische Vertreter ihrer Objektklasse mit typischen Eigenschaften und Artefakten ausgewählt (siehe Tabelle 5.2).

Aus der Gruppe der kompakten Objekte wird ein Oberflächenmodell einer Leber und eines Lymphknotens verwendet. Während beide starke Block- und Terrassenartefakte aufweisen, repräsentiert das Oberflächenmodell des Lymphknotens ein kompaktes *low-res.* Objekt mit einem schlechten Verhältnis von Dreiecksgröße zu Featuregröße. Ausgewählte Vertreter für die flachen Objekte sind die Oberflächenmodelle eines Halsmuskels (*M. sternocleidomastoideus*) sowie eines Beckenknochens. Ein stark verzweigter Gefäßbaum mit vielen abgelösten Elementen und ein etwas dickeres unverzweigtes Oberflächenmodell einer Halsschlagader (*A. carotis*) werden als Vertreter für die elongierten Objekte verwendet.



**Tabelle 5.2:** Zur Evaluation der Glättungsverfahren verwendete Oberflächenmodelle und ihre Eigenschaften.

						
	Leber	Lymphknoten	Halsmuskel	Becken	Gefäßbaum	A. carotis
Faces	69 480	3 294	20 300	65 554	15 730	2 056
Vertices	34 738	1 649	10 146	32 735	7 873	1 032
Voxel	169 485	1 664	11 174	55 523	4 224	637

Mit den ausgewählten Oberflächenmodellen, variablen Parametern und zu untersuchenden Glättungsverfahren sind bei der geplanten Evaluierung (6 Glättungsverfahren  $\times$  4 Iterationsstufen  $\times$  6 Wichtungsfaktoren  $\times$  6 Oberflächenmodelle  $=$ ) 864 Glättungsergebnisse zu analysieren und zu bewerten.

### 5.4.3 Ausgewählte Bewertungskriterien

Um den Vergleich der Glättungsverfahren und der erhaltenen Glättungsergebnisse übersichtlich zu halten, wird sich im Weiteren zur Bewertung der Glattheit der Oberflächenmodelle ( $\uparrow$ I) nur auf die Verteilung der *Maximalen Krümmung*  $k_1$  und auf die visuelle Inspektion der Ergebnisse konzentriert. Zur Bewertung der Genauigkeit bzw. des Fehlers zum originalen Oberflächenmodell ( $\uparrow$ II) werden die maximale symmetrische Hausdorff-Distanz  $D_{sH}(\mathcal{M}, \mathcal{M}')$  (als Maß des maximalen Fehlers) und das Restvolumen in Prozent des Originalvolumens  $V_{\%} = 100\% \times \frac{V(\mathcal{M}')}{V(\mathcal{M})}$  (zur Bewertung des Schrumpfens) herangezogen. Zur Beurteilung der praktischen Anwendbarkeit ( $\uparrow$ III) werden Effizienz und Rechenkomplexität über die von jedem Glättungsalgorithmus für jedes Oberflächenmodell benötigte Zeit  $t$  (auf einem PC mit 3.2GHz Intel<sup>®</sup> Pentium<sup>®</sup> 4 Prozessor und 1GB RAM) erfasst. Die Komplexität der geeigneten Parametrisierung der Verfahren wird zusätzlich konzeptionell in der Auswertung als Kriterium für die praktische Anwendbarkeit diskutiert.

Die Ergebnisse der Glättung spannen pro Glättungsverfahren und betrachtetem Objekt jeweils eine Matrix aus den untersuchten Iterationsstufen und Wichtungsstärken auf. Aus Analyse und Vergleich dieser Matrizen können die besten Glättungsverfahren und Parameter extrahiert und Glättungsempfehlungen für die verschiedenen Objektklassen abgeleitet werden. Die Präsentation aller Untersuchungsergebnisse würde den Rahmen der Arbeit sprengen, so werden im Folgenden nur ausgewählte Beispiele näher erläutert. Weiterführende Ergebnisse und Resultate sind auf den Projektwebseiten [Vis-Research Group 2007] zu finden.

## 5.5 Umsetzung und Implementierung

Für die Untersuchung und Evaluierung der Glättungsverfahren wurden die Laplace-, Laplace+ $\mathcal{HC}$ -, Taubin's  $\lambda$ - $\mu$ -, die *Mean Curvature Flow*-, *Mean*- und *Median*-Glättung einheitlich in einer gemeinsamen Softwareumgebung implementiert. Dies ermöglicht den direkten Vergleich der Glättungsverfahren unabhängig von Implementierungsdetails wie verwendeten Datenstrukturen, Speichermanagement und anderen Parametern verschiedener Softwareumgebungen.

Als gemeinsame Softwareumgebung wurde MEVISLAB von MeVis Research gewählt [MeVis Research 2007; MeVis 2007]. MEVISLAB ist eine Programmier- und Rapid Prototyping Umgebung für die medizinische Bildanalyse und Visualisierung, auf deren Basis Anwendungen zur Diagnose und Operationsplanung, zum Beispiel in der Leberchirurgie, existieren. Zur Implementierung wurde eine erweiterte MEVISLAB Version mit freundlicher Unterstützung von MeVis Research zur Verfügung gestellt, die im Rahmen dieser Arbeit um die in C++ programmierten Glättungsverfahren erweitert wurde.

### 5.5.1 Implementierungskonzept

Für die Implementierung wurde ein modulbasierter Ansatz gewählt, der definierte Schnittstellen zur Einbindung der Oberflächennetzglättung in die Pipeline komplexer Oberflächennetzverarbeitung bereitstellt. Das Glättungsmodul benötigt dabei, wie in Abbildung 5.7 dargestellt, als Eingabe ein Oberflächennetz und Parameter, die definieren, welches Glättungsverfahren mit welchen Parametern Verwendung findet. Nach der Verarbeitung im Modul werden das geglättete Oberflächenmodell sowie Parameter bzw. Angaben zur Bewertung des Glättungsergebnisses (siehe Abschnitt 5.4.3) nach außen gegeben.

Weiterhin wurden alle Glättungsverfahren als Submodul des Glättungsmoduls implementiert. Dabei bilden die verschiedenen Arbeitsschritte, wie das Ermitteln und explizite Speichern der Nachbarschaften sowie der Korrekturteil (das Zurückverschieben) bei der Laplace+ $\mathcal{HC}$ -Glättung eigene Submodule.

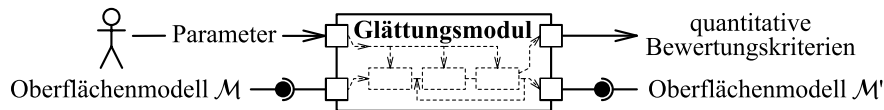
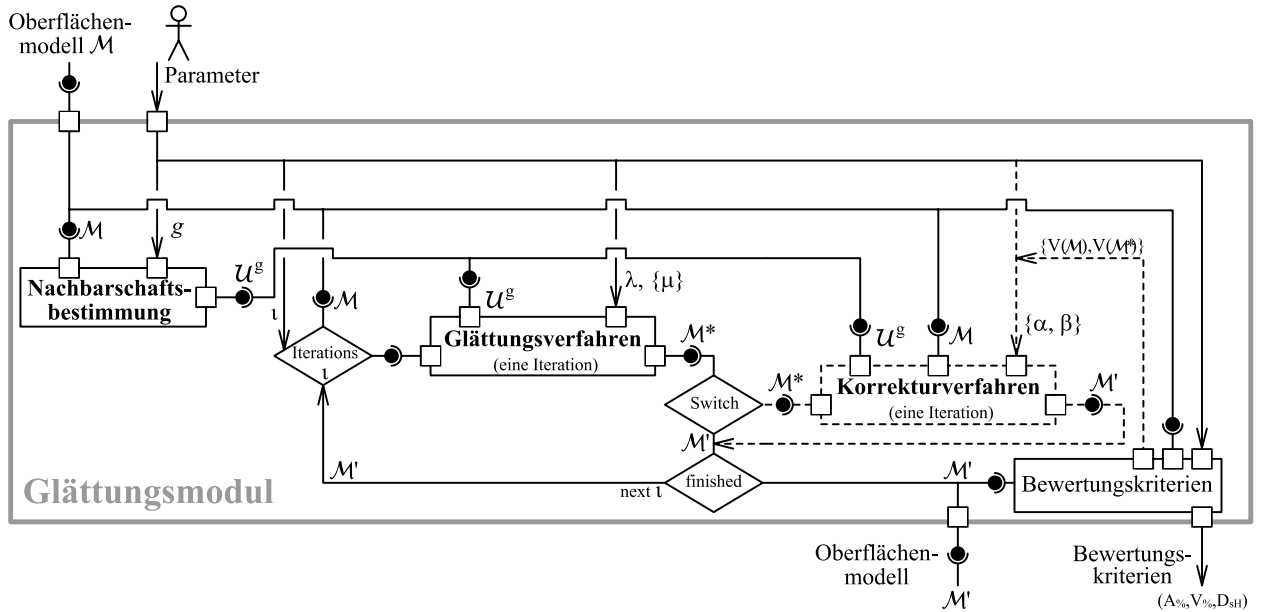


Abbildung 5.7: UML-Diagramm des Glättungsmoduls.

### 5.5.2 Modularisierung der Glättungsverfahren

Die Glättungsverfahren werden zur effizienten Umsetzung in die separaten Arbeitsabschnitte Nachbarschaftsbestimmung, Oberflächennetzglättung und Korrektur zum Volumenerhalt aufgeteilt. Ein weiteres Submodul sorgt für die Berechnung der quantitativen Bewertungskriterien, wie Volumen, Oberflächeninhalt und symmetrische Hausdorff-Distanz des Ausgangs- und Ergebnisoberflächenmodells. Diese Modularisierung erlaubt es auch anderen Glättungsverfahren, wie zum Beispiel der *Mean Curvature Flow* Glättung, von volumenerhaltenden Korrekturverfahren z.B. aus der Laplace+ $\mathcal{HC}$ -Glättung zu profitieren. Weiterhin stehen Erweiterungen der einzelnen Submodule gleich allen Glättungsverfahren zur Verfügung. So kann eine Erweiterung um anisotrope Nachbarschaften im Submodul zur Nachbarschaftsbestimmung danach gleich allen Glättungsverfahren zur Verfügung stehen. Auch die Implementierung zukünftig entwickelter Korrekturverfahren ist denkbar, die durch die Hintereinanderschaltung der einzelnen Submodule dann wiederum mit allen Glättungsverfahren kombiniert werden können.

Im Folgenden werden die einzelnen Submodule für die Glättung näher erläutert und Details ihrer Implementierung dargelegt. Abbildung 5.8 stellt im Detail das im Rahmen dieser Arbeit entwickelte Glättungsmodul und das Zusammenspiel aller Submodule grafisch dar.



**Abbildung 5.8:** Entwickeltes Glättungsmodul und Zusammenspiel aller Submodule im UML-Diagramm.

### Nachbarschaftsbestimmung und -repräsentation

Die Bestimmung der für die Glättung erforderlichen topologischen Nachbarschaften stellt das erste Submodul dar. Als Input sind das Oberflächennetz und als Parameter der gewünschte Grad  $g$  der topologischen Nachbarschaft erforderlich. Im Folgenden wird auch die verwendete Datenstruktur zur Repräsentation der Oberflächennetze näher erläutert.

MEVISLAB bietet intern die Repräsentation der Oberflächennetze über eine erweiterte *Winged-Edge* Datenstruktur an, welche zusätzlich von jedem Eckpunkt  $v_i$  des Oberflächennetzes einen Zeiger auf jede in ihm zusammenlaufende Kante und Fläche hat sowie jede Fläche  $f_i$  einen Zeiger auf alle sie definierenden Kanten und Eckpunkte besitzt (vergleiche Abschnitt 3.6.3). Diese *Winged-Edge* Datenstruktur ist relativ mächtig, bedarf aber eines entsprechend höheren Speicheraufwandes gegenüber einer Standard-*Winged-Edge* Repräsentation (Abschnitt 3.6.3). Dies hat jedoch keinen negativen Einfluss auf die Performance der Glättungsalgorithmen, da diese nur die Position der Eckpunkte eines Oberflächennetzes verändern müssen und alle anderen Zusatzinformationen der Datenstruktur unverändert lassen.

Die topologische Nachbarschaft ersten Grades  $\mathcal{U}_{v_i}^1$  eines Eckpunktes  $v_i$  ist in der verwendeten *Winged-Edge* Datenstruktur über das Traversieren aller Kanten  $e_j$ , die in  $v_i$  zusammenlaufen möglich. Dabei bilden dann die Endpunkte  $u_j$  aller Kanten  $e_j$  die topologische Nachbarschaft ersten Grades  $u_j \in \mathcal{U}_{v_i}^1$ . Bei der Bestimmung von Nachbarschaften höheren Grades ist jedoch zusätzlich Sorge zu tragen, dass keine Punkte mehrfach in die Nachbarschaft aufgenommen werden. Zusätzlich steigt bei einer naiven Implementierung der Aufwand zur Bestimmung der Nachbarschaft quadratisch an.

Implementiert wurde deshalb eine lineare Nachbarschaftssuche (siehe Listing 5.1), deren Ergebnis als zusätzliche explizite Information mit Zeigern auf alle Nachbarn eines gegebenen Nachbarschaftsgrades für jeden Oberflächenpunkt gespeichert wird. Dies ermöglicht einen minimalen linearen Aufwand<sup>3</sup>, um in jeder Iteration der Glättungsverfahren alle Nachbarpunkte zu traversieren. Das Ergebnis dieses Submoduls stellt die explizit gespeicherte Nachbarschaft

<sup>3</sup>Der Aufwand ist linear in Bezug auf die Anzahl  $m$  der topologischen Nachbarn  $g$ -ten Grades, wobei  $m$  üblich mit  $g!$  ansteigt.

**Listing 5.1:** Pseudocode zum Ermitteln der Nachbarn  $g$ -ten Grades von allen Punkten  $v_i$  eines Oberflächennetzes  $\mathcal{M}$ .

---

```

 $\mathcal{M} \leftarrow$  Winged Edge Repräsentation des Oberflächenmodells
 $g \leftarrow$  Grad der Nachbarschaft
FOR alle Eckpunkte  $v_i \in \mathcal{M}$  DO % für alle Eckpunkte
     $\mathcal{U}_{v_i}^g \leftarrow \{ \emptyset \}$  % Liste mit Nachbarn von  $v_i$ 
     $U \leftarrow \{ v_i \}$  % Liste mit aktiven Nachbarn
     $NU \leftarrow \{ \emptyset \}$  % Liste mit neu gefundenen Nachbarn
    FOR  $i$  von 1 bis  $g$  DO % suche Nachbarn 1. bis  $g$ -ten Grades
        FOR alle  $u \in U$  DO % besuche alle aktiven Nachbarn
            FOR alle Kanten  $e_{ij}$  von  $u$  DO % verfolge alle abgehenden Kanten
                 $u_j \leftarrow$  Eckpunkt den  $e_{ij}$  mit  $u$  verbindet
                IF  $u_j$  noch nicht markiert THEN
                    markiere  $u_j$  als besucht
                    füge  $u_j$  in  $\mathcal{U}_{v_i}^g$  ein
                    füge  $u_j$  in  $NU$  ein
                ENDIF
            ENDFOR
        ENDFOR
    ENDFOR
     $U \leftarrow NU$  % mache die neuen Nachbarn zu aktiven
     $NU \leftarrow \{ \emptyset \}$ 
ENDFOR
lösche die Markierung von allen  $u_j \in \mathcal{U}_{v_i}^g$ 
ENDFOR

```

---

dar. Ein UML-Moduldiagramm dieses Nachbarschaftssubmoduls ist in Abbildung 5.9 dargestellt.

Die Bestimmung der Nachbarschaft  $g$ -ten Grades einer Fläche  $f_i$  bei der *Mean*- und *Median*-Glättung wird dabei äquivalent zu der Nachbarschaft der Eckpunkte des Oberflächennetzes bestimmt.

Auch wenn sich diese Arbeit auf die Bestimmung und Repräsentation der topologischen Nachbarschaft beschränkt, ist das beschriebene Vorgehen leicht auf die Berücksichtigung von geodätischen Entfernungen erweiterbar bzw. als Submodul durch speziellere Verfahren (z.B. anisotrope Nachbarschaften) austauschbar.



**Abbildung 5.9:** UML-Diagramm des Submoduls zur Nachbarschaftsbestimmung.

## Glättung

Das zweite Submodul führt die eigentliche Glättung durch. Alle unterschiedlichen Glättungsverfahren führen damit zu einem anderen Submodul, welche dann je nach Bedarf verwendet werden. Als gemeinsame Eingabe erhalten alle Glättungsverfahren das Ausgangsoberflächennetz, die explizit gespeicherten Nachbarschaften und alle nötigen Glättungsparameter. Variabel ist hierbei die Anzahl nötiger Glättungsparameter, die entsprechend dem jeweils gewählten Glättungsverfahren definiert und übergeben werden können.

Entstanden sind hierbei Submodule für die Laplace-, Taubin's  $\lambda$ - $\mu$ -, *Mean Curvature Flow*-, Mean- und Median-Glättung. Für die Laplace+ $\mathcal{HC}$ -Glättung wurde kein eigenes Glättungsverfahren implementiert, da es sich hierbei um eine Laplace-Glättung gefolgt von einem Rückverschiebungs- bzw. Korrekturschritt handelt. Abbildung 5.10 stellt das Submoduldesign zur Glättung in UML-Komponentendiagrammform dar.



Abbildung 5.10: UML-Diagramm für die Submodule der Glättungsverfahren.

## Korrektur

Eine weitere Gruppe an Submodulen bilden die sogenannten Korrekturverfahren (Abb. 5.11). Bei der Laplace+ $\mathcal{HC}$ -Glättung stellt das Zurückverschieben der geglätteten Oberflächenpunkte entsprechend den möglichen Parametern  $\alpha$  und  $\beta$  ein Korrekturverfahren dar. Dieses kann nach einer einfachen Laplace-Glättung vorgenommen werden (entspricht der Laplace+ $\mathcal{HC}$ -Glättung). Dieses Korrekturverfahren kann aber auch nach jeder andersartigen Glättung verwendet werden, um das Schrumpfen zu minimieren. Ein weiteres implementiertes Korrekturverfahren stellt die Volumenskalierung dar, welche jedoch in den Untersuchungen aufgrund der beschriebenen Probleme (siehe Abschnitt 5.4.1) keine weitere Verwendung findet.



Abbildung 5.11: UML-Diagramm der Submodule der Korrekturverfahren.

### 5.5.3 Durchführung

Zur Durchführung wurde in MEVISLAB ein Netzwerk erstellt, welches die Untersuchung der Glättungsverfahren, unter Angabe aller zu untersuchenden Glättungsverfahren und den zu verwendenden Parametern als Batch-Prozess ermöglicht. Als Ergebnisse werden vordefinierte Ansichten, alle Bewertungskriterien und die resultierenden Oberflächennetze gespeichert.

Als Hardware wurde ein handelsüblicher PC mit 3.2GHz Intel® Pentium® 4 Prozessor und 1GB RAM verwendet.

## 5.6 Ergebnisse

Im Folgenden werden die Ergebnisse für die Glättung der Oberflächenmodelle präsentiert und diskutiert. Dazu wird zunächst jede Objektklasse einzeln betrachtet und die erreichten Ergebnisse für die jeweilige Gruppe von Oberflächenmodellen verglichen und ausgewertet. Ergänzend zu den in diesem Kapitel präsentierten Ergebnissen, werden in Anhang A alle beim Vergleich der Glättungsverfahren erhobenen Diagramme aufgeführt.

### 5.6.1 Glättungsergebnisse kompakter Objekte

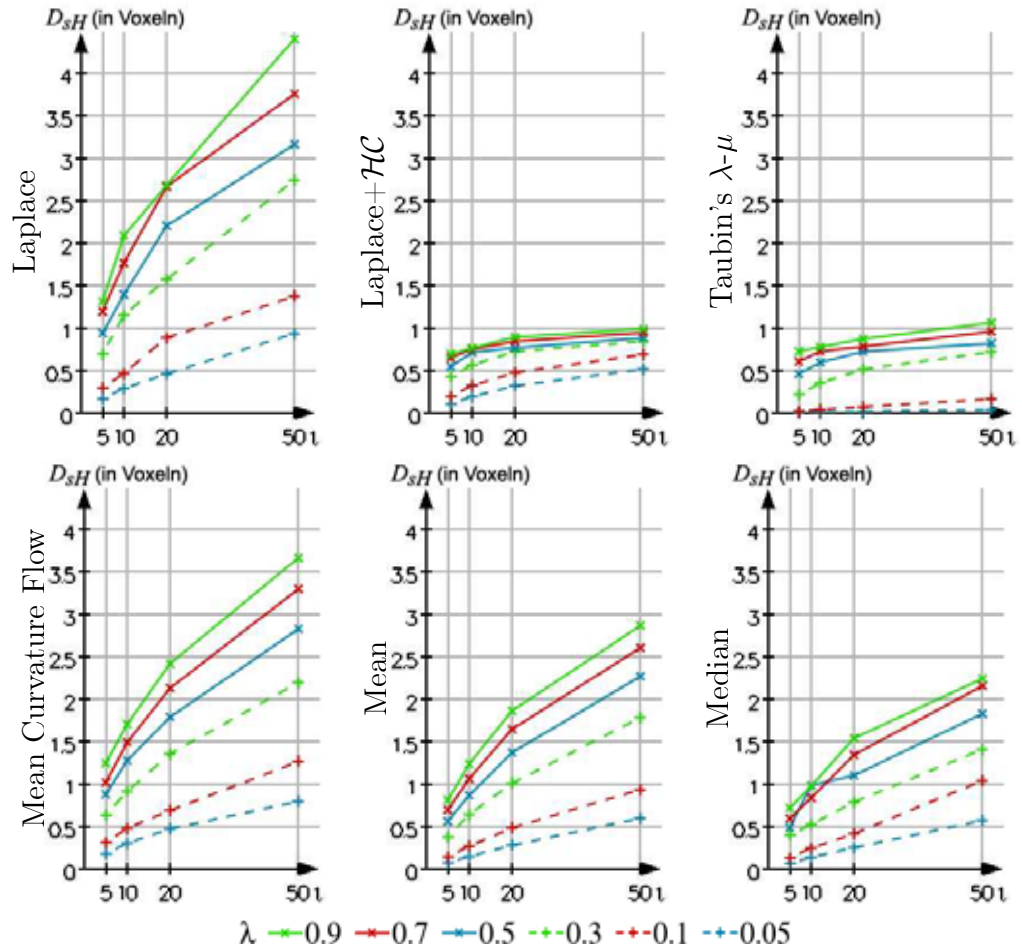
Die kompakten (sowohl *high-res.* als auch *low-res.*) Objekte erwiesen sich bei der Glättung mit allen Verfahren am robustesten gegenüber Volumenverlust und Verzerrung der Geometrie. Abbildung 5.12 stellt die symmetrische Hausdorff-Distanz ( $D_{sH}$  in Voxelkoordinaten) des geglätteten Oberflächenmodells der Leber unter Verwendung der unterschiedlichen Glättungsverfahren bei unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$  gegenüber. Aus dieser Gegenüberstellung für das Lebermodell ist ersichtlich, dass der maximale Fehler selbst bei der Laplace-Glättung mit unter fünf Voxelinheiten in einem moderaten Verhältnis zur Objektgröße steht. Beim *low-res.* Lymphknotenmodell ist der Fehler zwischen Original und Glättungsergebnissen im Verhältnis zur Objektgröße größer. Jedoch sind nur die Laplace- und *Mean Curvature Flow* Glättung mit einem Volumenerhalt  $V_{\%}$  weit unter 90% und einer symmetrischen Hausdorff-Distanz  $D_{sH}$  zwischen 3 und 6 Voxeln für das *low-res.* Lymphknotenmodell als ungeeignet einzustufen.

In Abbildung 5.13 und 5.14 werden die Glättungsergebnisse visuell und bezüglich der maximalen Krümmung bei maximaler untersuchter Parameterwahl ( $\iota = 50$  und  $\lambda = 0,9$ ) gegenübergestellt. Die Laplace- und *Mean Curvature Flow* Glättung erreichen hierbei die stärkste Glättung, allerdings nur mit starkem Schrumpfen der Objekte. Die *Mean*- und *Median*-Glättung erzeugen vermehrt Ergebnisobjekte mit scharfen Kanten (*Median* in Abb. 5.13) und Quaderform (*Mean* und *Median* in Abb. 5.14). Die Laplace+ $\mathcal{HC}$  und Taubin's  $\lambda$ - $\mu$  Glättung erhalten am besten das originale Volumen und die Ergebnisobjekte besitzen die geringste Abweichung  $D_{sH}$  zum Original. Jedoch können die starken Terrassenartefakte durch die Laplace+ $\mathcal{HC}$  und Taubin's  $\lambda$ - $\mu$  Glättung mit den maximalen Parametern der Untersuchung nicht ausreichend geglättet werden. Eine entsprechend höhere Anzahl an Iterationen bei der Laplace+ $\mathcal{HC}$  und Taubin's  $\lambda$ - $\mu$  Glättung kann jedoch potentiell auch die Terrassenartefakte bei immer noch sehr gutem Volumenerhalt  $V_{\%} \simeq 98\%$ , geringer Distanz  $D_{sH} < 3$  und effizienter Rechenzeit  $t < 20\text{sec}$  im kompakten *high-res.* Lebermodell glätten (siehe Abb. 5.15). Jedoch kann Taubin's  $\lambda$ - $\mu$  Glättung mit steigender Iterationsanzahl instabil werden, wie bereits in Abschnitt 5.4.1 erläutert. In Abbildung 5.15 sind Auswirkungen der Instabilität von Taubin's  $\lambda$ - $\mu$  Glättung mit  $\iota = 400$  Iterationen und  $\lambda = 0,9$  vor allem als Verstärkung hoher Frequenzen/Krümmungswerte erkennbar.

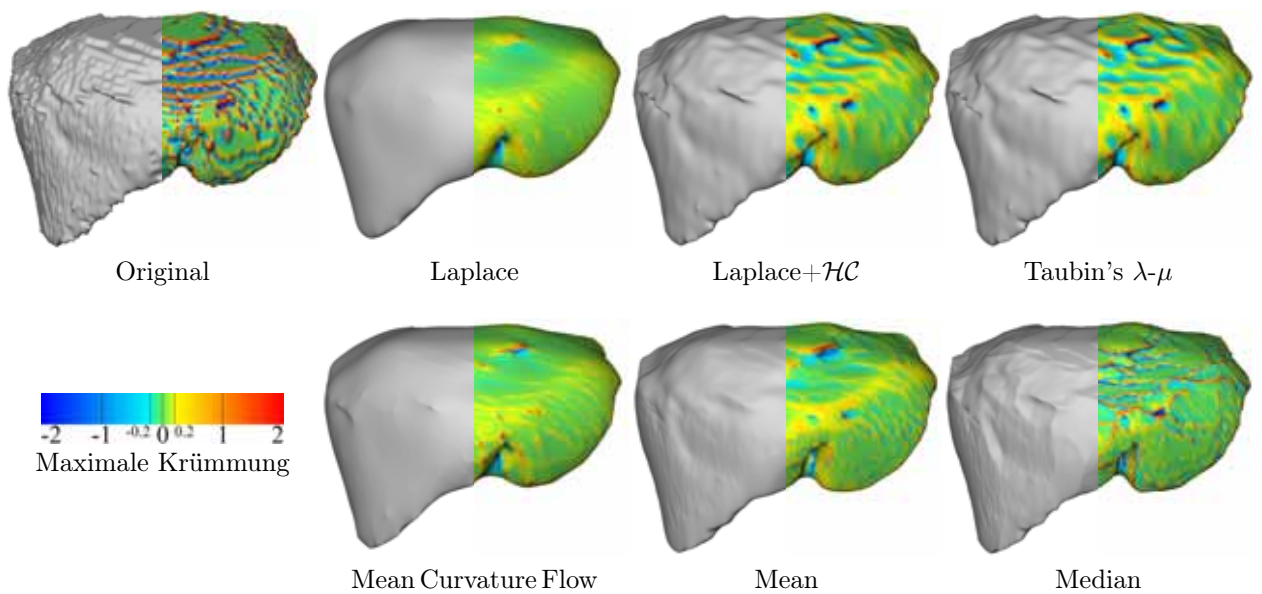
Im kompakten *low-res.* Lymphknotenmodell lässt sich die Terrassenstufe im oberen Drittel jedoch nicht ohne größere Fehlerwerte und Formverfremdung reduzieren.

In Bezug auf die Effizienz benötigen die Laplace, Laplace+ $\mathcal{HC}$  und Taubin's  $\lambda$ - $\mu$  Glättung die geringste Rechenzeit, während die *Mean*- und *Median*-Glättung um den Faktor 10 langsamer sind. So brauchten die *Mean*- und *Median*-Glättung für  $\iota = 50$  Iterationen bereits etwas über 60 bzw. 80 Sekunden.

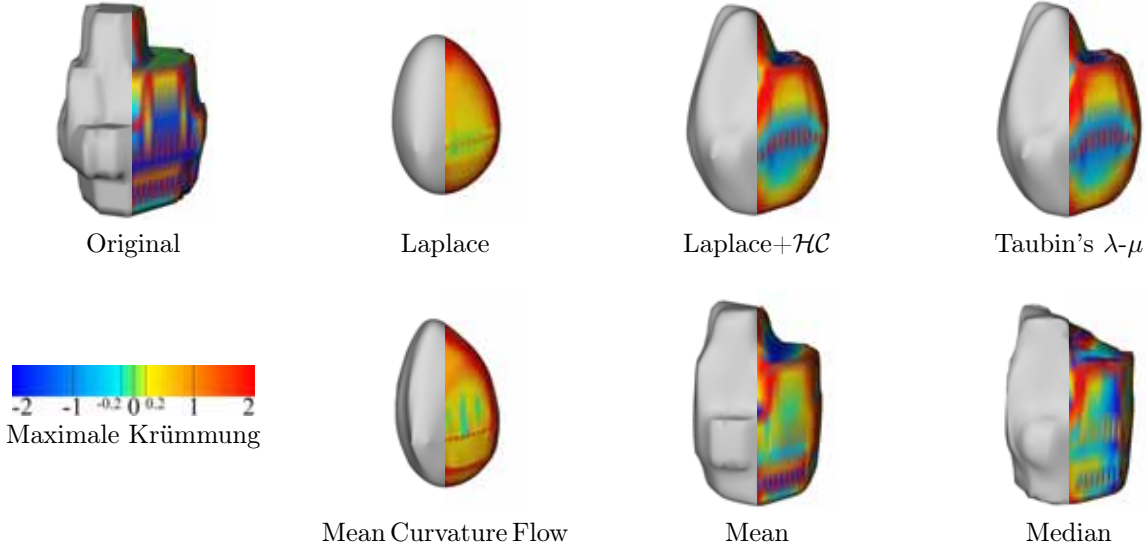
Zusammenfassend erwiesen sich zur Glättung von Oberflächenmodellen kompakter Strukturen Taubin's  $\lambda$ - $\mu$  Glättung und die Laplace+ $\mathcal{HC}$  Glättung jeweils mit einer Wichtung  $0,5 \leq \lambda \leq 0,9$  und Anzahl an Iterationen von  $20 \leq \iota \leq 50$  als angemessen. Beide Verfahren



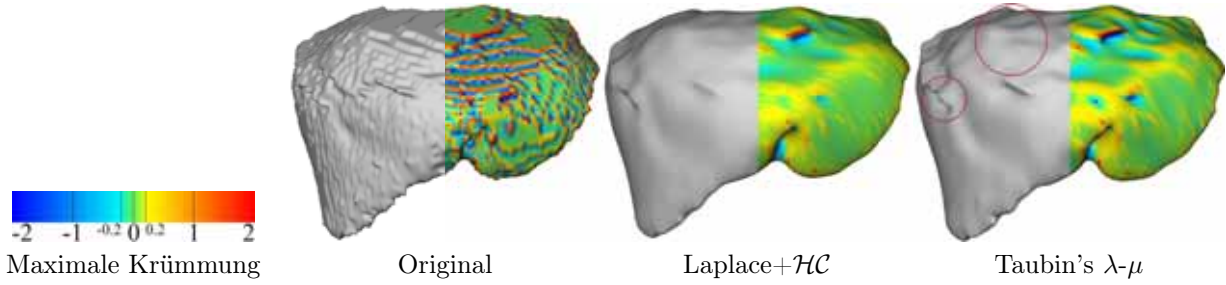
**Abbildung 5.12:** Symmetrische Hausdorff-Distanz ( $D_{sH}$ ) der Glättungsergebnisse der verschiedenen Filter zum initialen Oberflächenmodell der Leber (in Voxelkoordinaten) bei unterschiedlicher Iterationsanzahl  $l$  und Wichtung  $\lambda$ .



**Abbildung 5.13:** Glättungsergebnisse des Oberflächenmodells der Leber bei  $l = 50$  Iterationen und einer Wichtung mit  $\lambda = 0,9$ . (links: beleuchtete Oberfläche; rechts: maximale Krümmung  $k_1$ )



**Abbildung 5.14:** Glättungsergebnisse des Oberflächenmodells des Lymphknotens bei  $\iota = 50$  Iterationen und einer Wichtung mit  $\lambda = 0,9$ .



**Abbildung 5.15:** Reduktion der Terrassenartefakte im Lebermodell bei  $\iota = 400$  Iterationen und  $\lambda = 0,9$ . Das Glättungsergebnis von Taubin's  $\lambda$ - $\mu$  Glättung wird dabei durch die Instabilität des Verfahrens bei der gewählten Parameterkombination und der hohen Iterationsanzahl beeinträchtigt (Kreise).

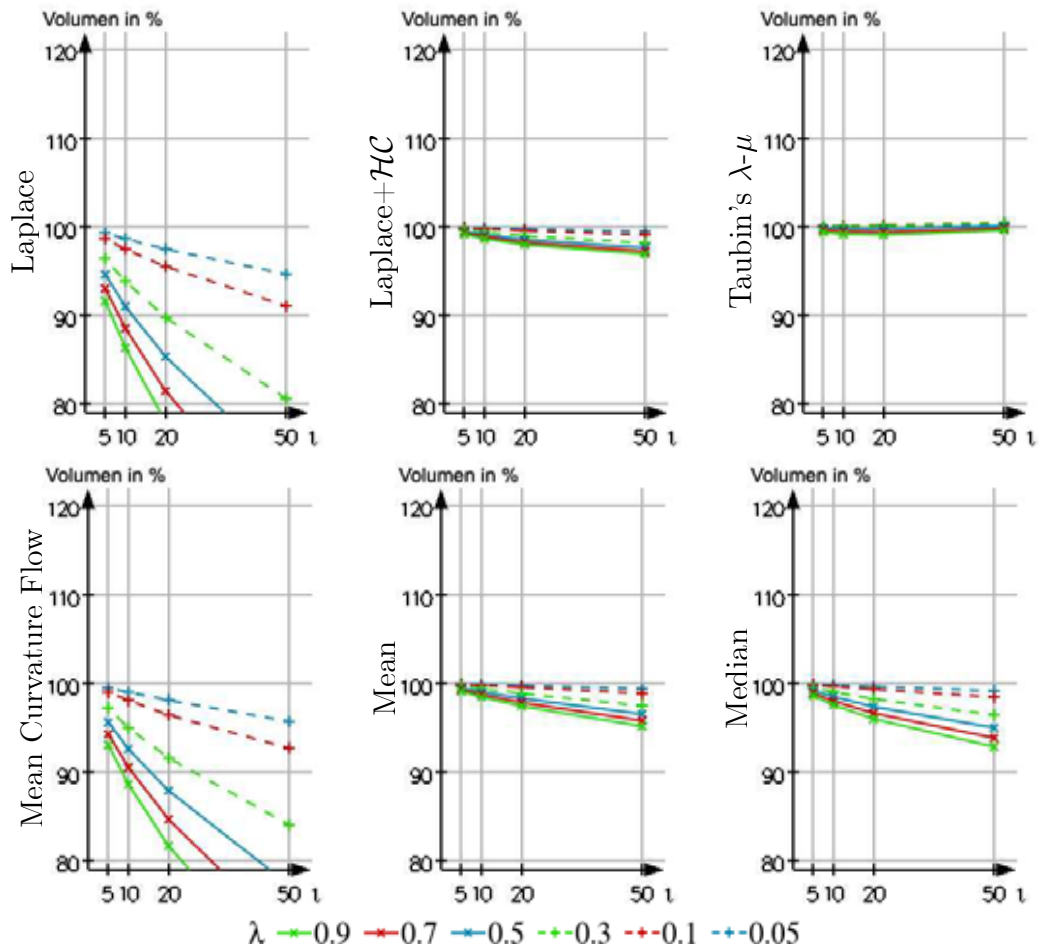
konnten dabei ähnlich visuell und durch die erhaltene Genauigkeit überzeugen. Starke Terrassenartefakte können jedoch nicht ganz zufrieden stellend geglättet werden. Durch weiteres Erhöhen der Iterationsanzahl kann dies zwar immer besser erreicht werden, jedoch erhöhen sich die Fehlermaße und es kann speziell bei Taubin's  $\lambda$ - $\mu$  Glättung zu einem instabilen Verhalten und unerwünschten Resultaten führen.

### 5.6.2 Glättungsergebnisse flacher Objekte

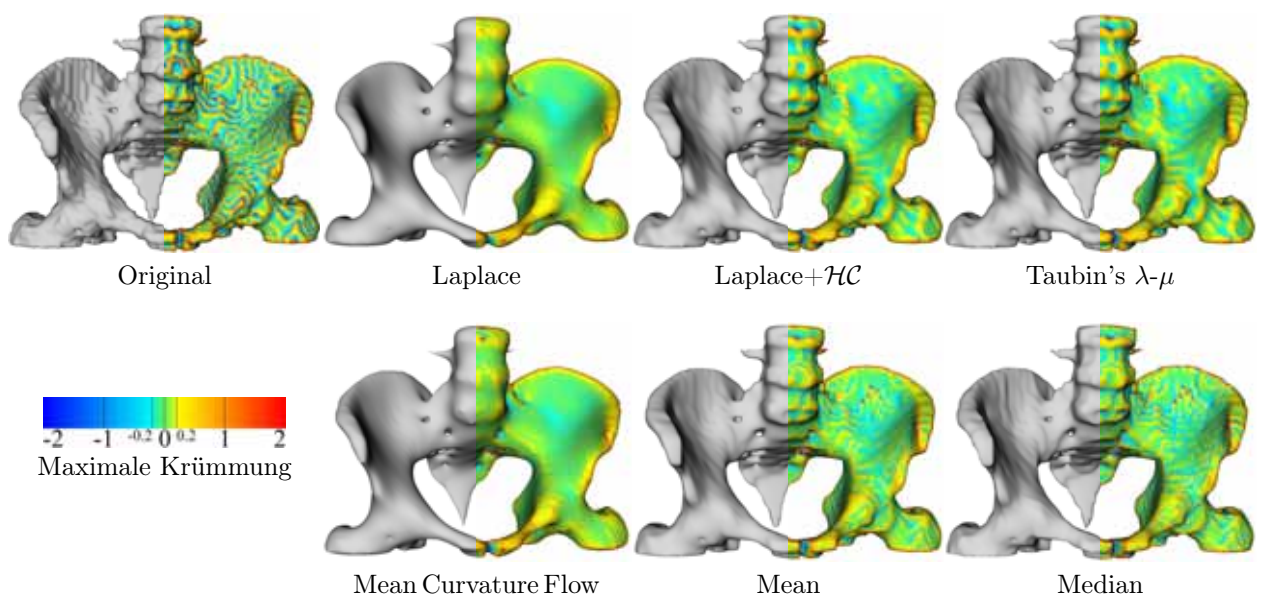
Die Glättung flacher Objekte erweist sich aufgrund größerer Objektverformung und Volumenverluste als komplexer als bei den kompakten Objekten. Abbildung 5.16 stellt den Volumenerhalt bei der Glättung des Halsmuskels mit den verschiedenen Glättungsverfahren dar. Die Laplace- und *Mean Curvature Flow* Glättung führen bei den flachen *high-res.* und *low-res.* Objekten zu Restvolumina  $V_{\%}$  weit unter 90 Prozent und entsprechend hohen Distanzwerten zum Ausgangsmodell. Abbildung 5.17 und 5.18 stellt die Glättungsergebnisse für die untersuchten Oberflächenmodelle bei  $\iota = 20$  Iterationen und Wichtung  $\lambda = 0,7$  gegenüber.

Artefakte im *high-res.* Oberflächenmodell des Beckens lassen sich mit der Laplace+ $\mathcal{HC}$ - und Taubin's  $\lambda$ - $\mu$ -Glättung mit geringen Distanzwerten ( $D_{sH} < 1$  Voxel) und sehr gutem Vo-

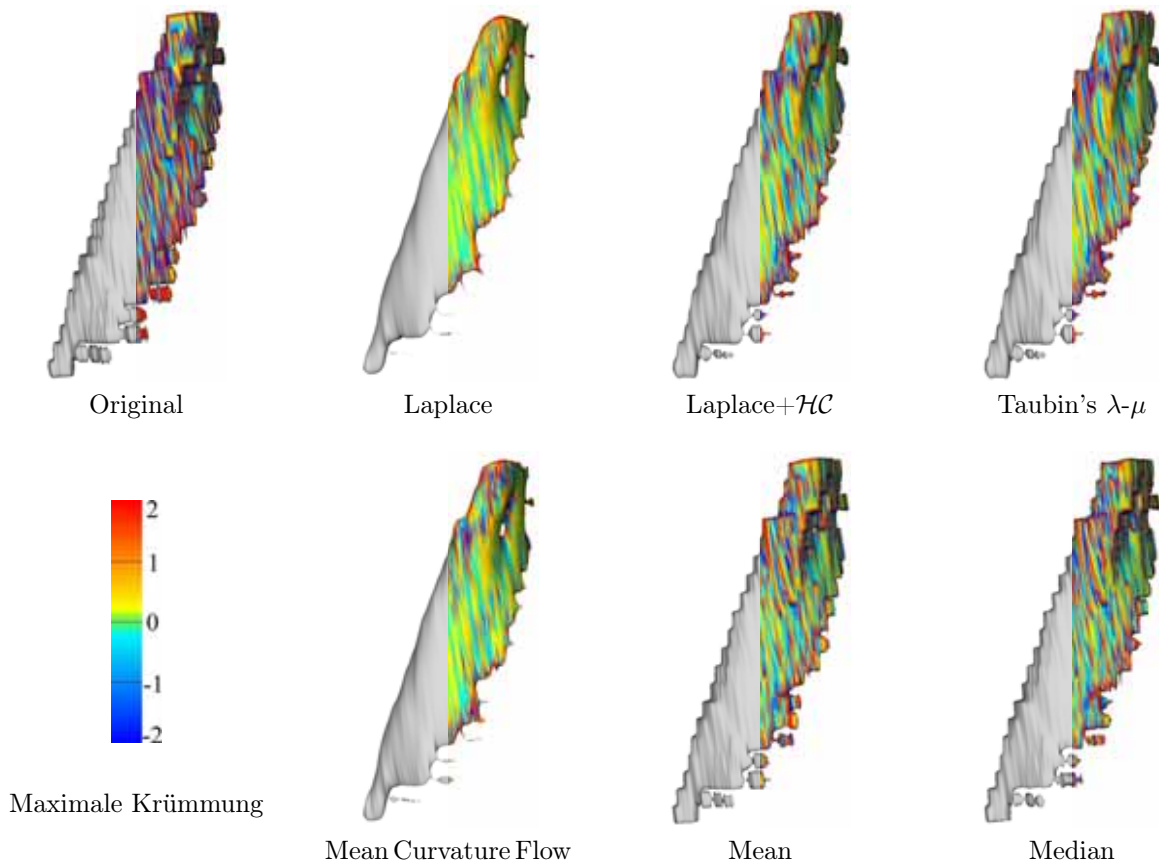




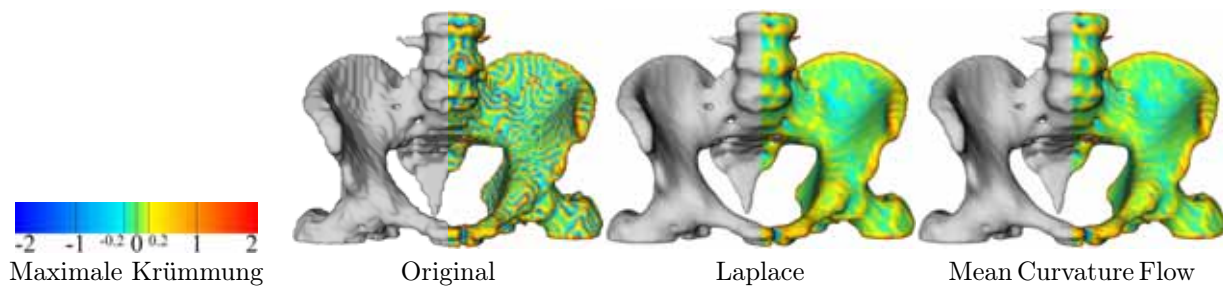
**Abbildung 5.16:** Volumenerhalt ( $V_{\%}$  in % des Originals) der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells des Halsmuskels mit unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .



**Abbildung 5.17:** Glättungsergebnisse des Oberflächenmodells des Beckens bei  $\iota = 20$  Iterationen und einer Wichtung mit  $\lambda = 0,7$ .



**Abbildung 5.18:** Glättungsergebnisse des Oberflächenmodells des Halsmuskels bei  $\iota = 20$  Iterationen und einer Wichtung mit  $\lambda = 0,7$ .



**Abbildung 5.19:** Vergleichbare Glättungsergebnisse des Oberflächenmodells des Beckens mit Laplace- und *Mean Curvature Flow* Glättung bei  $\iota = 10$  Iterationen und einer Wichtung mit  $\lambda = 0,3$  zu den Ergebnissen in Abbildung 5.17.

lumenerhalt  $V_{\%} = 100\% \pm 2\%$  reduzieren. Mit der *Mean*- und *Median*-Glättung sind visuell ähnliche Ergebnisse, aber mit größeren Distanzwerte  $n$  ( $D_{sH} \approx 2$  Voxel) erreichbar. Bei ähnlichen Fehlerwerten können beim Beckenmodell auch mit der Laplace- und *Mean Curvature Flow* Glättung bei  $\lambda = 0,3$  nach  $\iota = 10$  Iterationen ähnlich gute Glättungsergebnisse erzielt werden (siehe Abb. 5.19). Gründe hierfür sind die stärkere Glättungswirkung beider Verfahren bereits bei geringerer Iterationsanzahl  $\iota$  und geringerer Wichtung  $\lambda$  sowie das Fehlen großflächiger Terrassenartefakte, welche im Oberflächenmodell der Leber eine ausreichende Glättung durch die zwei Verfahren bei geringen Parameterwerten nicht möglich machten.

Das *low-res.* Oberflächenmodell des Halsmuskels lässt sich im Gegensatz zum *high-res.* Beckenmodell mit keinem der untersuchten Glättungsverfahren angemessen glätten. Die Laplace- und *Mean Curvature Flow* Glättung führen bereits bei geringer Parameterwahl zu großen Fehlerwerten (siehe Abb. 5.16). Bei moderaten Parameterwerten ( $\iota = 20$ ,  $\lambda = 0,7$ ) sind die Fehlerwerte der verbleibenden Verfahren ( $V_{\%} > 95\%$ ,  $D_{sH} \approx 3$ ) bereits deutlich schlechter als beim *high-res.* Beckenmodell, jedoch können dabei Block- und Terrassenartefakte, wie in Abbildung 5.18 dargestellt, nicht reduziert werden.

Zusammenfassend konnten die Artefakte im *high-res.* Beckenmodell mit allen Verfahren bei geringen Fehlerwerten gut reduziert werden. Die Glättungsverfahren sind jedoch nicht in der Lage, die Artefakte im *low-res.* Halsmuskellmodell zu reduzieren bzw. ermöglichen dies nur bei großen Fehlerwerten ( $V_{\%} \ll 90\%$ ,  $D_{sH} \gg 5$ ).

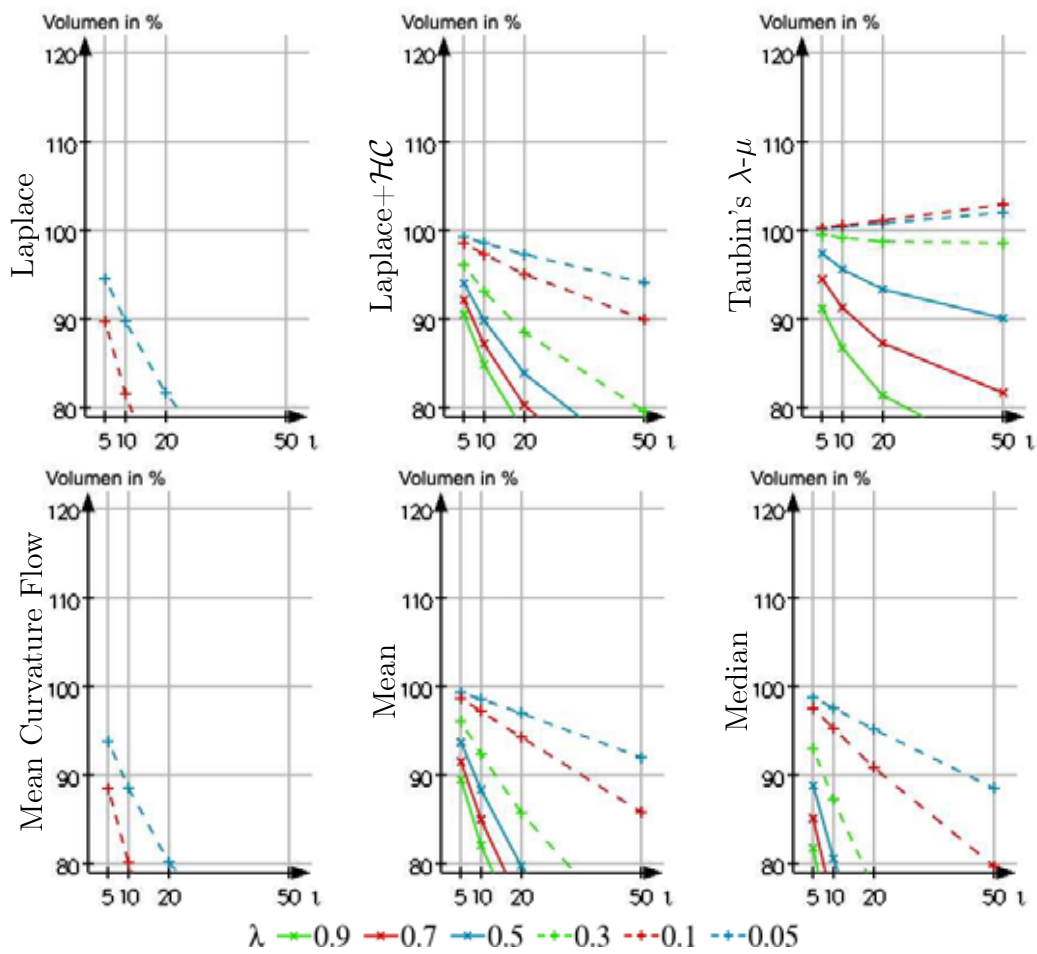
### 5.6.3 Glättungsergebnisse elongierter Objekte

Beim stark verzweigten Gefäßbaum und dem Oberflächenmodell der A. carotis als *low-res.* bzw. *high-res.* Vertreter der elongierten Objekte führen alle Filter zu einem sehr starken Volumenverlust und einem extremen Schrumpfen kleiner Objektteile und abgelöster Strukturen. Während beim Oberflächenmodell der A. carotis Taubin's  $\lambda$ - $\mu$  Glättung das Volumen mit  $V_{\%} = 100\% \pm 5\%$  am besten und die Laplace+ $\mathcal{HC}$  Glättung mit  $V_{\%} > 85\%$  noch weit besser als alle anderen Verfahren erhalten können, so ist der Volumenverlust beim Oberflächenmodell des Gefäßbaumes bei allen Verfahren dramatisch (siehe Abb. 5.20).

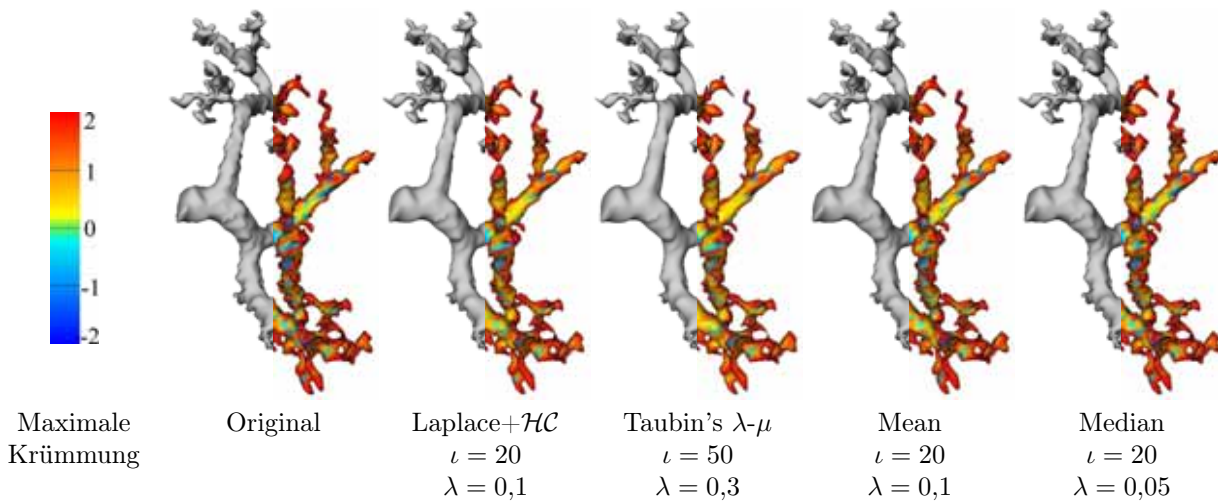
Für das Oberflächenmodell des Gefäßbaumes stellt Abbildung 5.21 die besten Glättungsergebnisse aller Verfahren mit einer maximalen symmetrischen Hausdorff-Distanz  $D_{sH} \leq 1$  Voxel und einem Restvolumen von  $V_{\%} \geq 95\%$  gegenüber. Die Laplace- und *Mean Curvature Flow* Glättung entfallen dabei, da diese bereits bei  $\iota = 5$  Iterationen und einer Wichtung von  $\lambda = 0,05$  zu größeren Abweichungen führen. Alle anderen untersuchten Verfahren können im gegebenen Fehlertoleranzbereich kleinere Artefakte mildern, jedoch nicht wesentlich reduzieren (siehe Abb. 5.21).

Wie in Abbildung 5.22 dargestellt, ist bei der Glättung des Oberflächenmodells der A. carotis im Gegensatz zum Gefäßbaum mit Taubin's  $\lambda$ - $\mu$  Glättung und etwas geringer auch mit der Laplace+ $\mathcal{HC}$  Glättung zumindest eine moderate Reduktion der Stufenartefakte bei gutem Volumenerhalt ( $V_{\%} \geq 95\%$ ) möglich.

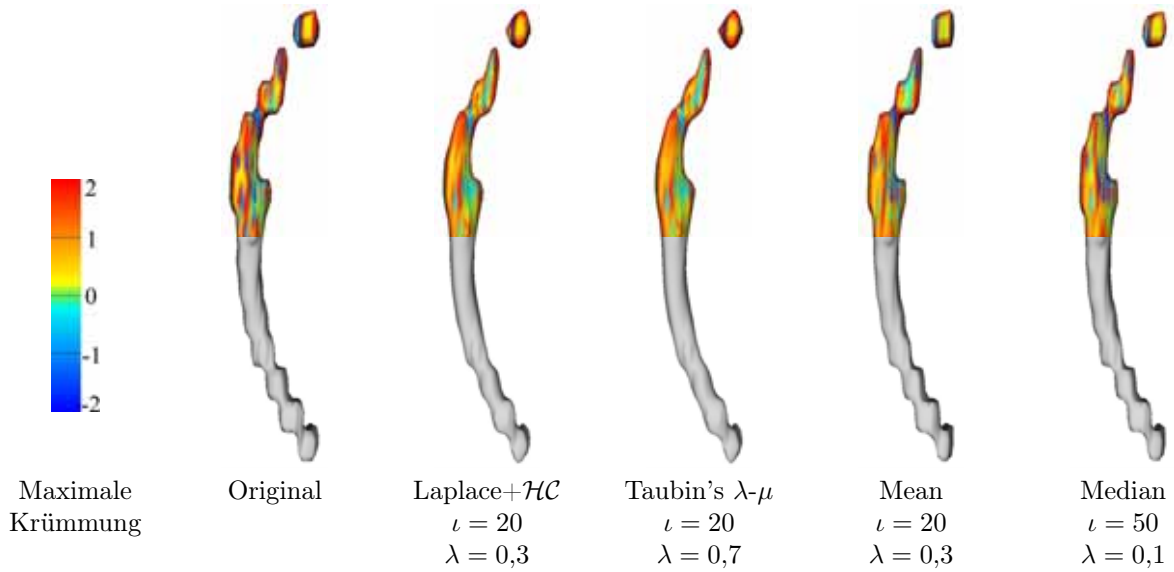
Zusammenfassend sind die Laplace- und *Mean Curvature Flow* Glättung für die untersuchten elongierten Objekte als absolut ungeeignet einzustufen. Gute Glättungsergebnisse lassen sich nur mit Taubin's  $\lambda$ - $\mu$  Glättung und der Laplace+ $\mathcal{HC}$  Glättung beim *high-res.* Oberflächenmodell der A. carotis erzielen. Die Glättung des stark verzweigten Gefäßbaumes (mit vielen kleinen Ästen und abgelösten Elementen) hingegen ist mit keinem der Verfahren erfolgversprechend.



**Abbildung 5.20:** Volumenerhalt ( $V_{\%}$  in % des Originals) der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells des Gefäßbaumes mit unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .



**Abbildung 5.21:** Beste Glättungsergebnisse des Oberflächenmodells des Gefäßbaumes bei maximaler symmetrischer Hausdorff-Distanz  $D_{sH} \leq 1$  und einem Restvolumen von  $V_{\%} \geq 95\%$ .



**Abbildung 5.22:** Beste Glättungsergebnisse des Oberflächenmodells der A. carotis bei maximaler symmetrischer Hausdorff-Distanz  $D_{sH} \leq 2$  und einem Restvolumen von  $V_{\%} \geq 95\%$ .

## 5.7 Glättungsempfehlungen

Die vergleichende Evaluation der Glättungsverfahren zeigt, dass mit der Laplace- und *Mean Curvature Flow* Glättung die größten Volumenverluste und Abweichungen zu den Ausgangsmodellen auftreten, was die Verfahren für medizinische Oberflächenmodelle ungeeignet macht. Bei der *Mean*- und *Median*-Glättung treten nicht so starke Abweichungen auf, jedoch neigen die Verfahren dazu, scharfe Kanten und viereckige Formen zu erzeugen, wodurch medizinische Oberflächenmodelle in ihrer Form meist stark verfremdet und Block- und Terrassenartefakte nicht reduziert werden. Den besten Volumenerhalt bei den gleichzeitig geringsten Distanzwerten erreichen die Laplace+ $\mathcal{HC}$ - und Taubin's  $\lambda$ - $\mu$  Glättung. Mit Taubin's  $\lambda$ - $\mu$  Glättung sind dabei meist die geringsten Distanzwerte zu erreichen, was die im Folgenden abgeleiteten Glättungsempfehlungen beeinflusst.

**Kompakte Objekte:** Für die Glättung von kompakten Objekten sollte Taubin's  $\lambda$ - $\mu$ -Glättung mit  $20 \leq \iota \leq 50$  Iterationen und einem Wichtungsfaktor  $0,5 \leq \lambda \leq 0,9$  verwendet werden. Alternativ kann auch die Laplace+ $\mathcal{HC}$ -Glättung mit gleichen Parametern verwendet werden. Bei starken Terrassenartefakten können höhere Iterationszahlen verwendet werden, wobei dies bei Taubin's  $\lambda$ - $\mu$ -Glättung zu instabilen Filtern führen kann. In [Bade et al. 2006a] ist auch die Untersuchung der topologischen Nachbarschaft zweiten Grades aufgeführt. Diese zeigt, dass überwiegend durch die Verwendung der topologischen Nachbarschaft zweiten Grades das gleiche Glättungsergebnis, wie bei einfacher Nachbarschaft und erhöhter Iterationszahl möglich ist. Bei Taubin's  $\lambda$ - $\mu$ -Glättung kann dies sogar mit leicht geringeren Fehlerwerten bei kompakten Objekten möglich sein. Dieser Vorteil der größeren Nachbarschaft kann darauf zurückgeführt werden, dass im ersten Schritt von Taubin's  $\lambda$ - $\mu$ -Glättung die geglätteten Oberflächenpunkte über eine größere Fläche gemittelt werden und somit die durch anfängliche Instabilität verstärkten hohen Frequenzen implizit unterdrückt werden. Entsprechend kann die topologische Nachbarschaft zweiten Grades bei starken Terrassenartefakten statt einer erhöhten Iterationszahl eingesetzt werden.

**Flache Objekte:** Für die flachen *high-res.* Objekte sollte ebenfalls Taubin's  $\lambda$ - $\mu$ -Glättung, allerdings bei geringerer Iterationsanzahl  $\iota \leq 20$ , verwendet werden. Die Laplace+ $\mathcal{HC}$ -Glättung ist dabei Taubin's  $\lambda$ - $\mu$ -Glättung sehr ähnlich, führt aber meist zu einem leicht stärkeren Volumenverlust. Beide Verfahren sind dabei der *Mean*- und *Median*-Glättung vorzuziehen, da diese wiederum dazu neigen, Kanten im Modell zu erzeugen. Für die Glättung von flachen *low-res.* Objekten bietet sich, aufgrund des großen Volumenverlustes und der hohen Distanzwerte bei den anderen Verfahren, wiederum Taubin's  $\lambda$ - $\mu$ -Glättung an. Die Reduktion größerer Block- und Terrassenartefakte ist bei akzeptablen Distanzwerten hier jedoch nicht möglich.

**Elongierte Objekte:** Für die Glättung relativ gleichförmiger, kaum oder gar nicht verzweigter elongierter *high-res.* Objekte kann wiederum Taubin's  $\lambda$ - $\mu$ -Glättung mit  $\iota = 20$  Iterationen und einer Wichtung  $0,5 \leq \lambda \leq 0,7$  verwendet werden. Elongierte Objekte mit vielen Verzweigungen und kleinen Ästen können mit keinem der Verfahren angemessen geglättet werden. Für die Glättung von Gefäßbäumen ist daher die Glättung auf der Oberflächennetzebene ungeeignet. Aufgrund der Relevanz von Blutgefäßen in der chirurgischen Ausbildung und Interventionsplanung wird in Kapitel 7 die qualitativ hochwertige Darstellung elongierter Strukturen gesondert behandelt.

## 5.8 Diskussion und Fazit

Die Klassifizierung von Oberflächenmodellen in kompakte, flache und elongierte Objekte für die Untersuchung erwies sich grundlegend als sinnvolle und intuitive Unterteilung. Sie ermöglicht es, grob Aussagen über das Glättungsverhalten, zu erwartende Fehler- und Abstandswerte der Glättungsergebnisse für beliebige medizinische Modelle abzuschätzen. Die Untersuchung zeigt aber auch, dass im Detail die Art der vorhandenen Artefakte (Block- bzw. Terrassenartefakte), das Krümmungsverhalten und das Verhältnis von Dreiecksgröße zu Featuregröße in den Oberflächennetzen die erreichbaren Glättungsergebnisse stark beeinflussen. Sind keine Terrassenartefakte vorhanden, so können *high-res.* Objekte (mit kleiner Dreiecksgröße zu großer Featuregröße) leichter mit allen untersuchten Verfahren geglättet werden. Hingegen können Objekte mit starken Krümmungswerten bei gleichzeitig geringer Featuregröße (*low-res.*), wie dem Blutgefäßbaum und dem Halsmuskel, mit keinem der Glättungsverfahren adäquat geglättet werden. Weiterhin stellt die vorgenommene Objektklassifizierung nur eine grobe Richtlinie dar, da meist lokal sogar sehr unterschiedliche Verhältnisse von Krümmungswerten und Dreiecks- zu Featuregröße über die Oberflächenmodelle verteilt sein können.

Ferner zeigt die Untersuchung, dass sich das Volumen kompakter Oberflächenmodelle unter der Glättung nur unmerklich ändert, während die maximale symmetrische Hausdorff-Distanz durchaus rapide ansteigen kann. Umgekehrt verhält es sich hingegen bei elongierten Objekten, bei denen die Hausdorff-Distanz geringe Änderungen zeigt, diese geringen Abweichungen aber bereits einen großen Anteil des Gesamtvolumens ausmachen und sich entsprechend das Volumen dramatisch verändert. Folglich ist die Beurteilung des Fehlers des geglätteten Modells zum Original über die symmetrische Hausdorff-Distanz als Abstandsmaß und gleichzeitig über das Restvolumen in Prozent vom Originalvolumen nützlich.

Zur Evaluation der Glättungsverfahren wurde die Volumenskalierung von Desbrun et al. [1999] (siehe Abschnitt 5.4.1, S. 66) ausgeschlossen, da diese Objekte verzerrt und ungeeignet im Raum verschiebt. Kürzlich stellten Eckstein et al. [2007] eine neue Variante zum Erhalt

des Volumens vor. Die vorgestellte Methode kann grundsätzlich für alle Glättungsverfahren eingesetzt werden und könnte somit die volumenerhaltenden Eigenschaften aller Verfahren verbessern. Eine Untersuchung der erreichbaren Glättung und Genauigkeit der Glättungsverfahren mit dieser neuen Methode zum Volumenerhalt stehen noch aus.

In Bezug auf die Glättungsergebnisse zeigt sich, dass eine ausreichende Reduktion von Block- und Terrassenartefakten meist nur bei Abweichungen über einem Voxel zum Ausgangsobjekt möglich ist. Bereits bei Abweichungen über einem halben Voxel führt dies dazu, dass die geglätteten Oberflächenmodelle nicht mehr korrekt die Grenzfläche zwischen Objekt-Voxeln und als Hintergrund markierten Voxeln darstellen. Entsprechend können sich nach der Glättung Hintergrundvoxel innerhalb und Objektvoxel außerhalb des Oberflächenmodells befinden. Folglich stellt das geglättete Oberflächenmodell keine korrekte Innen-Außen-Repräsentationen der Originaldaten mehr dar. Zusammenfassend ist weiterhin die Glättung von filigranen und stark verzweigten elongierten Objekten (Blutgefäße) nicht möglich. Im folgenden Kapitel werden deshalb Lösungsmöglichkeiten für die erkannten Probleme entwickelt.





# Voxelbegrenzte Oberflächennetzglättung

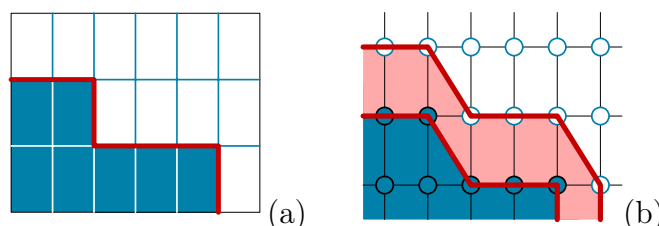
Wie die Evaluation der Glättungsverfahren für medizinische Oberflächenmodelle (Kapitel 5) gezeigt hat, führen Verfahren zur Glättung von Oberflächennetzen zu inkorrekten Innen-Außen Repräsentationen. Auf der reinen Oberflächennetzebene, auf der die vorgestellten Glättungsverfahren arbeiten, kann die Korrektheit der Innen-Außen Repräsentationen der Glättungsergebnisse nicht garantiert werden. Hierzu sind Informationen zur korrekten Innen-Außen Abgrenzung aus der Bildebene erforderlich. Im Folgenden wird deshalb ein Verfahren vorgestellt, welches bei der Glättung von Oberflächennetzen die Oberflächennetzglättung auf den Voxelbereich beschränkt, der eine korrekte Innen-Außen Repräsentation garantiert. Die nötigen Informationen aus der Bildebene können dabei direkt aus der Bildebene übernommen oder aus dem bereits extrahierten Oberflächenmodell gewonnen werden. Letzteres ermöglicht dabei die Beschränkung der Oberflächennetzglättung auf den korrekten Innen-Außen Voxelbereich auch ohne Kenntnisse über die originalen Bilddaten.

## 6.1 Vorbetrachtungen

Um eine korrekte Innen-Außen Repräsentation auch nach der Glättung von Oberflächenmodellen garantieren zu können, bietet sich die Begrenzung der Glättung auf den Voxelbereich an, der die Grenze zwischen Hintergrund und Objektvoxeln repräsentiert. Dazu definieren wir, dass eine Oberfläche dann eine korrekte Innen-Außen Repräsentation binär klassifizierter Voxeldaten darstellt, sofern sich der Mittelpunkt aller Hintergrundvoxel außerhalb bzw. direkt auf der Oberfläche und der Mittelpunkt aller Objektvoxel innerhalb bzw. direkt auf der Oberfläche befinden. Damit wird ein zulässiger Toleranzbereich für eine korrekte Innen-Außen abgrenzende Oberfläche, wie in Abbildung 6.1 dargestellt, durch den Bereich zwischen den Mittelpunkten von Objektvoxeln und von in ihrer direkten Nachbarschaft befindlichen Hintergrundvoxeln aufgespannt.

Um bei der Glättung von Oberflächenmodellen eine korrekte Innen-Außen Repräsentation zu erhalten, muss die Bewegung der Oberflächenpunkte unter der Glättung auf den zulässigen Toleranzbereich beschränkt werden. Entsprechende Begrenzungen können als sogenannte *Constraints* in den Glättungsprozess eingebunden werden.

Dieser Strategie entsprechend wird in dieser Arbeit die *Diamond Constrained* Glättung von Oberflächenmodellen eingeführt. Das vorgestellte Verfahren kann dabei als Übertragung des Ansatzes der *Constrained Elastic Surface Nets* CESN (siehe Abschnitt 4.2.4) auf das



**Abbildung 6.1:** Binäre Segmentierungsmaske (a) und zugehöriger Toleranzbereich für eine korrekte Innen-Außen Repräsentation (b). Alle im Toleranzbereich definierten Oberflächen grenzen korrekt Objekt und Hintergrund bzw. Innen und Außen des Objektes ab.

Standardverfahren zur Oberflächenextraktion *Marching Cubes* MC (Abschnitt 4.2) verstanden werden. Im Gegensatz zum CESN-Ansatz kann dadurch auf der einen Seite auf ein spezielles Verfahren zur Oberflächenextraktion verzichtet werden und auf der anderen Seite wird es möglich, auch im Nachhinein standardmäßig mittels *Marching Cubes* MC extrahierte Oberflächen (auch ohne Kenntnis der Bilddaten) so zu glätten, dass korrekte Innen-Außen Repräsentationen garantiert werden können.

Zunächst werden dazu in Abschnitt 6.2 verschiedene Möglichkeiten, die Bewegung von Oberflächenpunkten bei der Glättung zu beschränken, betrachtet und die Verwendung diamantenähnlicher Formen als Beschränkung (*Diamond Constraint*) hergeleitet. Danach wird in Abschnitt 6.3 die Umsetzung und Implementierung der Diamond Constraint Glättung als Erweiterung des Glättungsmoduls aus Abschnitt 5.5 beschrieben. Daraufhin werden Ergebnisse der Glättung mit *Diamond Constraint* in Abschnitt 6.4 präsentiert und mit den Ergebnissen verwandter Arbeiten verglichen. Abschließend werden die erreichten Ergebnisse in Abschnitt 6.5 zusammengefasst und diskutiert.

## 6.2 Begrenzung der Glättungsbewegung

In diesem Abschnitt werden verschiedene Strategien zur Begrenzung der Bewegung von Oberflächenpunkten bei der Glättung vorgestellt. Zum Verständnis sei aus Abschnitt 4.2 noch einmal kurz zusammengefasst, dass zur Extraktion von Oberflächen aus Volumendaten mit MC und anderen zellbasierten Verfahren (z.B. CESN und DualMC) immer acht Voxel (jeweils vier aus zwei benachbarten Schichten) zu Zellen zusammengefasst werden. In diesen Zellen wird dann anhand der Markierung der Voxel (Hintergrund vs. Objektvoxel) entschieden, ob und wo Oberflächenpunkte erzeugt werden. Die Größe der Zellen in jeder Dimension ergibt sich dabei aus dem Abstand zwischen den Voxeln in der jeweiligen Dimension ( $\delta x, \delta y, \delta z$ ).

### 6.2.1 Cell-Size Constraint

Im CESN-Ansatz werden bei der Oberflächenextraktion ausschließlich Oberflächenpunkte in den Zellmittelpunkten aller Zellen, die zur Oberfläche beitragen<sup>1</sup>, erzeugt. Bei der anschließenden Glättung wird die Bewegung der Oberflächenpunkte entsprechend auf plus/minus die halbe Zellgröße in jede Dimension  $\pm \frac{1}{2}(\delta x, \delta y, \delta z)$  beschränkt (*Cell-Size Constraint*). Bei der Oberflächenextraktion mittels CESN wird dadurch erreicht, dass kein Oberflächenpunkt seine Zelle verlässt und eine korrekte Innen-Außen Repräsentation erzeugt wird.

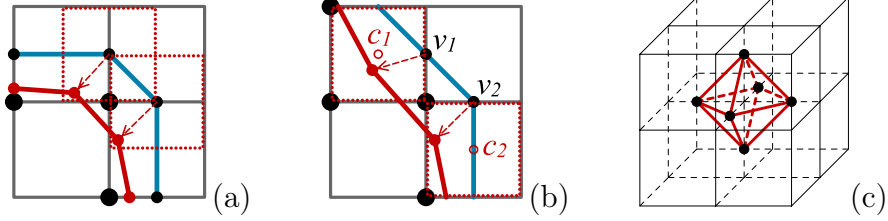
Für die Oberflächenextraktion mittels MC trifft dies jedoch nicht zu. Da beim MC-Verfahren Oberflächenpunkte nur auf den Zellgrenzen erzeugt werden, kann die Begrenzung  $\pm \frac{1}{2}(\delta x, \delta y, \delta z)$  der Zellgröße, wie in Abbildung 6.2 (a), keine korrekte Innen-Außen Repräsentation garantieren. Jedoch bietet dieser Ansatz bereits eine definierte Fehlerschranke von einem halben Voxel zum Ausgangsmodell, welche von allen in Kapitel 5 zuvor evaluierten Glättungsverfahren deutlich überschritten wird.

### 6.2.2 Cell-Center Constraint

Beim DualMC-Ansatz werden bei der Oberflächenextraktion mehrere Oberflächenpunkte innerhalb einer Zelle erzeugt. Bei der anschließenden Glättung wird die Bewegung der Ober-

---

<sup>1</sup>Zellen die mindestens einen als Objektvoxel und einen als Hintergrund markierten Voxel enthalten (vergleiche Seite 44 ff).



**Abbildung 6.2:** Inkorrekte Innen-Außen Repräsentationen mit *Cell-Size* und *Cell-Center Constraint* bei mittels Marching Cubes (MC) extrahierten Oberflächen: (a) Begrenzte Bewegung der Oberflächenpunkte auf  $\pm\frac{1}{2}(\delta x, \delta y, \delta z)$  (*Cell-Size Constraint*); (b) Begrenzte Bewegung der Oberflächenpunkte  $v_1$  und  $v_2$  in Relation zum jeweiligen Zellzentrum  $c_1$  bzw.  $c_2$  (*Cell-Center Constraint*); (c) 3D-Fall bei dem sechs Oberflächenpunkte in Relation zu acht Zellen beschränkt werden müssten.

flächenpunkte relativ zum Zentrum der Zelle, in der sie erzeugt wurden, begrenzt. Dabei gilt, dass ein Oberflächenpunkt nach der Glättung eine Position mit einem maximalen Abstand von  $\pm\frac{1}{2}(\delta x, \delta y, \delta z)$  zum Zentrum seiner Zelle annehmen darf (*Cell-Center Constraint*). Dies stellt die Verallgemeinerung des *Cell-Size Constraint* auf beliebig verteilte Punkte in definierten Zellen dar.

Bei mittels MC erzeugten Oberflächen befindet sich jeder Oberflächenpunkt jedoch direkt auf der Zellgrenze zwischen vier benachbarten Zellen, wodurch die korrekte Wahl der richtigen Zelle und damit des richtigen Zellzentrums für diese Art der Glättungsbegrenzung unmöglich wird [Bade et al. 2007a]. Aus Abbildung 6.2 (b) ist dabei ersichtlich, wie die Wahl einer beliebigen der möglichen Zellen zu inkorrekten Innen-Außen Repräsentationen führen kann. In Abbildung 6.2 (c) ist darüber hinaus ein simpler Fall dargestellt, bei dem die Wahl der richtigen Zelle für jeden Oberflächenpunkt im dreidimensionalen Raum unmöglich ist. Entsprechend kann die *Cell-Center Constraint* Methode bei mit MC erzeugten Oberflächen nicht zu korrekten Innen-Außen Repräsentationen beitragen.

### 6.2.3 Cell-Edge Constraint

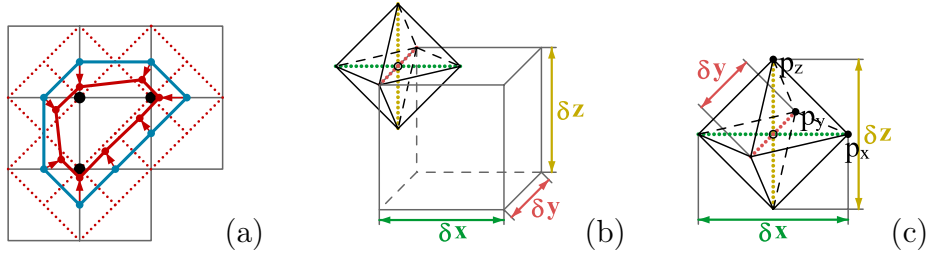
Bei der MC-Extraktion von Oberflächen befindet sich jeder Oberflächenpunkt direkt auf der Zellgrenze zwischen vier benachbarten Zellen. Um korrekte Innen-Außen Repräsentationen auch nach dem Glätten der Oberflächenpunkte garantieren zu können, kann die Bewegung der Oberflächenpunkte unter der Glättung entlang der Zellgrenze beschränkt werden (*Cell-Edge Constraint*). Hierbei kann sich jeder Oberflächenpunkt nur in Richtung der Zellgrenze maximal um eine halbe Länge der Zelle vor und zurück bewegen.

Durch das *Cell-Edge Constraint* bleibt jeder Oberflächenpunkt auf der Zellgrenze, auf der er erzeugt wurde. Dieses Vorgehen bevorzugt übermäßig stark die Korrektheit und Unversehrtheit des Ausgangsmodells gegenüber der Glättung. Dadurch kann eine korrekte Innen-Außen Repräsentation garantiert werden, jedoch ist die erreichbare Glättung sehr begrenzt.

### 6.2.4 Diamond Constraint

Da die Glättung mit *Cell-Size* und *Cell-Center Constraint* für MC-extrahierte Oberflächen keine korrekten Ergebnisse garantieren kann und eine Glättung mit *Cell-Edge Constraint* zu keiner ausreichenden Glättung führt, wird die Glättung mit einem neuen *Diamond Constraint* eingeführt.

Wie in Abbildung 6.3 (a) für den zweidimensionalen Raum dargestellt, können die mittels MC auf Zellgrenzen extrahierten Oberflächenpunkte beliebig innerhalb eines an den



**Abbildung 6.3:** Das *Diamond Constraint* beschränkt die Bewegung von Oberflächenpunkten bei der Glättung in 2D auf einen Rhomboid (a) und in 3D auf einen diamantenförmigen Bereich (b, c), welche sich über die Zellgröße aufspannen.

Zellgrenzen zentrierten und auf die Spitze gedrehten Rechtecks bzw. allgemeiner Rhomboids bewegt werden, ohne inkorrekte Innen-Außen Repräsentationen zu erzeugen. Im 3D-Raum ist dies mit gleichen gutartigen Eigenschaften auf einen diamantenförmigen Bereich mit acht Facetten (siehe Abb. 6.3 (b, c)) übertragbar. Dieser, vereinfacht *Diamant* genannte Bereich liegt zentriert auf den Zellgrenzen und hat dieselben Ausmaße ( $\delta x, \delta y, \delta z$ ) wie die Zellen bei der Oberflächenextraktion.

Ein Vorteil, der sich durch die Begrenzung der Bewegung auf Diamanten im Vergleich zur alleinigen Begrenzung auf die Zellgrenzen (*Cell-Edge Constraint*) ergibt, besteht darin, dass die Diamanten auf benachbarten Zellgrenzen den vorhandenen Zwischenraum lückenlos ausfüllen. Dadurch steht während der Glättung mit *Diamond Constraint* ein viel größerer Raum für zulässige Bewegungen der Oberflächenpunkte zur Verfügung, wodurch eine bessere Glättung des Ausgangsmodells erreicht werden kann.

Ein weiterer Vorteil gegenüber der Glättung mit *Cell-Edge Constraint* ist, dass für die Glättung mit *Diamond Constraint* von Oberflächenmodellen, die aus binär klassifizierten Volumendaten extrahiert wurden, die Information über die Zellgröße bzw. Voxelgröße in den Volumendaten ausreichend ist, um die Diamanten für die Glättung mit *Diamond Constraint* exakt bestimmen zu können.

## 6.3 Umsetzung und Implementierung

Für die Glättung mit *Constraints* zur Begrenzung der Bewegung von Oberflächenpunkten wird das in Abschnitt 5.5 entwickelte Modul zur Glättung von Oberflächenmodellen um ein Submodul für die *Constraints* erweitert. Das *Constraints* Submodul wird dabei nach bzw. alternativ zum Submodul für Korrekturverfahren in die Iterationsschleife des Glättungsmoduls eingebunden.

### 6.3.1 Constraints Submodul

Alle zuvor in Abschnitt 6.2 vorgestellten Verfahren zur Begrenzung der Glättungsbewegung wurden als *Constraint* Submodule, wie in Abbildung 6.4 dargestellt, umgesetzt. Standardmäßig ermitteln diese selbstständig aus dem gegebenen Ausgangsmodell  $\mathcal{M}$ , wie in [Bade et al. 2007a] beschrieben, die für die *Constraint* nötigen Informationen (z.B. Zellgröße). Alternativ können die Informationen auch aus den originalen Volumendaten ermittelt oder vom Anwender als Parameter eingegeben werden. Die Bestimmung der erforderlichen Informationen direkt aus dem Ausgangsmodell macht das Verfahren von den originalen Bilddaten

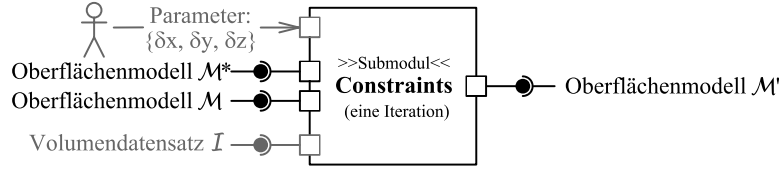


Abbildung 6.4: UML-Diagramm des Submoduls für die Glättung mit *Constraints*.

unabhängig und kann dabei auf der reinen Oberflächennetzebene (vergleiche Abb. 4.1, S. 41) bleiben und dennoch voxelgenau arbeiten.

Pro Glättungsiteration werden dann im *Constraint* Submodul die neuen Positionen der Oberflächenpunkte nach der Glättung  $v^* \in \mathcal{M}^*$  in Bezug auf das gewählte *Constraint* überprüft und gegebenenfalls auf eine Position im zugelassenen Bereich zurückgesetzt. Beim einfachen *Cell-Size* und *Cell-Center Constraint* wird dazu der Verschiebungsvektor  $\vec{dv}$  zwischen der neuen Position  $v^*$  und der initialen Ausgangsposition  $o$  bzw. dem gegebenen Zellzentrum  $c$  berechnet ( $\vec{dv} = c - v^*$ ). Daraufhin werden die einzelnen Komponenten des Verschiebungsvektors ( $\vec{dv}.x$ ,  $\vec{dv}.y$ ,  $\vec{dv}.z$ ) auf die Zellgröße in die entsprechende Raumrichtung abgeschnitten (geclippt), sofern sie diese überragen. Entsprechend ergibt sich dann die durch das *Constraint* begrenzte neue Position  $v'$  über  $v' = c + G(\vec{dv})$ , wobei  $G()$  die Clippingfunktion darstellt. Im Folgenden wird dies für das *Diamond Constraint* detailliert dargestellt.

### 6.3.2 Umsetzung des Diamond Constraint

Die Ausdehnung des Diamanten für das *Diamond Constraint* ist über die Zellgröße in alle Raumrichtungen ( $\delta x$ ,  $\delta y$ ,  $\delta z$ ) bekannt. Damit kann für jede der acht Facetten des Diamanten die Ebenengleichung der Form  $ax + by + cz + D = 0$  berechnet werden. Dies macht es möglich, jede Facette als Ebene mit Normalenvektor  $\vec{n} = (a, b, c)$  und dem Abstand  $D$  zum Zentrum des Diamanten zu behandeln. Dabei ist der Abstand  $D$  der Facetten vom Diamantenzentrum für alle der gleiche.

Da die Facetten relativ zum Zentrum des Diamanten für alle Oberflächenpunkte in einem Oberflächenmodell konstant sind und sich auch nicht während der Glättung verändern, können für die acht Facetten die acht Normalen  $\vec{n}_1 \dots \vec{n}_8$  und ein  $D$  einmal vorberechnet und über die gesamte Glättung wiederverwendet werden. Das Zentrum der Diamanten für jeden Oberflächenpunkt bildet dabei jeweils die originale Position  $o$  des jeweiligen Oberflächenpunktes im Ausgangsmodell [Bade et al. 2007a].

Während der Glättung kann dann der Verschiebungsvektor  $\vec{dv}$  eines geglätteten Oberflächenpunktes  $v^*$  zum Zentrum des Diamanten  $o$  über  $\vec{dv} = o - v^*$  berechnet werden. Der Verschiebungsvektor muss dann an den Facetten des Diamanten abgeschnitten bzw. geclippt werden, um die neue, das Constraint erfüllende Position nach der Glättung  $v'$  zu finden.

Um effizient den Verschiebungsvektor nur mit der Facette zu clippen, in dessen Richtung der Verschiebungsvektor zeigt, genügt eine simple Überprüfung der Vorzeichen der drei Komponenten des Verschiebungsvektors. Daraufhin wird der Schnittpunkt des Verschiebungsvektors mit der entsprechenden Facette berechnet. Sofern ein Schnittpunkt existiert, wird dieser als neue Position  $v'$  für den behandelten Oberflächenpunkt verwendet. Listing 6.1 fasst das beschriebene Vorgehen noch einmal zusammen.

**Listing 6.1:** Pseudocode für das Vorgehen beim *Diamond Constraint*


---

```

% Vorbereitung der Facetten des Diamanten -----
% Normale von Facette 8
 $\vec{n}_8 \leftarrow (p_y - p_x) \otimes (p_z - p_x)$  %  $p_x, p_y, p_z$  siehe Abbildung 6.3(c)
 $\vec{n}_8 \leftarrow \vec{n}_8 / \|\vec{n}_8\|$  % Normalisierung
% Normale von Facette 1 % Facette 1 liegt Facette 8 gegenüber  $\Rightarrow$ 
 $\vec{n}_1 \leftarrow -\vec{n}_8$  % Facette 1 hat inverse Normale
% Berechnung aller weiteren Normalen ist analog
...
% Berechnung des Abstandes der Facetten zum Zentrum des Diamanten
% allgemein gilt:  $D = |t \times \vec{n} \odot \vec{d}|$  mit  $t = 1$ ,  $\vec{n} = \vec{n}_8$  und  $\vec{d} = (p_x, 0, 0)$  folgt:
 $D = |1 \times \vec{n}_8.x \times p_x|$ 

% Clipping Funktion für Diamond Constraint -----
diamondClipping( $\vec{dv}$ ) {
  % Bestimme die Facette in deren Richtung  $\vec{dv}$  zeigt
  IF  $\vec{dv}.x > 0$  THEN  $f \leftarrow 5$  ELSE  $f \leftarrow 1$ 
  IF  $\vec{dv}.y > 0$  THEN  $f \leftarrow f + 2$ 
  IF  $\vec{dv}.z > 0$  THEN  $f \leftarrow f + 1$ 
  % Berechne ob  $\vec{dv}$  zu lang ist und geclippt werden muss
  denom  $\leftarrow |\vec{n}_f \odot \vec{dv}|$  %  $\vec{n}_f$  ist die Normale der Facette f
  IF denom  $> D$  THEN % clipp D
     $t \leftarrow D / \text{denom}$  % Skalierungsfaktor für maximale Länge
     $\vec{dv} \leftarrow \vec{dv} \times t$ 
  ENDIF
  RETURN  $\vec{dv}$  % gibt  $\vec{dv}$  zurück
}

```

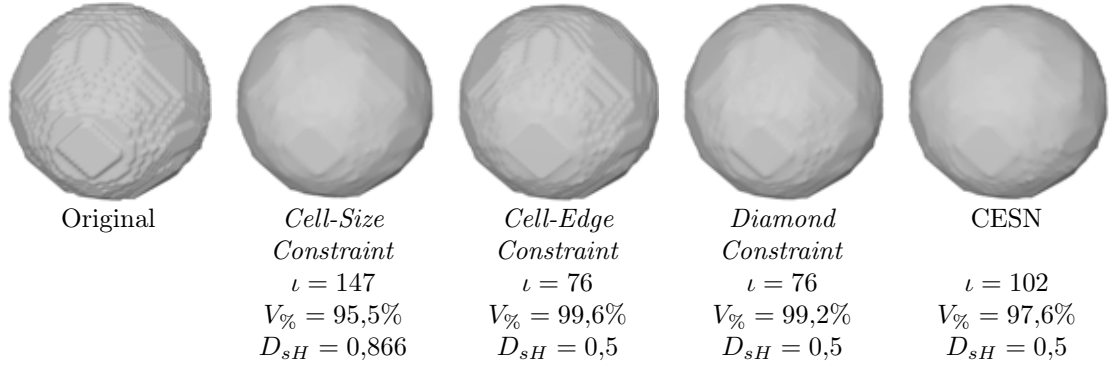
---

## 6.4 Ergebnisse

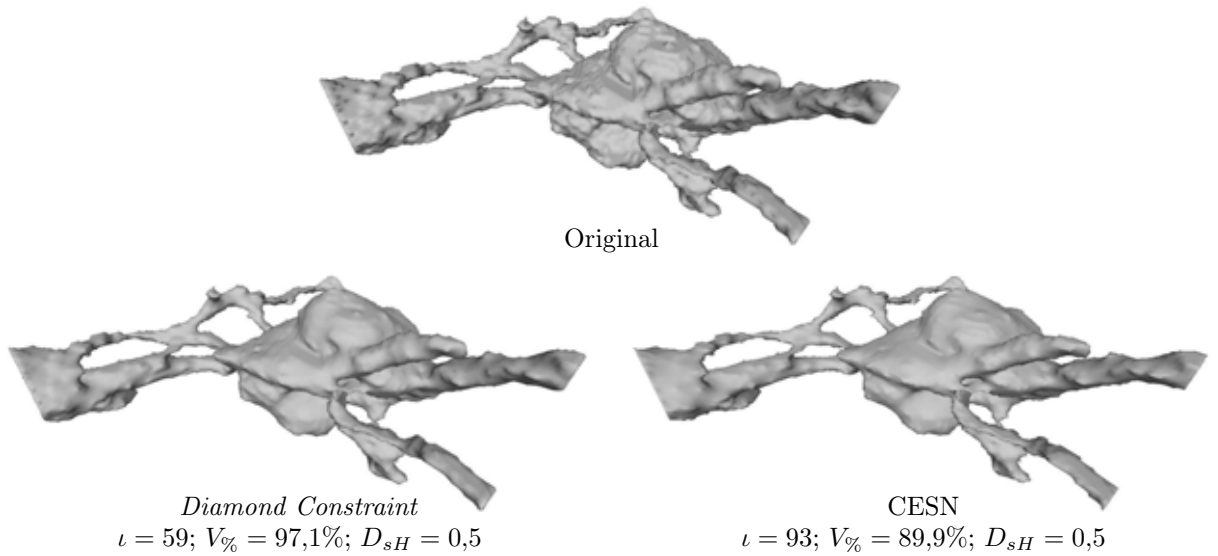
Durch die Verwendung des *Diamond Constraint* kann bei der Glättung von Oberflächenmodellen garantiert werden, dass diese nach der Glättung immer noch eine korrekte Innen-Außen Repräsentation der ursprünglichen Volumendaten darstellen. Abbildung 6.5 stellt für das Oberflächenmodell einer binär gesampelten Kugel die Glättungsergebnisse gegenüber. Hierzu wurde das mit MC extrahierte Oberflächenmodell der Kugel mittels Laplace-Glättung und jeweils einem der unterschiedlichen *Constraints* zusammen geglättet. Diesen Ergebnissen wird zusätzlich das mittels CESN aus den gleichen Volumendaten extrahierte und geglättete Oberflächenmodell gegenübergestellt (Abb. 6.5 rechts).

Aus der Gegenüberstellung ist ersichtlich, dass mit dem *Cell-Size Constraint* die beste Glättungswirkung erreicht wird. Jedoch schrumpft dieses Modell auch am stärksten und stellt keine korrekte Innen-Außen Repräsentation mehr dar (siehe Abschnitt 6.2). Dennoch kann mit dem *Cell-Size Constraint* die maximale Distanz ( $D_{sH}$ ) zum Ausgangsmodell auf maximal eine halbe Voxeldiagonale beschränkt werden. Dies macht das Verfahren im Vergleich zu den Glättungsverfahren ohne *Constraints* überlegen (vergleiche Kapitel 5).

Korrekte Innen-Außen Repräsentationen können hingegen bei der Verwendung des *Cell-Edge Constraint* und *Diamond Constraint* garantiert werden. Die erreichbare Glättungs-



**Abbildung 6.5:** Einfache Laplace-Glättung mit *Constraints* des Oberflächenmodells einer binär gesampelten Kugel mit einem Radius von 30 Voxeln. (Die Glättung wurde jeweils abgebrochen, sobald die maximale Änderung innerhalb einer Glättungsiteration unter  $\frac{1}{500}$  der Voxelgröße lag.)



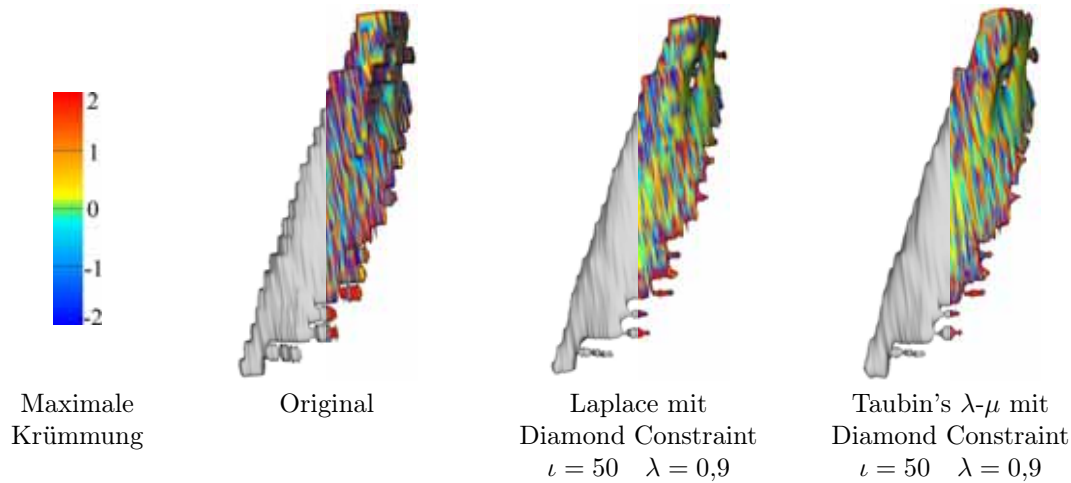
**Abbildung 6.6:** Glättungsergebnisse für ein aus Patientenaufnahmen extrahiertes Oberflächenmodell eines Aneurysmas. (Die Glättung wurde jeweils abgebrochen, sobald die maximale Änderung innerhalb einer Glättungsiteration unter  $\frac{1}{500}$  der Voxelgröße lag.)

wirkung ist jedoch beim *Diamond Constraint* am größten (siehe Abb. 6.5). Die maximale Distanz ( $D_{sH}$ ) zum Ausgangsmodell ist hierbei auf eine halbe Voxelgröße beschränkt. Ähnliche Ergebnisse können nur durch CESN (siehe Abb. 6.5 rechts) und DualMC erreicht werden (vergleiche Abschnitt 4.2.4). Dies betrifft sowohl die visuellen Ergebnisse als auch die quantitativen Fehlerwerte, wie in Abbildung 6.5 und 6.6 erkennbar.

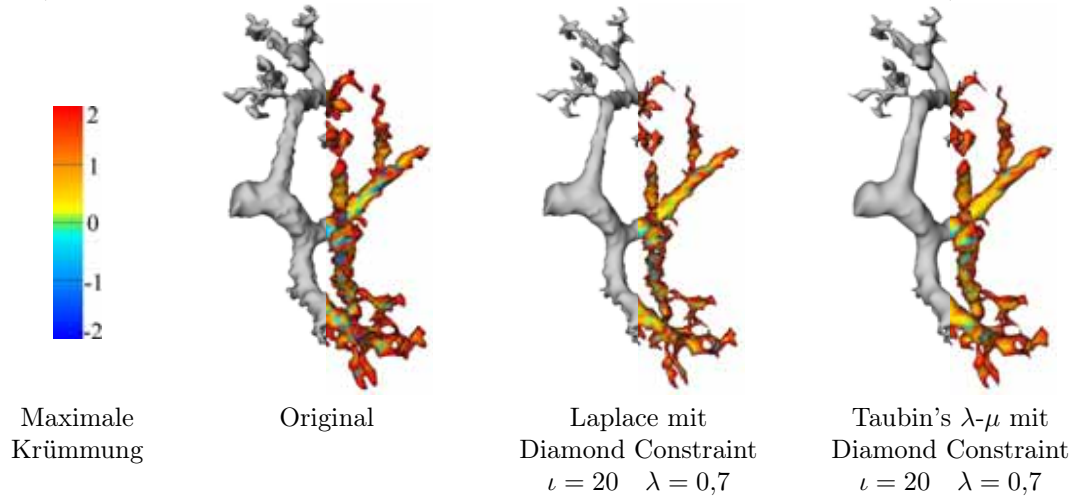
Jedoch verwenden Gibson [1998] in ihrem CESN-Ansatz und Nielson [2004] in ihrem DualMC-Ansatz nur einen Glättungsansatz, der der Laplace-Glättung entspricht. Folglich sind mit der in der vorliegenden Arbeit gewählten Implementation des *Diamond Constraint* durch die Kombination mit anderen Glättungsverfahren bessere Glättungsergebnisse möglich. Diese sind jedoch theoretisch auch mit dem CESN-Ansatz unter Einbindung der entsprechenden Glättungsverfahren möglich.

Das *Diamond Constraint* ermöglicht es nun auch, die Oberflächenmodelle des Halsmuskels und des Gefäßbaumes aus Kapitel 5 angemessen zu glätten. Dies war zuvor mit den Glättungsverfahren ohne *Constraints* nicht zufriedenstellend möglich. Abbildung 6.7 und 6.8 stellen exemplarisch die Glättungsergebnisse der Laplace und Taubin's  $\lambda$ - $\mu$  Glättung mit dem *Diamond Constraint* für den Halsmuskel und den Gefäßbaum dar. Die Treppenstu-





**Abbildung 6.7:** Ergebnisse des mit *Diamond Constraint* geglätteten Oberflächenmodells des Halsmuskels. (Ergebnismodelle stellen eine korrekte Innen-Außen Repräsentation dar.)



**Abbildung 6.8:** Ergebnisse des mit *Diamond Constraint* geglätteten Oberflächenmodells des Gefäßbaums. (Ergebnismodelle stellen eine korrekte Innen-Außen Repräsentation dar.)

fen im Halsmuskel sowie die überwiegende Anzahl der störenden Zacken in den Ästen des Gefäßbaums können dabei durch Taubin's  $\lambda$ - $\mu$  Glättung mit *Diamond Constraint* geglättet werden.

## 6.5 Zusammenfassung und Diskussion

Der Vergleich verschiedener Verfahren zur Glättung von Oberflächennetzen in Kapitel 5 zeigt, dass die Glättung und Artefaktreduktion bei den untersuchten Oberflächenmodellen nicht immer zufriedenstellend möglich war. Ein Grund hierfür ist, dass die Oberflächenmodelle unter der Glättung meist zu stark schrumpfen und sich dadurch zum Teil sehr große Abstände zum Ausgangsmodell ergeben. Infolgedessen stellen die meisten Ergebnisoberflächen keine korrekte Repräsentation der ursprünglichen Volumendaten mehr dar.

Um bei der Glättung von Oberflächenmodellen eine korrekte Innen-Außen Repräsentation zu erhalten, können Methoden Verwendung finden, welche die Bewegung der Oberflächenpunkte unter der Glättung auf einen zulässigen Toleranzbereich beschränken.



Dieser Strategie entsprechend wurde das *Diamond Constraint* bei der Glättung von Oberflächenmodellen eingeführt. Das *Diamond Constraint* ermöglicht dabei die Glättung von Oberflächenmodellen mit einem maximalem Fehler von einem halben Voxel. Gleichzeitig wird durch das *Diamond Constraint* sichergestellt, dass die Oberflächenmodelle nach der Glättung immer noch eine korrekte Innen-Außen Repräsentation der ursprünglichen Volumendaten darstellen.

Das *Diamond Constraint* Verfahren kann dabei als Übertragung des Ansatzes der *Constrained Elastic Surface Nets* CESN auf das Standardverfahren zur Oberflächenextraktion *Marching Cubes* MC (Abschnitt 4.2) verstanden werden. Die erreichten Ergebnisse gleichen hierbei in Bezug auf Genauigkeit und Glättungswirkung denen der CESN.

Alle nötigen Parameter für das *Diamond Constraint* Verfahren können aus den zugrunde liegenden Volumendaten oder auch, wie in [Bade et al. 2007a] beschrieben, aus dem Oberflächenmodell selbst ermittelt werden. Letzteres ermöglicht es auf der reinen Oberflächennetzebene auch ohne Kenntnis der Bilddaten eine korrekte Innen-Außen Repräsentation garantieren zu können.

Neben den genannten Vorteilen der voxelbegrenzten Oberflächennetzglättung mittels *Diamond Constraint* können mit dem Verfahren dennoch schmale, nur ein Voxel breite Strukturen zu einem Punkt bzw. einer Linie kollabieren. Allgemein kann der maximale Fehler von einem halben Voxel dünne Strukturen erheblich schrumpfen lassen. Folglich ist die Oberflächennetzglättung mit *Diamond Constraint* für die Reduktion von Artefakten in filigranen Objekten und Objektteilen (z.B. Blutgefäßen) nicht geeignet.



# Approximation von Oberflächenmodellen segmentierter Gefäßstrukturen

---

Zur Reduktion von Artefakten in Oberflächenmodellen von Gefäßbäumen, die aus binären Segmentierungsmasken extrahiert wurden, stehen wie die Kapitel 4 und 5 zeigen, keine effektiven Mittel zur Verfügung. Die Verwendung von *Constraints*, im Besonderen dem zuvor beschriebenen *Diamond Constraint* (Kapitel 6), verbessern die Möglichkeiten der Artefaktreduktion. Jedoch scheitern auch diese Verfahren an sehr kleinen, nur wenige Voxel starken Strukturen. Folglich müssen aus sehr kleinen Strukturen bestehende Objekte wie Blutgefäße separat Behandlung finden.

Für die Visualisierung von Gefäßstrukturen wird in diesem Kapitel, aus der Betrachtung existierender Ansätze zur Erzeugung von glatten qualitativ hochwertigen Oberflächenmodellen heraus (Abschnitt 7.2), die Approximation von Oberflächen mittels impliziter Oberflächen behandelt. Dabei werden Multi-level Partition of Unity Implicits (MPUIs) aufgrund ihrer Effizienz und Robustheit ausgewählt und für die Extraktion von Oberflächenmodellen segmentierter Gefäßstrukturen erweitert und angepasst (Abschnitt 7.3). Das spezialisierte MPUI-Verfahren ermöglicht dabei die effiziente Extraktion glatter Oberflächenmodelle von segmentierten Gefäßstrukturen. Die resultierenden Oberflächenmodelle (Ergebnisse siehe Abschnitt 7.5) weichen dabei im Mittel nur ca. um das 0,18-fache einer Voxeldiagonalen vom Standard MC Ergebnis ab. Dadurch steht mit den spezialisierten MPUIs ein effizientes Verfahren zur Extraktion von glatten (artefaktfreien) Oberflächenmodellen segmentierter Gefäßstrukturen zur Verfügung. Die Genauigkeit der extrahierten Oberflächenmodelle ist dabei den in dieser Arbeit untersuchten Glättungsverfahren und dem entwickelten *Diamond Constraint* Ansatz im Mittel überlegen. Mit den spezialisierten MPUIs erzeugte Oberflächenmodelle weisen jedoch lokal in Vertiefungen noch Abweichungen weit über einem Voxel auf, was Spielraum für Verbesserung lässt (Abschnitt 7.6).

## 7.1 Vorbetrachtungen

Die Visualisierung von Blutgefäßen ist wichtiger Bestandteil in der computerunterstützten Diagnose und Interventionsplanung. Bei der Interventionsplanung steht vor allem die Topologie und räumliche Lage von Gefäßbäumen und ihrer Verästelungen im Vordergrund. Hierbei muss zum Beispiel abgeschätzt werden, von welchen Blutgefäßen bestimmte Organabschnitte versorgt werden, oder ob ein Zugang an einer Stelle möglich ist, ohne dabei wichtige Gefäße zu verletzen.

Die Diagnostik erfordert Gefäßvisualisierungen, welche neben einer korrekten Abbildung des Verzweungsverhaltens auch die genaue Gefäßmorphologie wiedergeben. Dies ist erforderlich, um eventuell pathologische Veränderungen repräsentieren und diagnostizieren zu können.

Für beide Anwendungsgebiete ist eine möglichst glatte Repräsentation ohne Block- und Terrassenartefakte wünschenswert. Gleichzeitig ist vor allem für die Diagnostik eine mög-

lichst voxelgenaue Abbildung nötig. Eine besondere Herausforderung stellt die korrekte Rekonstruktion dünner, auch nur ein Voxel breiter Gefäßäste dar.

## 7.2 Verwandte Arbeiten

Alle existierenden Methoden zur Erzeugung qualitativ hochwertiger Oberflächenmodelle von segmentierten Gefäßstrukturen können grob in modellfreie und modellbasierte Verfahren unterteilt werden. Modellfreie Verfahren verwenden dabei die segmentierten Volumendaten als Input, während modellbasierte Verfahren Modellannahmen treffen und meist vereinfacht von einer Mittellinie und einem lokalen Gefäßdurchmesser zur Beschreibung der Gefäßstruktur ausgehen. In Abschnitt 4.2.3 wurde auf einige der Verfahren bereits kurz eingegangen, dies soll hier vertieft werden. Einen detaillierten Überblick geben Preim und Oeltze [2007].

### 7.2.1 Modellfreie Extraktion von Oberflächenmodellen

Zu den modellfreien Verfahren zur Erzeugung von Oberflächenmodellen zählen alle Methoden zur Oberflächenextraktion (Abschnitt 4.2). Diese kommen dabei meist in Verbindung mit Verfahren zur Glättung der extrahierten Oberflächenmodelle (Abschnitt 4.3) zum Einsatz.

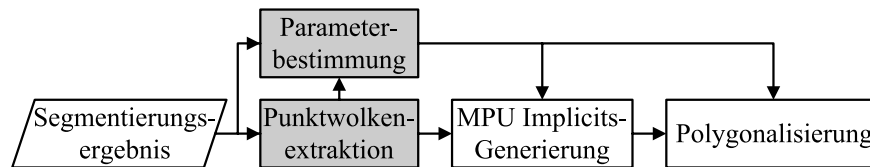
Die besten Ergebnisse können dabei mit den CESN bzw. mit dem *Diamond Constraint* Ansatz erreicht werden, da diese die Bewegung von Oberflächenpunkten während der Glättung auf einen Voxel begrenzen. Jedoch sind diese Verfahren nicht für kleine Blutgefäßstrukturen geeignet, da diese oft nur einen Voxel breit sind. Solche Strukturen degenerieren durch die Verfahren zu einer Linie (vergleiche Kapitel 6).

Eine andere Art modellfreier Oberflächenextraktionsverfahren approximieren Oberflächenmodelle aus einer Punktmengen- bzw. Punktwolkenrepräsentation eines Objektes, wie sie zum Beispiel von 3D-Laser-Range Scannern erzeugt werden. Zur Oberflächenapproximation wird dazu auf Basis der Punktwolke meist eine implizite Oberfläche erzeugt, welche über radiale Basisfunktionen (RBFs) [Turk und O'Brien 2002; Kojekine et al. 2003], *Level Sets* [Zhao und Osher 2002], *Multi-level Partition of Unity Implicits* (MPUIs) [Ohtake et al. 2003] oder *Regularized Membrane Potentials* [Jalba und Roerdink 2006] bestimmt wird. Der approximative Charakter der impliziten Oberflächen macht die erzeugten Oberflächen relativ robust gegenüber Rauschen und Ausreißern in der Punktwolke.

### 7.2.2 Modellbasierte Erzeugung von Oberflächenmodellen

Eine verbreitete Modellannahme ist, dass die darzustellenden Gefäßstrukturen einen kreisrunden Querschnitt besitzen. Dieser Annahme folgend entstanden vor allem Techniken, die Gefäßstrukturen über grafische Primitive mit kreisrundem Querschnitt nachbilden. Masutani et al. [1996] verwendeten Zylinder und Hahn et al. [2001] Kegelstümpfe. Es fanden auch *Simplex Meshes* [Bornik et al. 2005] oder zur Approximation sehr dünner Strukturen, die nicht komplett zusammenhängend segmentiert werden konnten, B-spline Oberflächen [Höhne et al. 2000] Anwendung. Weiterhin konnten *Convolution Surfaces* (CS) mit angepasstem Filterdesign und geeigneten Beschleunigungsstrategien erfolgreich zur Erzeugung von modellbasierten Oberflächenmodellen von Gefäßstrukturen angewendet werden [Oeltze und Preim 2004, 2005].

Modellbasierte Verfahren besitzen gegenüber modellfreien Verfahren zwei Nachteile: sie erfordern ein höheres Maß an Vorverarbeitung (für die Bestimmung der Mittellinien und



**Abbildung 7.1:** Approximation von Oberflächenmodellen segmentierter Gefäßstrukturen mit MPUIs. (modifiziert aus [Schumann 2006])

Gefäßradien) und sie bleiben aufgrund der Modellannahmen hinter der mit modellfreien Verfahren erreichbaren Genauigkeit zurück.

### 7.2.3 Fazit

Modellbasierte Verfahren sind für die genaue Repräsentation von segmentierten Strukturen ungeeignet. Für die Rekonstruktion von Oberflächenmodellen segmentierter Gefäßbäume wird deshalb die Oberflächenapproximation aus Punktwolken verwendet. Hierbei wird die Verwendung von MPUIs bevorzugt, da diese eins der schnellsten und robustesten Verfahren darstellen [Jalba und Roerdink 2006]. Erfolgreich konnten bereits Braude et al. [2007] medizinische Oberflächenmodelle über MPUIs rekonstruieren.

## 7.3 Approximation von Oberflächenmodellen segmentierter Gefäßstrukturen mit MPUIs

Ein Ansatz zur Approximation von Oberflächenmodellen aus segmentierten Volumendaten wurde von Braude et al. [2007] vorgestellt. Durch das Vorhandensein schmaler, meist nur ein Voxel breiter Gefäßstrukturen ist das Verfahren nicht für Gefäßstrukturen praktikabel. Hinzu kommt, dass im Verfahren von Braude et al. [2007] eine Reihe an Parametern frei eingestellt werden müssen, wobei nicht erklärt wird, wie gute Parameter zu finden sind. Für Gefäßstrukturen wird deshalb in [Schumann et al. 2007] eine Spezialisierung der von Braude et al. [2007] vorgestellten Vorgehensweise eingeführt, die im Folgenden vorgestellt wird.

Anhand einer gegebenen Punktwolke mit Normalen wird mittels MPUIs eine implizite Oberfläche approximiert (Abschnitt 7.3.1). Die erforderliche Punktwolkenrepräsentation mit zugehörigen Normalen wird dabei (angepasst an die Repräsentation von dünnen Strukturen) aus den segmentierten Volumendaten gewonnen (Abschnitt 7.3.2). Zur Steuerung von Genauigkeit und Glattheit resultierender Oberflächenmodelle werden nötige Parameter im Verfahren anhand der zu repräsentierenden Volumendaten ermittelt bzw., da wo es möglich ist, an die Approximation von Gefäßstrukturen angepasste voreingestellte Parameter verwendet (Abschnitt 7.3.3). Final wird die mittels MPUIs erhaltene implizite Oberfläche polygonalisiert. Abbildung 7.1 stellt den Gesamtprozess dar.

### 7.3.1 MPUIs

Die Rekonstruktion von Oberflächenmodellen mit *Multi-level Partition of Unity Implicits* (MPUIs) geht von einer Punktwolke aus, bei der für jeden Punkt eine Normale bekannt ist. Ohtake et al. [2003] führen für ihren MPUIs Ansatz (I) eine adaptive Unterteilung der Punktwolke durch. Dazu wird diese zunächst so skaliert, dass die Diagonale ihrer Bounding Box eine Längeneinheit groß ist. Mit einem Octree wird die skalierte Punktwolke dann unterteilt,

wobei jede Zelle des Octrees einen kugelförmigen Raum aufspannt. Der initiale Radius  $R$  dieses kugelförmigen Raumes ist dabei  $\alpha d$ , wobei  $d$  die Länge der Diagonale der Octreezelle ist. Sollte eine Zelle weniger als  $n_{min}$  Punkte enthalten, wird der Radius  $R$  iterativ um  $\lambda R$  erhöht, bis sich mindestens  $n_{min}$  Punkte im Einflussbereich der Zelle befinden.

Im zweiten Schritt (II) werden alle in einer Zelle enthaltenen Punkte über eine oder mehrere quadratische Funktionen approximiert. Dabei werden je nach Konstellation der Punkte und ihrer Normalen spezielle Strategien bereitgestellt [Ohtake et al. 2003]. Daraufhin wird der Abstand der lokalen Approximation zu den Punkten in der Octreezelle bestimmt. Überschreitet dieser eine gegebene Fehlertoleranz  $\epsilon_0$ , ist die gefundene Approximation noch nicht genau genug und die Octreezelle wird weiter unterteilt. Dies wird solange rekursiv durchgeführt, bis für alle Blätter des Octrees lokale Approximationen gefunden sind bzw. die maximale Rekursionstiefe  $level_{max}$  erreicht ist. Die vom Anwender zu spezifizierenden Parameter  $\alpha$ ,  $n_{min}$ ,  $\lambda$ ,  $\epsilon_0$  und  $level_{max}$  haben dabei Einfluss auf das Laufzeitverhalten sowie auf die erreichbare Glattheit und Genauigkeit.

Nachdem alle lokalen Approximationen bestimmt sind, werden diese (III) in eine globale Approximation überführt. Dazu wird eine Wichtungsfunktion über den Einflussbereich jeder Octreezelle gelegt, die mit steigendem Abstand zum Zentrum der Zelle den Einfluss abfallen lässt. Die globale Approximation bzw. implizite Oberfläche ergibt sich dabei über die Summe der gewichteten lokalen Approximationen.

Abschließend wird ein Oberflächenmodell über die (IV) Polygonalisierung der impliziten Oberfläche mit Bloomenthal's *Implicit Surface Polygonizer* (ISP) [Bloomenthal 1994] rekonstruiert. Dabei handelt es sich um einen beschleunigten MC Algorithmus für implizite Oberflächen, der auf den Kanten eines Würfelements Schnittpunkte mit der durch einen Iso-Wert definierten Oberfläche berechnet. Beschleunigt ist der ISP gegenüber dem MC, da er Würfelemente entlang der impliziten Oberfläche verfolgt (*surface tracking*) und damit die Betrachtung leerer Würfel bzw. Zellen stark reduziert wird. Die Größe des Würfelementes *grid\_size* und der zu verwendende Iso-Wert sind vom Anwender einzustellen.

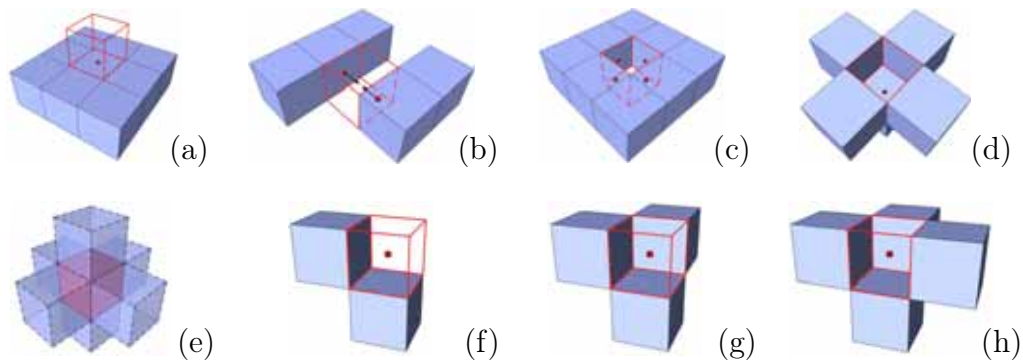
### 7.3.2 Punktwolkenextraktion

Bei der Punktwolkenextraktion erzeugen Braude et al. [2007] für jeden Randvoxel des segmentierten Objektes einen Punkt. Dünne, nur ein Voxel breite Strukturen werden dabei nur durch eine Linie von Punkten repräsentiert, die kein Volumen aufspannen. Eine Rekonstruktion dieser Strukturen mit MPUIs wäre nicht möglich.

Als Spezialisierung für die Punktwolkenextraktion bei Gefäßstrukturen wird deshalb die Verwendung aller Hintergrundvoxel in direkter Nachbarschaft zu einem Objektvoxel vorgeschlagen. Im Volumen dieser im Folgenden *äußere Randvoxel* genannten Hintergrundvoxel werden dann Punkte erzeugt. Unter Berücksichtigung der 3D-6er Nachbarschaft eines äußeren Randvoxel  $v$  ergeben sich sechs Fälle bei denen wie folgt Punkte erzeugt werden [Schumann et al. 2007]:

- Fall 1 Nur ein Objektvoxel ist zu  $v$  benachbart (Abb. 7.2 (a)). In diesem Fall wird ein Punkt im Zentrum der Grenzfläche zwischen  $v$  und dem benachbarten Objektvoxel erzeugt.
- Fall 2  $v$  hat genau zwei benachbarte Objektvoxel, die sich gegenüberliegen (Abb. 7.2 (b)). Es werden zwei Punkte in den Zentren der Grenzflächen erzeugt.

- Fall 3  $v$  hat vier benachbarte Objektvoxel, die alle in einer Ebene liegen (Abb. 7.2 (c)). Hier werden vier Punkte in den Zentren der Grenzflächen erzeugt.
- Fall 4  $v$  hat fünf benachbarte Objektvoxel, so dass  $v$  in einer Einbuchtung liegt (Abb. 7.2 (d)). Es wird ein Punkt im Zentrum der Grenzfläche des Objektvoxels erzeugt, dem kein Objektvoxel gegenüberliegt (Boden der Einbuchtung).
- Fall 5  $v$  ist komplett von sechs Objektvoxeln umgeben, so dass  $v$  einen Hohlraum im Inneren darstellt (Abb. 7.2 (e)). In diesem Fall wird kein Punkt erzeugt und der ein Voxel große Hohlraum ignoriert.
- Fall 6 In allen anderen Fällen bilden die benachbarten Objektvoxel eine Treppstufe und es wird nur ein Punkt im Zentrum von  $v$  erzeugt (Abb. 7.2 (f), (g) und (h)).



**Abbildung 7.2:** Fallbasierte Punktwolkenextraktion in äußerem Randvoxel je nach Konstellation benachbarter Objektvoxel. (modifiziert aus [Schumann 2006])

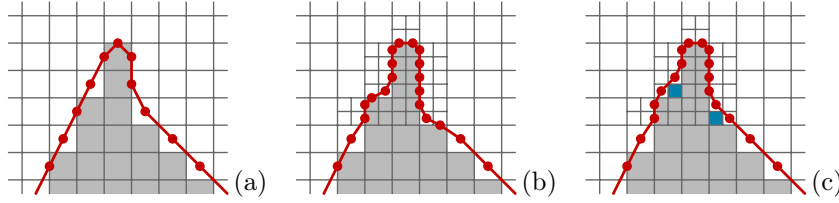
### Bestimmung von Normalen

Für die Oberflächenapproximation mit MPUIs ist pro Punkt jeweils auch eine Normale erforderlich. In den Fällen, in denen nur ein Punkt in einem äußeren Randvoxel  $v$  erzeugt wurde (Fall 1, 4 und 6), kann deren Normale über den Gradienten von  $v$  in den segmentierten Volumen berechnet werden. Hierzu wird der Gradient über die partiellen Ableitungen, wie in [Tiede 1999] beschrieben, ermittelt. Andere Verfahren dazu, wie in [Zucker und Hummel 1981] beschrieben oder in [Braude 2005] verwendet sind auch möglich, zumal sich die resultierenden Normalen nur wenig unterscheiden [Pommert 2004].

In den Fällen, in denen zwei oder mehr Punkte erzeugt werden (Fall 2 und 3), kann die Gradientenberechnung nicht verwendet werden, da hier für jeden Punkt die gleiche Normale berechnet werden würde. Weiterhin kann es zu Null-Vektoren kommen, wenn wie in Fall 3 (Abb. 7.2 (c)) eine symmetrische Nachbarschaft um  $v$  existiert. In diesen Fällen wird der Normalenvektor der Grenzflächen, auf denen die Punkte liegen, genommen (siehe Abb. 7.2 (b)).

### Überabtastung dünner Strukturen

Zur adäquaten Repräsentation von dünnen Strukturen müssen diese durch mehr Punkte repräsentiert werden, da sonst kugelförmige Artefakte durch die Approximation mit MPUIs auftreten können [Schumann 2006]. Hierzu werden die Voxel des Volumendatensatzes in acht *Sub-Voxel* unterteilt. Eine entsprechende Überabtastung des gesamten Volumens würde die



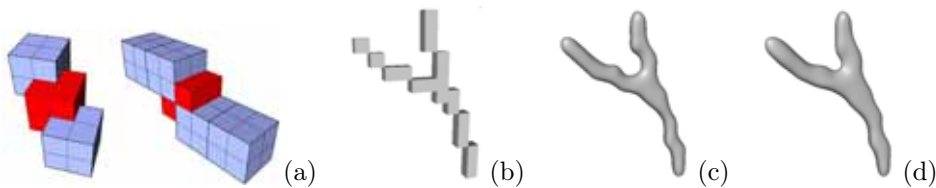
**Abbildung 7.3:** Punktwolkenextraktion: (a) ohne Überabtastung, (b) lokale Überabtastung, wobei die Punkte in den *Sub-Voxel* stark dem originalen Voxelgitter folgen und (c) zusätzlich als Objektvoxel markierte *Sub-Voxel* mildern die Stufenartefakte. (modifiziert aus [Schumann et al. 2007])

Anzahl an Punkten dramatisch erhöhen. Da eine erhöhte Anzahl an Punkten nur bei dünnen Strukturen benötigt wird, werden diese mit einer Top-Hat-Transformation mit einem  $3 \times 3 \times 3$  Strukturelement detektiert. Anschließend werden nur die äußeren Randvoxel der detektierten dünnen Strukturen in *Sub-Voxel* aufgeteilt. Für jedes dieser *Sub-Voxel* werden analog der vorgestellten Fallunterscheidung (S. 98) Punkte und Normalen für die Punktwolkenrepräsentation erzeugt.

Durch diese Überabtastung folgen die extrahierten Punkte jedoch sehr stark dem Voxelgitter, wodurch Stufenartefakte begünstigt werden (Abb. 7.3 (b)). Um dies auszugleichen, werden wie in Abbildung 7.3 (c) dargestellt, einige der entstandenen *Sub-Voxel* als Objektvoxel markiert. Hierzu wird die Konstellation von Objektvoxeln in ihrer 3D-18er Nachbarschaft betrachtet. Ein *Sub-Voxel* wird dabei genau dann als zusätzliches Objektvoxel markiert, wenn *alle* der drei folgenden Kriterien erfüllt sind:

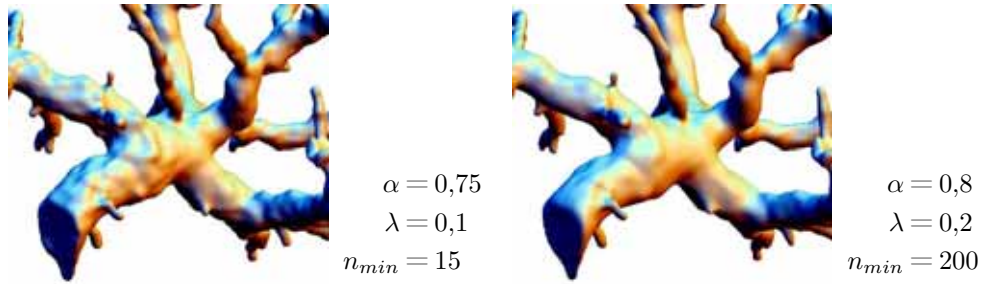
- Das *Sub-Voxel* befindet sich in keiner Mulde oder Röhre (Fall 3 und 4 siehe Abb. 7.2)
- Das *Sub-Voxel* grenzt mindestens an zwei Objektvoxel an, die sich keine gemeinsame Fläche teilen und
- mindestens eins dieser Objektvoxel teilt sich eine gemeinsame Fläche mit dem *Sub-Voxel*.

*Sub-Voxel* in Mulden oder Röhren dürfen nicht als Objektvoxel markiert werden, da diese dadurch sonst gefüllt würden. In Abbildung 7.4 (a) sind Beispiele für zusätzlich als Objektvoxel markierte *Sub-Voxel* dargestellt. Resultierende Punktwolken erlauben dadurch eine glattere Approximation (Abb. 7.4 (c) und (d)). Die extrahierte Punktwolke wird dann in Weltkoordinaten transformiert, nachdem zuvor noch automatisch Parameter für die Oberflächenapproximation abgeleitet wurden.



**Abbildung 7.4:** Überabtastung: (a) Beispiele für zusätzlich als Objektvoxel markierte *Sub-Voxel* (rot). Approximierte Oberflächen: (b) initiale Objektvoxel, (c) Stufenartefakte durch Überabtastung und (d) glattere Oberfläche durch zusätzlich als Objektvoxel markierte *Sub-Voxel*. (modifiziert aus [Schumann et al. 2007])





**Abbildung 7.5:** Glattere Oberfläche durch höhere Parameterwerte für  $\alpha$ ,  $\lambda$  und  $n_{min}$ : (a) Werte nach Ohtake et al. [2003] und (b) mehr Punkte für lokale Approximation mit höheren Parameterwerten. (modifiziert aus [Schumann et al. 2007])

### 7.3.3 Automatische Parameterwahl

Die Erzeugung der impliziten Oberfläche mit MPUIs sowie deren Polygonalisierung sind über viele Parameter zu steuern (siehe Abschnitt 7.3.1). Schumann [2006] untersucht dazu die Auswirkung der verschiedenen Parameter und schlägt angepasst für die Approximation von Gefäßstrukturen geeignete Default-Werte sowie automatische, auf die aktuellen Daten angepasste Mechanismen zur Parameterbestimmung vor. Diese werden im Folgenden vorgestellt. Für eine ausführliche Besprechung sei auf [Schumann 2006] verwiesen.

Für  $n_{min}$ ,  $\alpha$  und  $\lambda$  können feste Werte für alle Datensätze verwendet werden. Dabei werden für die Approximation von Gefäßstrukturen die Werte im Gegensatz zu den von Ohtake et al. [2003] verwendeten erhöht. Dadurch wird erzwungen, dass mehr Punkte pro Octreezelle für eine lokale Approximation verwendet werden. Dadurch kann eine glattere Oberfläche rekonstruiert werden (Abb. 7.5).

Die Parameter  $\epsilon_0$ ,  $level_{max}$  und  $grid\_size$  werden in Abhängigkeit von der Bounding Box des Volumendatensatzes in Voxelkoordinaten berechnet. Dadurch kann die Genauigkeit und Auflösung der Approximation und Polygonalisierung automatisch an die des Volumendatensatzes angepasst werden.

Als maximale Abweichung lokaler Approximationen zur Punktwolke wird die halbe Voxelgröße verwendet.  $\epsilon_0$  im skalierten Raum ergibt sich dabei aus der Länge der Diagonale des unskalierten Volumens in Voxelkoordinaten  $l_{diag}$  über:

$$\epsilon_0 = \frac{1}{2} \cdot \frac{\sqrt{3}}{l_{diag}} \quad (7.1)$$

Für die Approximation mittels MPUIs muss zusätzlich die maximale Tiefe  $level_{max}$  des Octrees bestimmt werden, um Artefakte durch zu wenig Punkte in einer Octreezelle zu vermeiden. Dazu wird das maximale Unterteilungslevel so gewählt, dass die kleinste Octreezelle die Größe eines Voxels hat. Hierzu wird die geringste Ausdehnung der Bounding Box der Punktwolke in Voxelkoordinaten  $(ext_x, ext_y, ext_z)$  bestimmt und daraus  $level_{max}$  wie folgt abgeleitet:

$$level_{max} = \lceil \log_2(\min(ext_x, ext_y, ext_z)) \rceil \quad (7.2)$$

Für die Polygonalisierung wird die Größe des Würfелеlementes bzw. des Gitters  $grid\_size$  etwas kleiner als die originale Voxelgröße gewählt:

$$grid\_size = \frac{0,7}{\max(ext_x, ext_y, ext_z)} \quad (7.3)$$

Für die Polygonalisierung der impliziten Oberfläche wird ein Iso-Wert etwas über 0 gewählt, da andernfalls schmale Strukturen bei der Polygonalisierung verfehlt werden können.  $\frac{1}{10}$  der erlaubten Fehlertoleranz  $\epsilon_0$  scheint dafür geeignet. Dies entspricht  $\frac{1}{20}$  der Voxeldiagonale.

## 7.4 Umsetzung

Das beschriebene Verfahren zur Approximation von Oberflächen segmentierter Gefäßstrukturen wurde im Rahmen einer Diplomarbeit in MEVISLAB [MeVis Research 2007] implementiert [Schumann 2006]. Für Ergebnisse durch das MPUI-Verfahren von Ohtake et al. [2003] wurde die von Ohtake et al. bereitgestellte MPUI Software<sup>1</sup> genutzt.

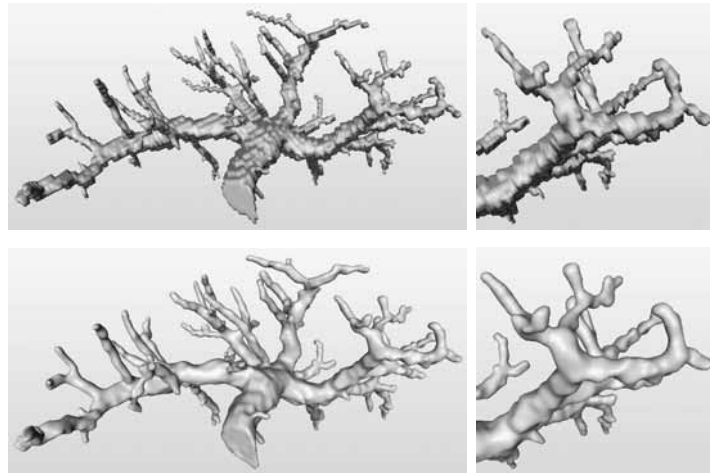
## 7.5 Ergebnisse

Wie Abbildung 7.6 und 7.7 illustrieren, können mit Hilfe des vorgestellten MPUIs Ansatzes effizient Block- und Stufenartefakte segmentierter Gefäßstrukturen vermieden werden. Abbildung 7.8 stellt die erreichbaren Ergebnisse im Vergleich zu *Convolution Surfaces* (CS) [Oeltze und Preim 2005] und *Constrained Elastic Surface Nets* (CESN) [Gibson 1998] dar. Dabei ist zu beobachten, dass Morphologie und Topologie der Gefäßstrukturen am besten mit den MPUIs rekonstruiert werden können. In Kontrast zu den Ergebnissen mit CS können kleine Oberflächendetails korrekt repräsentiert werden. Darüber hinaus können CS die Morphologie der Gefäße durch die Modellannahme eines kreisrunden Querschnitts nur annähernd darstellen. Im Vergleich zu den CESN können schmale Gefäße besser rekonstruiert werden.

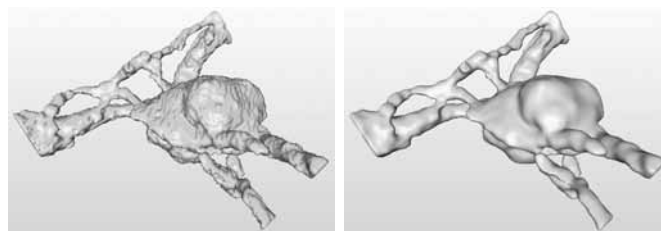
In Tabelle 7.1 werden die Verfahren bezüglich ihrer Genauigkeit gegenübergestellt. Hierzu wurde mit AMIRA [Mercury Computer Systems 2007] jeweils der Abstand der resultierenden Oberflächen zur mittels *Marching Cubes* (MC) extrahierten Oberfläche berechnet. Der Median-Abstand mit MPUIs ist sehr ähnlich zu dem der CESN. Jedoch fällt der maximale Abstand mit den MPUIs größer als mit den CESN aus. Während der maximale Abstand bei den CESN im Maximum auf  $\frac{1}{2}$  der Voxeldiagonale beschränkt ist (siehe Abschnitt 6.2), wird diese Grenze beim vorgestellten MPUIs Ansatz überschritten. Diese Maximalwerte treten an kleinen Einbuchtungen auf, welche nicht durch genügend Punkte in der Punktwolke repräsentiert werden. Die verwendete Strategie zur Überabtastung berücksichtigt diese Einbuchtungen nicht, da diese bei der Repräsentation von Gefäßstrukturen wenig relevant sind. Weitere Ursachen für diese Abweichungen sind die begrenzte Octreetiefe sowie der verwendete Iso-Wert über 0. Abgesehen von diesen Ausreißern des maximalen Abstandes ähneln sich das vorgestellte MPUI-Verfahren und CESN in Bezug auf die erreichbare Genauigkeit, wobei beide Verfahren die Segmentierungsergebnisse genauer abbilden als dies mit CS möglich ist.

In Bezug auf die Effizienz unterscheiden sich CS, MPUIs und CESN vom MC durch die Anzahl der Dreiecke  $\|\mathcal{F}\|$ , aus denen die erzeugten Oberflächen zusammengesetzt sind sowie in der benötigten Rechenzeit  $t$  (siehe Tabelle 7.2). Die Anzahl der mit dem vorgestellten MPUIs Ansatz erzeugten Dreiecke ist ca. um das 1,2 fache größer als mit MC und CESN. Dies liegt daran, dass eine etwas kleinere Gittergröße für die Polygonalisierung gewählt wird (siehe Gleichung 7.3 Abschnitt 7.3.3). Die größere Abweichung der Dreiecksanzahl beim Lebergefäßbaum (Tab. 7.2, Zeile zwei) entsteht durch die anisotrope Voxelauflösung in den

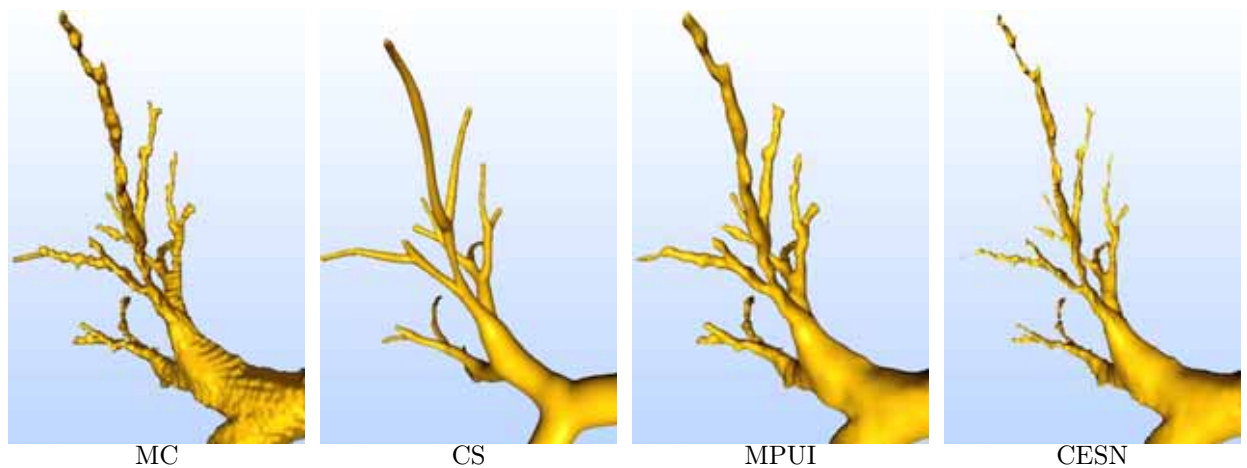
<sup>1</sup>[www.mpi-sb.mpg.de/~ohtake/mpu\\_implicit](http://www.mpi-sb.mpg.de/~ohtake/mpu_implicit)s



**Abbildung 7.6:** Oberflächenrekonstruktion eines Lebergefäßbaumes mit *Marching Cubes* (MC) (oben) und dem vorgestellten MPUI-Verfahren (unten). (modifiziert aus [Schumann 2006])



**Abbildung 7.7:** Oberflächenrekonstruktion eines Aneurysmas mit MC (links) und dem vorgestellten MPUI-Verfahren (rechts). (modifiziert aus [Schumann 2006])



**Abbildung 7.8:** Oberflächenrekonstruktion eines Bronchialbaums mit *Marching Cubes* (MC), *Convolution Surfaces* (CS) [Oeltze und Preim 2005], dem vorgestellten MPUI-Verfahren und mittels CESN [Gibson 1998]. [Schumann et al. 2007]

Datensatz	CS		MPUIs		CESN	
	med	max	med	max	med	max
Bronchialbaum	0,31	5,53	0,16	1,40	0,17	0,44
Lebergefäßbaum	0,30	6,87	0,15	0,84	0,14	0,47
Cerebralbaum	0,32	2,70	0,20	1,68	0,21	0,39
Aneurysma	0,72	6,25	0,19	1,91	0,14	0,40

**Tabelle 7.1:** Genauigkeit der Oberflächenrekonstruktion mit CS, dem vorgestellten MPUIs Ansatz und CESN. Beispielhaft wird für vier Datensätze jeweils der Median (med) und das Maximum des Abstandes zur MC Oberfläche als Vielfaches der Voxeldiagonale angegeben. (modifiziert aus [Schumann et al. 2007])

Datensatz	MC		CS		MPUIs		CESN	
	$\ \mathcal{F}\ $	$t$	$\ \mathcal{F}\ $	$t$	$\ \mathcal{F}\ $	$t$	$\ \mathcal{F}\ $	$t$
Bronchialbaum	166k	3	201k	36	180k	38	166k	13
Lebergefäßbaum	80k	2	125k	15	167k	14	81k	5
Cerebralbaum	115k	3	93k	9	142k	28	115k	8
Aneurysma	54k	1	39k	11	61k	5	50k	2

**Tabelle 7.2:** Effizienz der Oberflächenrekonstruktion mit CS, MPUIs und CESN. Aufgelistet ist die Dreiecksanzahl  $\|\mathcal{F}\|$  sowie die benötigte Rechenzeit  $t$  in Sekunden. (modifiziert aus [Schumann et al. 2007])

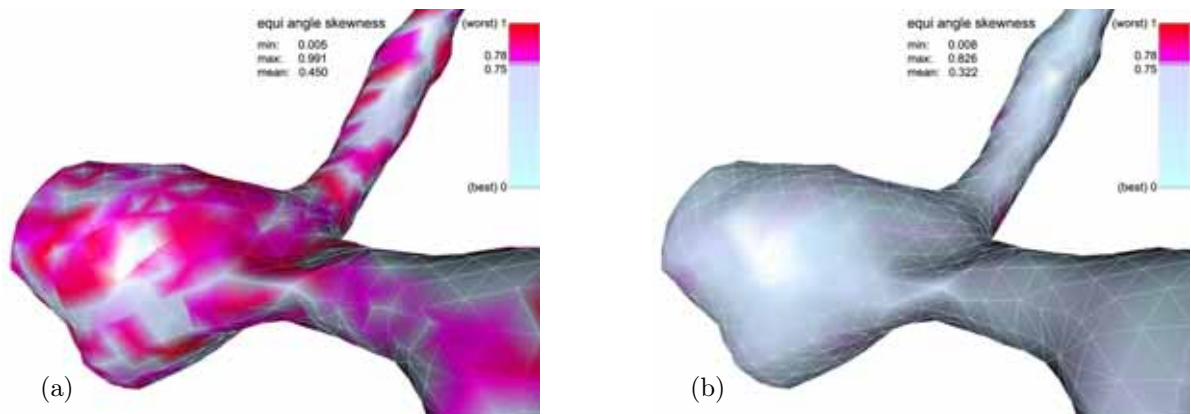
Originaldaten. Dadurch passen bei der Polygonalisierung der MPUIs mehr als zwei Gitterelemente in einen Voxel. Die durch die MPUIs erzeugte Dreiecksanzahl ist jedoch in einem akzeptablen Bereich, der von moderner Grafikhardware in interaktiven Bildraten verarbeitet werden kann. Die Berechnungszeit ist generell mit MC und CESN am kürzesten. Mit MPUIs kann die Geschwindigkeit bei Bedarf auf Kosten von Glattheit und Genauigkeit mit geeigneter Parameterwahl gesenkt werden (z.B.: erhöhte Gittergröße bei der Polygonalisierung).

## 7.6 Zusammenfassung und Diskussion

Sowohl „normale“ Glättungsverfahren (Kap. 5) als auch die Glättung mit *Constraints* (Kap. 6) sind für Gefäßstrukturen mit schmalen bis zu einem Voxel breiten Elementen nicht geeignet. Unter Berücksichtigung derart schmaler Strukturen kann das vorgestellte auf MPUIs basierende Verfahren die qualitativ hochwertigsten Ergebnisse erzielen, während bestmöglich Morphologie und Topologie abgebildet werden. Mit einer durchschnittlichen Median-Abweichung von 0,18 Voxeldiagonalen ist das Verfahren fast so genau wie die CESN (0,17). Die automatische Bestimmung der Parameterwerte für das Verfahren erleichtert zusätzlich den praktischen Einsatz. Die erreichte Qualität sowie die Genauigkeit des Verfahrens machen einen Einsatz in der Gefäßdiagnostik möglich. Letztere kann jedoch noch weiter verbessert werden, indem z.B. die Überabtastung der Voxel auf kleine Einbuchtungen erweitert wird. Darüber hinaus kann die Genauigkeit und Effizienz des Verfahrens mit einem adaptiven Algorithmus zur Polygonalisierung gesteigert werden. So kann eine lokale, krümmungsbasierte Anpassung der Gittergröße die Anzahl erzeugter Dreiecke und damit auch die Berechnungszeit senken sowie die genauere Repräsentation von kleinen Einbuchtungen erhöhen.

Ein Anwendungsgebiet, in dem das Verfahren bereits erfolgreich erprobt wurde, ist die Simulation des Blutflusses in Gefäßen [Bade et al. 2007b]. Hierbei werden artefaktfreie, mög-

lichst genaue Oberflächenmodelle als Input zur Erzeugung von Volumengittern für die numerische Flusssimulation (CFD) benötigt. Block- und Terrassenartefakte sowie eine nur schlechte Repräsentation der genauen Morphologie (wie bei den CS) würden dabei zu falschen Simulationsergebnissen führen. Für die Erzeugung von Volumengittern für die CFD muss jedoch das Oberflächenmodell zusätzlich auch eine gute Dreiecksqualität aufweisen. Diese wird, bedingt durch den verwendeten Polygonalisierer, derzeit nicht erreicht (Abb. 7.9 (a)). Über das Kollabieren und Flippen von Kanten im Oberflächenmodell kann jedoch eine ausreichende Dreiecksqualität erreicht werden (Abb. 7.9 (b)) [Bade et al. 2007b]. Dies ist jedoch ein weiterer Grund, sich zukünftig mit einer verbesserten Polygonalisierung der MPUIs zu befassen.



**Abbildung 7.9:** Verbesserung der Dreiecksqualität durch Kollabieren kleiner Kanten und Dreiecke: (a) Ausschnitt aus einem mittels Bloomenthal's *Implicit Surface Polygonizer* (ISP) [Bloomenthal 1994] polygonalisierten MPUIs Ergebnis, (b) gleicher Ausschnitt wie in (a) nach dem Kollabieren und Flippen von Kanten nach [Bade et al. 2007b]



---

# Effiziente 3D-Navigation

---

Die Attraktivität und das Potential der Computerunterstützung in der Ausbildung und Interventionsplanung ist unter anderem darin begründet, dass durch die Computerunterstützung mit den dargestellten Inhalten interagiert werden kann. Dies macht das gefahrlose Erproben von Therapieentscheidungen sowie eine intensivere Auseinandersetzung mit dargestellten Informationsräumen möglich [Ritter 2005].

In diesem Kapitel werden Techniken zur interaktiven Exploration von medizinischen Oberflächenmodellen und 3D-Szenen betrachtet. Dabei wird das Rotieren der Szene über direkte Manipulation mit einer Standard-PC-Maus als wichtigste Interaktionsmethode genauer untersucht. Hierzu werden zuerst Kriterien für zweckmäßige und steuerbare Rotationstechniken zusammengetragen, die dann zur Analyse verschiedener existierender Ansätze genutzt werden. Des Weiteren werden die existierenden Rotationstechniken bezüglich ihrer Angemessenheit für die Exploration von medizinischen Szenen evaluiert. Die Ergebnisse der Evaluation führen zur Identifikation der effizientesten Rotationstechnik für die Exploration von 3D-Szenen für die chirurgische Ausbildung und Interventionsplanung.

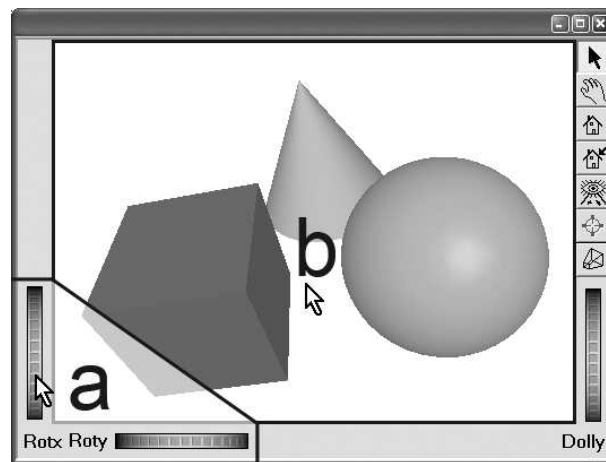
## 8.1 Problem- und Anforderungsanalyse

Dreidimensionale Visualisierungen bergen ein großes Potential für die chirurgische Ausbildung und Interventionsplanung. Detaillierte und hoch aufgelöste Oberflächenmodelle aus medizinischen Daten können dank moderner Grafikhardware mit interaktiven Bildwiederholraten selbst auf handelsüblichen Computern selbstständig exploriert werden. Trotz der Vorteile von 3D-Visualisierungen werden diese nur von einer Minderheit von Anwendern genutzt, welche 3D-Ansichten wiederum nur für einige wenige ausgewählte Aufgaben benutzen. Stattdessen analysieren Ärzte CT- und MRT-Daten vornehmlich Schicht für Schicht und planen so ihre Interventionen. Ein Grund, dass weitgehend auf dreidimensionale Visualisierungen verzichtet wird bzw. diese nur eingeschränkt genutzt werden, kann in der sehr komplexen Mensch-Computer-Interaktion zur 3D-Navigation und -Exploration sowie im Fehlen von angemessenen intuitiven Interaktionstechniken gesehen werden.

### 8.1.1 Problemanalyse

Im dreidimensionalen Raum sind zur exakten Navigation sechs Freiheitsgrade zu kontrollieren (1-3: Bewegung in alle drei Raumrichtungen; 4-6: Orientierung bzw. Rotation jeweils um eine der drei Raumrichtungsachsen). Im Fall der dynamischen Interaktion kommt die Bestimmung der Bewegungsgeschwindigkeit als zusätzlicher Parameter hinzu.

Die technisch einfachste Variante eine entsprechende Vielzahl an Parametern kontrollieren zu können, ist die indirekte Manipulation über separate Widgets, wie zum Beispiel „*scroll wheels*“ für die Rotation (siehe Abb. 8.1 (a)). Jedoch erhöht die indirekte Kopplung der Widgets und ihrer Manipulation zur Steuerung der eigentlichen Szene die kognitive Belastung. Überlegen sind deshalb Methoden zur direkten Manipulation, die eine direkte Interaktion mit der Szene ermöglichen (siehe Abb. 8.1 (b)). Geräte, die eine direkte Manipulation mit allen Freiheitsgraden im dreidimensionalen Raum ermöglichen, sind jedoch meist für Standard-2D-Interaktionen (wie z.B. in normalen Windowsanwendungen) ungeeignet und



**Abbildung 8.1:** Standardkomponenten zur Exploration von 3D-Szenen am Beispiel des Open Inventor Toolkit [SGI 2007] mit: (a) *scroll wheels* zur indirekten und (b) einem maussensitiven Anzeigebereich zur direkten Manipulation der 3D-Ansicht.

werden unter anderem deshalb immer noch wenig verwendet. Entsprechend spezialisierte Eingabegeräte bieten sich für Spezialarbeitsplätze (z.B. für Medizinische Workstations zur Interventionsplanung) an. Die Abhängigkeit von dem Vorhandensein und der Verfügbarkeit solcher Geräte ist jedoch für die computerunterstützte chirurgische Ausbildung, die unter anderem ein ort- und zeitunabhängiges Lernen ermöglichen soll, nicht praktikabel.

Die Steuerung bzw. Manipulation aller Freiheitsgrade im dreidimensionalen Raum ist jedoch nicht in allen Anwendungen nötig. Bei der Exploration von medizinischen Szenen kann hierzu zwischen zwei grundlegenden Anwendungsszenarien unterschieden werden: (1) die Exploration eines Organs von innen und damit seiner Innenseite (z.B. bei der virtuellen Endoskopie) und (2) die Exploration der Außenseite eines Organs bzw. allgemein mit Blick von außen auf das zu betrachtende Objekt (z.B. bei der Osteotomie- oder Leberoperationsplanung). Beim ersten Anwendungsszenario kann bildlich von einem fliegenden Vehikel („*flying-vehicle*“ und „*camera-in-hand*“ Metapher) gesprochen werden, dessen Steuerung von Bewegungsrichtung, Orientierung und Geschwindigkeit zur Exploration der Szene alle angesprochenen Freiheitsgrade benötigt. Bei der Betrachtung der Szenenobjekte von außen (zweites Anwendungsszenario) genügt zur Exploration bei entsprechend kompakten Szenen die Rotation der zu betrachtenden Szene um alle drei Raumachsen („*world-in-hand*“ Metapher). Mit dieser Art der Interaktion fühlt sich die Steuerung der Rotation so an, als ob der Nutzer das Objekt an sich dreht und nicht die Kamera. Dies hat sich als am besten geeignet für das Rotieren entsprechender Szenen erwiesen [Partala 1999]. Wie die in dieser Arbeit aufgeführten Anwendungsbeispiele zeigen, kann grundsätzlich mit Abgrenzung von Anwendungen der virtuellen Endoskopie von entsprechend kompakten und von außen zu betrachtenden Szenen für die chirurgische Ausbildung und Interventionsplanung ausgegangen werden. Entsprechend bieten sich so genannte Rotationstechniken zur Rotation der Szene bzw. zum Umfliegen der Szene mit der virtuellen Kamera (auch „*orbiting*“ genannt) unter Nutzung der „*world-in-hand*“ Metapher an.

In Softwaresystemen, zum Beispiel für geometrisches Modellieren und Computer-Aided Design (CAD), wird eine Vielzahl an Rotationstechniken mit unterschiedlichen Verhaltensweisen eingesetzt. Verwendete Rotationstechniken in existierenden Systemen für die chirurgische Ausbildung und Interventionsplanung sind meist durch das zugrunde liegende Visualisierungssystem und die von diesem bereitgestellten Interaktionstechniken bestimmt. Dabei



ist äquivalent zum Einsatz von problemangepassten und wahrnehmungsbasiert motivierten Visualisierungen, die Bereitstellung von aufgabenangepassten Interaktionstechniken zum Explorieren und Navigieren der dreidimensionalen Visualisierung von Bedeutung.

Aufgaben, die sich Anwendern beim Explorieren von und Navigieren in 3D Visualisierungen in der computerunterstützten medizinischen Ausbildung und Interventionsplanung stellen, betreffen dabei die Exploration der gesamten Szene (z.B. Beurteilung der allgemeinen Lagebeziehung und Form der momentan betrachteten Organe) oder die genauere Inspektion einzelner Organe und Organteile (z.B. zur Identifikation von krankhaften Veränderungen oder zur Bestimmung von Resektionslinien oder Frakturierungskanten). Hierbei können sich sowohl freie Interaktionen (z.B. das Absuchen einer Szene nach relevanten Strukturen) als auch wohldefinierte Aufgaben (z.B. das Heranfahren (Zoomen) und Inspizieren einer relevanten Struktur nach der anderen, um z.B. deren Form und relative Lage zu beurteilen) ergänzen. Die Abarbeitung einer definierten Aufgabe kann dabei meist mehrere Interaktionsschritte beinhalten, wie zum Beispiel das Aufsuchen einer geeigneten Blickposition auf ein Objekt, das Heranfahren an dieses, die Anpassung der Visualisierungsparameter des Objektes sowie die Manipulation oder Interaktion mit der Szene (z.B. durch Einbringen von Schnittmarken, Abmessen von Abständen, ...). Während eine geeignete Rotationstechnik die freie Exploration der Szene gut unterstützen kann, stellen näher definierte Interaktionsaufgaben (z.B.: das Heranfahren und Betrachten von einem Objekt nach dem anderen) weiterhin eine meist sehr komplexe Herausforderung dar.

Um Einsatz und Handhabung von 3D Visualisierungen für potentielle Anwender intuitiv und zweckgemäß zu gestalten, werden folgende Ziele näher verfolgt:

- Analyse von Anforderungen an zweckmäßige und steuerbare Rotationstechniken,
- Vergleich und Evaluation von existierenden mausgesteuerten Rotationstechniken, und
- Unterstützung zielgerichteter dreidimensionaler Navigations- und Explorationsaufgaben

Hierbei fokussiert sich die Betrachtung auf das *Orbitting* bzw. Rotieren der virtuellen Kamera um eine 3D-Szene. Das *Zoomen* bzw. Heranfahren an Objekte sowie das Verschieben der Ansicht bzw. des aktuellen *look-at*-Punktes werden dabei in den folgenden Untersuchungen nicht mit betrachtet. Dies ist möglich, da das Rotieren, Zoomen und Verschieben bei allen für diese Arbeit relevanten Interaktionstechniken durch Gedrückthalten der verschiedenen Maustasten und alternativ über Drücken zusätzlicher Steuerungstasten auf der Tastatur realisiert wird. Folglich stellen hierbei Rotation, Zoomen und Verschieben unterschiedliche Interaktionsmodi dar, welche separat ausgeführt werden. Auch bei beidhändigen (*bimanual*) Interaktionstechniken zur Exploration von 3D-Szenen ist es üblich, die Rotation der Szene von einer Hand und das Zoomen und Verschieben der Ansicht von der anderen Hand steuern zu lassen [Hinckley et al. 1998; Xia et al. 2007].

Nur bei der Verwendung von 3D-Eingabegeräten mit sechs Freiheitsgraden ist die simultane einhändige Steuerung von Rotation, Zoomen und Verschieben möglich. 3D-Eingabegeräte sind jedoch, trotz bereits langjähriger Verfügbarkeit, noch immer wenig verbreitet. Ein potentiell ausschlaggebender Grund hierfür ist, dass bei 3D-Eingabegeräten unvermeidbar der einfache Zeigecharakter oder der flexible Ruhepunkt, wie sie bei einer Computermouse vorhanden sind, verloren gehen. Erstes trifft dabei auf 3D-Eingabegeräte, wie die SpaceMouse und den neueren SpaceNavigator<sup>1</sup> zu, welche nur die Eingabe einer Richtung und eines Betrages

---

<sup>1</sup><http://www.3dconnexion.de/>

(*rate-based input*) erlauben. Letzteres trifft auf mittels 3D-Tracking verfolgte *Handles* als 3D-Eingabegeräte zu, welche in einer gewählten Position gehalten werden müssen und nicht einfach losgelassen bzw. abgelegt werden können, ohne die Ansicht zu verändern [Zhai 1998; Ware und Rose 1999]. Beides macht dabei 3D-Eingabegeräte unter anderem nur bedingt für Standardinteraktionen, z.B. zur Bedienung und Steuerung einer 2D-GUI, praktikabel.

Entsprechend bietet es sich an, 3D-Eingabegeräte von der Untersuchung auszuschließen und die Steuerung von 3D-Rotationen separat zu untersuchen. Zusätzlich wird die Untersuchung beidhändiger Interaktion zur Rotation und zum Zoomen und Verschieben ausgeschlossen, um den Interaktionsaspekt des 3D-Rotierens separat untersuchen zu können. Die separate Rotation der Szene in einer Hand bei beidhändigen Interaktionsmetaphern legitimiert dieses Vorgehen, obwohl Wechselwirkungen bei simultaner beidhändiger Interaktion nicht ausgeschlossen werden können. Für Untersuchungen von 3D-Eingabegeräten und beidhändiger (*bimanual*) Interaktion sei vergleichend auf [Hinckley et al. 1998; Ritter 2005; Rhijn und Mulder 2006; Xia et al. 2007] verwiesen.

### 8.1.2 Anforderungsanalyse

Für die Evaluation der Rotationstechnik stellt sich die systematische Exploration einer komplexen 3D-Szene (zum Beispiel beim Suchen nach einem Zielobjekt) in Verbindung mit dem Rotieren eines gefundenen Zieles in das Zentrum des 3D-Views (zum Beispiel zur weiteren Inspektion) als wichtigste Aufgabe heraus. Bei der Leberoperationsplanung und der Planung von Neck Dissections ist dieses Vorgehen die Hauptinteraktionsform zur Beurteilung von Lagebeziehungen z.B. von Tumoren, Lymphknoten und Gefäßen (vergleiche Abschnitt 2.2 und 2.3).

Wie aus den in Kapitel 2 betrachteten Anwendungen in der chirurgischen Ausbildung und Interventionsplanung verallgemeinert werden kann, besteht ein grundsätzliches medizinisches Anwendungsszenario daraus, dreidimensionale medizinische Szenen systematisch zu explorieren (zum Beispiel zum Betrachten und Beurteilen jedes Teils bzw. Abschnittes eines Organs). Entsprechend sollten Interaktionstechniken mit einem vorhersehbaren Verhalten von Vorteil sein, welches das Abschätzen von Effekten einer Interaktion vorweg erlaubt sowie das einfache Zurückkehren zu einer vorherigen Blickposition ermöglicht.

Basierend auf diesen Beobachtungen können die folgenden vier allgemeingültigen Anforderungen als wichtigste Voraussetzung für eine vorhersehbare und benutzerfreundliche Rotationstechnik benannt werden [Bade et al. 2005]:

Anforderung 1: *Gleiche Aktion - gleiche Reaktion.*

Anforderung 2: *Die Rotationsrichtung sollte der Eingabebewegungsrichtung ähneln.*

Anforderung 3: *Die 3D-Rotationsinteraktion sollte transitiv und kommutativ sein.*

Anforderung 4: *Die Rotationsgeschwindigkeit sollte an individuelle Bedürfnisse anpassbar sein.*

Die erste Anforderung stellt eine grundsätzliche Regel in der Mensch-Computer-Interaktion dar, welche die Wichtigkeit eines vorhersagbaren und reproduzierbaren Interaktionsverhaltens unterstreicht. Beispielsweise sollte sich bei der gleichen Bewegung mit der Maus der Mauszeiger nicht mal in die eine Richtung und ein anderes Mal in eine andere Richtung bewegen.

Anforderung 2 trägt der *Stimulus-Response* (S-R) Kompatibilität und der kinästhetischen Übereinstimmung [Britton et al. 1978] zwischen der Richtung einer Anwenderaktion (z.B.

Mausbewegung) und der Richtung der Reaktion des Computers (z.B. Rotation eines Objektes) Rechnung.

Anforderung 3 ergänzt die ersten beiden Anforderungen um eine Berücksichtigung des natürlichen transitiven und kommutativen Verhaltens von Bewegungen im euklidischen Raum. Da eine Bewegung mit einem Eingabegerät wie der Maus von einem Startpunkt  $A$  zu Punkt  $B$  und dann zu Punkt  $C$  an der gleichen Position endet, wie eine direkte Bewegung von  $A$  nach  $C$ , sollte dies auch für eine Reaktion auf eine solche Aktion gelten. Im Zusammenhang mit der Steuerung von 3D-Rotationen scheint diese Eigenschaft entscheidend, um Anwendern ein einfaches Zurückkehren zu einer initialen Blickposition zu ermöglichen.

Während die ersten drei Anforderungen wichtig für ein vorhersagbares Verhalten sind, berücksichtigt Anforderung 4 individuelle Anwender- und Anwendungsbedürfnisse. Das Anpassen des Verhältnisses zwischen Eingabegeschwindigkeit und resultierender Geschwindigkeit (*control-to-display (C/D) ratio* [Chapanis und Kinkade 1972]) ist nötig, um den besten Kompromiss zwischen Schnelligkeit und Genauigkeit entsprechend den Anwender- und Anwendungsbedürfnissen zu finden. Das Einstellen dieses Verhältnisses ist deshalb entscheidend für die Effizienz, Genauigkeit und Anwenderzufriedenheit.

Die zusammengetragenen Anforderungen spannen einen formellen Rahmen auf, um existierende Rotationstechniken systematisch auf ihr Verhalten und ihre Eigenschaften hin zu untersuchen. Im nachstehenden Abschnitt wird dazu der Stand der Technik von existierenden Rotationstechniken analysiert und verglichen.

## 8.2 3D-Rotationstechniken – Stand der Technik

In diesem Abschnitt werden existierende Rotationstechniken, die den aktuellen Stand der Forschung darstellen, analysiert und verglichen. Die Anforderungsliste aus Abschnitt 8.1.2 wird dabei verwendet, um die untersuchten Rotationstechniken formal zu klassifizieren und ihre Vor- und Nachteile aufzuzeigen. Betrachtet werden hierbei alle relevanten Rotationstechniken, die eine direkte Manipulation mit einer Standard-PC-Maus ermöglichen. In der Literatur werden diese Techniken meist als *Virtual Trackball (VT)* bezeichnet.

Die hier vorgenommene Untersuchung der Rotationstechniken ist auf Aspekte der Anwendbarkeit und Benutzerfreundlichkeit fokussiert. Detaillierte Beschreibungen der zugrunde liegenden Mathematik sowie von Implementierungsdetails der besprochenen Rotationstechniken würden den Rahmen der Arbeit sprengen, können jedoch aus den zitierten Publikationen entnommen werden.

### 8.2.1 Chen's Virtual Trackball

*Chen's Virtual Trackball (Chen's VT)*, auch als *Virtual Sphere* bezeichnet [Chen et al. 1988], kann als dreidimensionale Kugel, die sich hinter dem Bildschirm befindet, aufgefasst werden. Die zweidimensionale Bildschirmposition der Maus wird auf diese Kugel projiziert, um einen korrespondierenden 3D-Punkt auf der Kugeloberfläche zu erhalten. Eine 2D-Eingabebewegung von einem 2D-Punkt zu einem zweiten, wird dabei auf die Rotation von einem auf die Kugel projizierten 3D-Punkt zum zweiten abgebildet. Für eine detaillierte mathematische Beschreibung sei auf [Henriksen et al. 2004] verwiesen.

Jedoch weist das von Chen et al. [1988] auf 2D-Berechnungen zur Bestimmung von Rotationsachse und Winkel vereinfachte mathematische Verfahren Inkonsistenzen auf. So sind

resultierende Rotationsachsen nicht immer senkrecht zum 2D-Verschiebungsvektor der Maus [Henriksen et al. 2004]. Da sowohl die Implementierungen von Bell's VT [Bell 1988] als auch Shoemake's VT [Shoemake 1992] (beide werden nachstehend noch besprochen) einen identischen Ansatz verfolgen und über 3D-Projektionsberechnungen diese Inkonsistenzen beheben, soll Chen's VT hier nicht näher betrachtet werden.

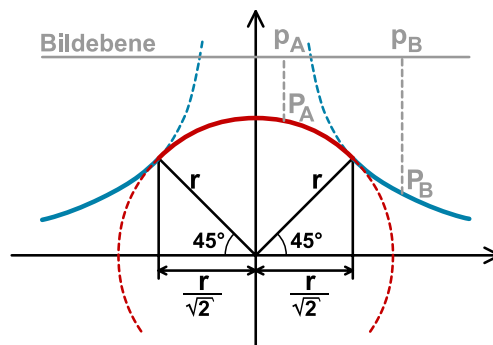
### 8.2.2 Bell's Virtual Trackball

Die von Bell [1988] implementierte Rotationstechnik (Bell's VT oder auch *Virtual Sphere*) stellt eine Erweiterung von Chen's VT [Chen et al. 1988] dar. Anstatt die 2D-Mausposition auf eine Kugel zu projizieren, verwendet Bell eine Kombination aus Kugel und dreidimensionaler hyperbolischer Parabel, auf die die Mauspositionen abgebildet werden (siehe Abbildung 8.2). Entsprechend kann die Szene mit dieser Technik gleichmäßig und fließend gesteuert werden. Mit Blick auf die aufgestellte Anforderungsliste (S. 110) erfüllt Bell's VT jedoch nur Anforderung 2.

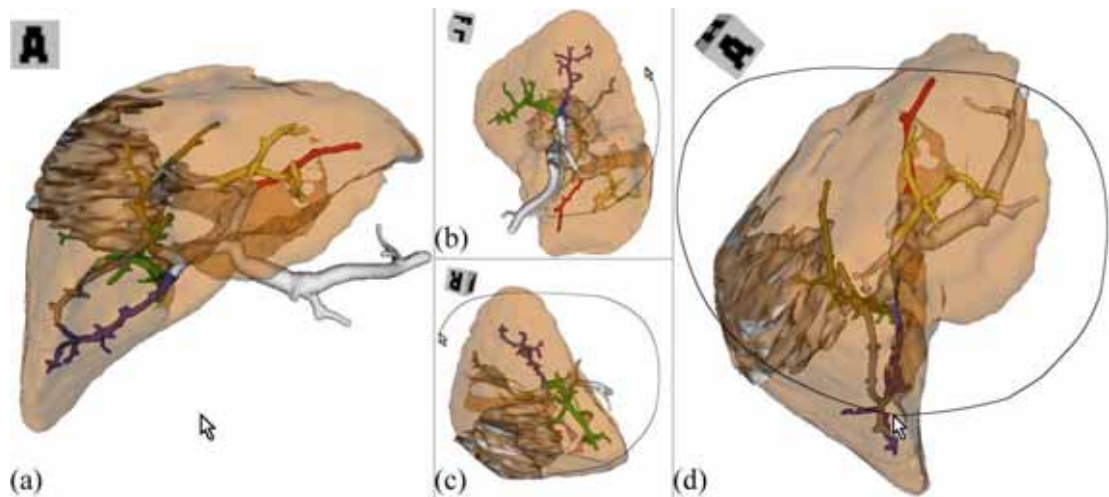
Eine Mausbewegung mit gleichem Bewegungsvektor ( $\Delta x, \Delta y$ ), begonnen an unterschiedlichen Positionen des 3D-Anzeigebereiches (3D-Views), bringt unterschiedliche Rotationen hervor. So rotiert zum Beispiel eine horizontale Mausbewegung im Zentrum des 3D-Views die Szene um den *up-vector* der virtuellen Kamera. Im Gegensatz dazu rotiert die gleiche horizontale Mausbewegung am unteren Rand des 3D-Views die Szene um die Sichtachse (*viewing-axis*) der virtuellen Kamera. Entsprechend können selbst einfache horizontale oder vertikale Mausbewegungen eine unvorhergesehene Rotation bewirken. Bell's VT erfüllt damit Anforderung 1 nicht.

Auch wenn die gleiche Aktion nicht die gleiche Reaktion hervorruft und sich die Rotationsrichtung je nach Mausposition im 3D-View ändert, bleibt die resultierende 3D-Rotation ähnlich der 2D-Mausbewegungsrichtung. Beispielsweise rotiert eine nach rechts gerichtete Mausbewegung am oberen Rand des 3D-Views die Szene nach rechts (im Uhrzeigersinn), während die gleiche Bewegung nach rechts am unteren Rand des 3D-Views die Szene nach links (gegen den Uhrzeigersinn) dreht. Dieses Verhalten verletzt wiederum Anforderung 1, erfüllt aber Anforderung 2.

Im Widerspruch zu Anforderung 3 ist die Kombination von zwei Rotationen mit Bell's VT nicht transitiv und auch nicht kommutativ. Unter anderem bewirkt infolgedessen eine Mausbewegung entlang einer kreisförmigen Bahn, dass die 3D-Szene zu kippen bzw. zu „trudeln“ anfängt. Aus demselben Grund führt, wie in Abbildung 8.3 illustriert, eine geschlossene Kreisbewegung mit der Maus nicht zu dem erwarteten geschlossenen Kreis der Rotation. Aber



**Abbildung 8.2:** Querschnitt der Projektionsgeometrie von Bell's VT, um 2D-Bildschirmpunkte in den 3D-Raum abzubilden.



**Abbildung 8.3:** Verwendung von Bell's VT zum Rotieren einer Leber mit Tumor und Blutgefäßen. (a)-(d) Zeigen aufeinander folgende Zeitschritte der Interaktion. Mit der Maus wird ein geschlossener Kreis beschrieben (schwarze Linie), wobei die resultierende Rotation keinen geschlossenen Kreis ergibt (vergleiche (a) und (d)).

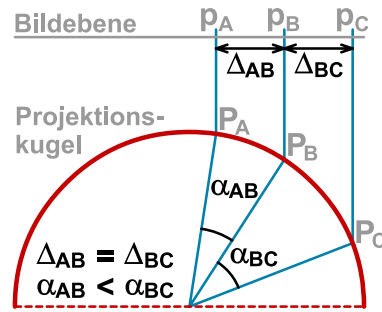
wenigstens ähnelt auch in diesem Fall die Rotationsrichtung der Mausbewegungsrichtung. So rotiert eine Kreisbewegung der Maus im Uhrzeigersinn die 3D-Szene auch im Uhrzeigersinn.

Bedingt durch die Bell's VT zugrunde liegende Projektion der 2D-Mausposition auf eine virtuelle 3D-Geometrie ist es nicht möglich, die Rotationsgeschwindigkeit (Anforderung 4) zufriedenstellend anzupassen. Die Rotation mit Bell's VT ist dabei auf  $90^\circ$  vom Zentrum des 3D-Views zu dessen Rand hin beschränkt. Dieser fest vorgegebene Rotationsbetrag macht die Rotationsgeschwindigkeit von der aktuellen Größe des 3D-Views abhängig. Entsprechend verändert eine größere bzw. kleinere Anzeige der 3D-Szene auf dem Bildschirm die Rotationsgeschwindigkeit. Je kleiner dabei der 3D-View wird, desto schneller wird die Szene rotiert. Diese Beschränkung des Rotationsbetrages kann durch Einbringen eines Skalierungsfaktors in die Berechnung der Rotationswinkel aufgehoben werden. Dies führt jedoch zu sehr unterschiedlichen Verhaltensweisen der Rotationstechnik, was in Abschnitt 8.2.3 näher diskutiert wird.

Trotz dieser Probleme (nur Anforderung 2 ist erfüllt) ist Bell's VT Bestandteil vieler 3D-Applikationen, insbesondere, weil Bell's VT als Rotationstechnik im Open Inventor Toolkit [SGI 2007; Strauss und Carey 1992] integriert ist.

### 8.2.3 Shoemake's Virtual Trackball

Shoemake's VT [Shoemake 1992], auch *Arcball* genannt, ist eine spezielle Variante von Chen's VT (siehe [Henriksen et al. 2004]). Shoemake [1992] projiziert ebenfalls die 2D-Mausposition auf eine Kugel, um Rotationsachse und -winkel zu bestimmen. Durch einen sorgfältig ausgewählten Skalierungsfaktor bei der Berechnung des Rotationswinkels sind Rotationen mit Shoemake's VT jedoch transitiv und kommutativ. Entsprechend ist Anforderung 3 erfüllt. Ähnlich zu Bell's VT wird auch Anforderung 2 erfüllt und Anforderung 1 nicht. Im Gegensatz zu Bell's VT wird jedoch bei Shoemake's VT die 3D-Szene zweimal um die Sichtachse rotiert, wenn die Maus einmal um die Projektionskugel herum bewegt wird. Dies kann zu Irritationen führen und verletzt leicht Anforderung 2.



**Abbildung 8.4:** Diskontinuitäten der Rotationswinkel bei Shoemake's VT am Rand der Projektionskugel: Der Abstand  $\Delta_{AB}$  zwischen den Bildschirmpunkten  $p_A$  und  $p_B$  entspricht dem Abstand  $\Delta_{BC}$  zwischen  $p_B$  und  $p_C$ . Dennoch ist der resultierende Rotationswinkel  $\alpha_{AB}$  zwischen den projizierten Punkten  $P_A$  und  $P_B$  geringer, als der Winkel  $\alpha_{BC}$  zwischen  $P_B$  und  $P_C$ .

Aufgrund der zugrunde liegenden Mathematik (Projektion auf virtuelle 3D-Kugel) ist es mit Shoemake's VT ebenfalls nicht möglich, die Rotationsgeschwindigkeit zufriedenstellend anzupassen (Anforderung 4). Je kleiner der 3D-View, desto schneller wird auch hier die Szene rotiert. Um dem entgegenzuwirken, könnte man entgegengesetzt proportional zur Größe des 3D-Views die Größe der Projektionskugel anpassen. Dies würde jedoch entweder zu großen Bereichen im 3D-View, in denen nur um die Sichtachse rotiert werden kann (bei sehr kleiner Projektionskugel), oder zu einer Projektionskugel größer als der 3D-View führen, was die Rotation um alle Achsen behindern würde. Weiterhin ist ähnlich zu Bell's VT die Rotation bei Shoemake's VT auf  $180^\circ$  vom 3D-View Zentrum zum Rand hin beschränkt. Wie bei Bell's VT bereits erwähnt, könnte dies durch Einführen eines Skalierungsfaktors bei der Berechnung des Rotationswinkels aufgehoben und die Rotationsgeschwindigkeit frei eingestellt werden. Jedoch würde zum Beispiel ein Faktor, der die Rotationsgeschwindigkeit halbiert, dazu führen, dass sich das Verhalten von Shoemake's VT an Chen's VT und Bell's VT angleicht. Entsprechend würden die Transitivität und Kommutativität (Anforderung 3) verloren gehen. Bei Shoemake's VT verändert sich weiterhin durch die verwendete Projektionskugel die Rotationsgeschwindigkeit vom Zentrum zum Rand der Kugel hin. Abbildung 8.4 illustriert diese Verzerrung durch die Projektion. Für Anwender von Shoemake's VT bedeutet dies, dass die Rotation am Rand der Kugel unter Verlust der Benutzerkontrolle und der Genauigkeit einzurasten scheint.

Ogleich die Implementierung von Shoemake's VT (siehe [Hultquist 1990]) einfacher als die von Bell's VT ist und Shoemake's VT immerhin eine Anforderung mehr erfüllt, ist diese Rotationstechnik nicht so weit verbreitet.

#### 8.2.4 Two-Axis Valuator Trackball

Beim Two-Axis Valuator Trackball (Two-Axis VT) (siehe [Chen et al. 1988]) wird die horizontale Mausbewegung auf eine Rotation um den *up-vector* der virtuellen Kamera abgebildet und die vertikale Mausbewegung auf eine Rotation um den *lateral-vector* (Vektor senkrecht zu *up-vector* und *view-vector* der Kamera). Diagonale Mausbewegungen werden auf eine Kombination von Rotationen um beide Achsen abgebildet. Entsprechend ist keine explizite Rotation um den *view-vector* möglich. Meist kann diese dann separat über das Drücken einer anderen Maustaste bzw. einer Steuerungstaste auf dem Keyboard gesteuert werden.

Im Gegensatz zu allen anderen hier betrachteten Rotationstechniken bewirkt beim Two-Axis VT die gleiche Mausbewegung immer die gleiche Rotation. Entsprechend ist der Two-

Axis VT die einzige Rotationstechnik, die Anforderung 1 erfüllt. Jedoch sind mit dem Two-Axis VT die resultierenden Rotationen nicht transitiv und auch nicht kommutativ (Anforderung 3). Weiterhin ähnelt die Rotationsrichtung der Mausbewegungsrichtung bis auf den Spezialfall von kreisenden Mausbewegungen. Beschreibt die Maus zum Beispiel eine Kreisbahn im Uhrzeigersinn, so dreht sich die Szene um den view-vector entgegen dem Uhrzeigersinn. Damit ist Anforderung 2 grundlegend erfüllt, wird aber in diesem speziellen Fall verletzt.

Durch die direkte Abbildung der 2D-Mausbewegung auf eine korrespondierende Rotation um die benannten zwei Achsen der virtuellen Kamera kann die Rotationsgeschwindigkeit einfach durch einen Skalierungsfaktor an persönliche Bedürfnisse und die Anforderungen einer Applikation angepasst werden. Entsprechend erfüllt der Two-Axis VT Anforderung 4.

Der Two-Axis VT wird weit verbreitet in Applikationen, wie zum Beispiel dem Visualization Toolkit (VTK) [Kitware Inc 2007] eingesetzt. Dies mag zum einen an der einfachen Implementierung liegen, aber zum anderen auch dadurch begründet sein, dass der Two-Axis VT drei der vier Anforderungen (siehe Abschnitt 8.1.2), mit kleinen Abstrichen in Bezug auf Anforderung 2, erfüllt.

### 8.2.5 Two-Axis Valuator with Fixed Up-vector

Für 3D-Modellierungsaufgaben (z.B. in 3D-Studio-Max [Autodesk 2007]) oder auch für verwandte Aufgaben (z.B. in Deep Exploration [Right Hemisphere 2007]) wird oft eine spezielle Variante des Two-Axis VT verwendet. Diese Variante benutzt den up-vector der 3D-Szene (*world-up-vector*) für die horizontale Rotation. Da dieser Vektor unveränderlich ist, wird die Rotationstechnik als „Two-Axis Valuator with fixed up-vector“ oder hier kürzer als „Fixed Two-Axis VT“ bezeichnet.

Die Verwendung des Fixed Two-Axis VT bietet die Vorteile von transitiven und kommutativen Rotationen – Anforderung 3 ist erfüllt. Leider führt das dazu, dass die resultierende Rotation auf die gleiche Mausbewegung jeweils von der aktuellen Rotation der Szene abhängt und somit Anforderung 1 nicht mehr erfüllt ist. Zusätzlich verhält sich die Rotationsrichtung gegensätzlich zur Mausbewegungsrichtung, sobald die 3D-Szene auf den Kopf rotiert wird. Entsprechend wird Anforderung 2 nicht erfüllt. Äquivalent zum Two-Axis VT ist auch beim Fixed Two-Axis VT durch die direkte Abbildung der 2D-Mausbewegung Anforderung 4 erfüllt.

Wie bereits erwähnt, findet der Fixed Two-Axis VT in existierenden 3D-Modellierungsprogrammen Anwendung. Dies mag durch das transitive und kommutative Verhalten der Rotation und die anpassbare Rotationsgeschwindigkeit begründet sein, da beides wichtige Voraussetzungen für das nötige Maß an Kontrolle und Genauigkeit in solchen Applikationen sind.

### 8.2.6 Zusammenfassung

Aus Abschnitt 8.2.1–8.2.5 ist ersichtlich, dass jede Rotationstechnik ein anderes Verhalten mit unterschiedlichen Eigenschaften (wie z.B. Transitivität, Kommutativität oder direkter *Stimulus-Response* (S-R) Kompatibilität und kinästhetischer Übereinstimmung) vorweist. In Tabelle 8.1 werden noch einmal alle Rotationstechniken bezüglich der jeweils erfüllten Anforderungen gegenübergestellt.

Zusammenfassend erfüllt keine der Rotationstechniken alle Anforderungen aus Abschnitt 8.1.2. Der Two-Axis VT erfüllt jedoch mit drei bzw. zweieinhalb von vier die meisten

Anforderungen gefolgt von Shoemake's VT und dem Fixed Two-Axis VT mit jeweils zwei und Bell's VT mit nur einer erfüllten Anforderung. Dies lässt darauf schließen, dass der Two-Axis VT am geeignetsten für die Exploration von 3D-Szenen in der medizinischen Ausbildung und Interventionsplanung ist. Da dies jedoch nur Ergebnisse einer theoretischen Analyse sind, soll im nächsten Abschnitt eine empirische Studie den Einfluss der untersuchten Faktoren und damit die Eignung der verschiedenen Rotationstechniken näher evaluieren.

**Tabelle 8.1:** Erfüllte Anforderungen der einzelnen Rotationstechniken.

Rotations- technik:	Bell's VT	Shoemake's VT	Two-Axis VT	Fixed Two-Axis VT
Anforderung 1	–	–	+	–
Anforderung 2	+	+	+ / –	–
Anforderung 3	–	+	–	+
Anforderung 4	–	–	+	+

(Anforderungsliste siehe S. 110)

## 8.3 Evaluation von Rotationstechniken

In Abschnitt 8.2 wurden existierende Rotationstechniken zusammengetragen und unter Berücksichtigung der aus medizinischen Anwendungen abgeleiteten Anforderungen (siehe Abschnitt 8.1.2) analysiert und verglichen. Auf diese Weise konnten die Eigenschaften und Probleme der verschiedenen Techniken identifiziert und bestimmte Probleme bei der Bedienung der jeweiligen Rotationstechnik aufgedeckt werden. Dennoch ist deren Einfluss auf Aspekte der Bedien- und Nutzerfreundlichkeit, wie Effizienz, Genauigkeit und Anwenderzufriedenheit, weitgehend unbekannt.

Im Folgenden werden existierende Evaluationsstudien von Rotationstechniken besprochen und ihre Defizite zur Eignungsabschätzung der Rotationstechniken für die Exploration medizinischer 3D-Szenen und Objekte offengelegt. Entsprechend wird anschließend das Konzept einer eigenen Evaluationsstudie vorgestellt und die Ergebnisse der Studie präsentiert (siehe auch [Bade et al. 2005]). Abschließend wird die Untersuchung der Rotationstechniken zusammenfassend diskutiert.

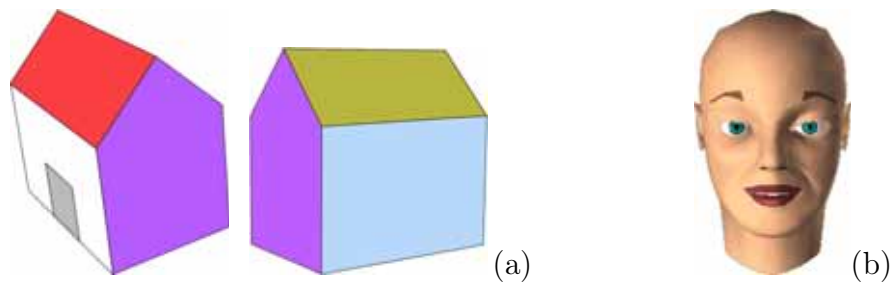
### 8.3.1 Vorarbeiten und Stand der Forschung

Nur vier Studien [Chen et al. 1988; Jacob und Oliver 1995; Hinckley et al. 1997; Partala 1999] sind in der Literatur zu finden, die bisher Rotationstechniken empirisch evaluiert haben [Henriksen et al. 2004]. Drei dieser Studien [Chen et al. 1988; Hinckley et al. 1997; Partala 1999] überprüfen dabei nur relativ einfache Rotationsaufgaben. In diesen sog. „*orientation matching*“-Aufgaben wurden die Probanden aufgefordert, ein Objekt in eine geforderte Zielorientierung zu rotieren, die durch ein Bild des Objektes in der gewünschten Orientierung vorgegeben war. Nur Jacob und Oliver [1995] überprüften darüber hinaus auch Inspektionsaufgaben, bei der die Probanden unter anderem die Anzahl der Fenster im Modell eines Hauses identifizieren sollten. Dabei konnten sie Unterschiede in der Eignung der verschiedenen Rotationstechniken für die beiden unterschiedlichen Aufgabengruppen identifizieren.



Die Unterschiede konnten dabei darauf zurückgeführt werden, dass die Inspektionsaufgaben im Gegensatz zu den „*orientation matching*“-Aufgaben eine geringere Genauigkeit der Rotationstechnik erforderten und eine schnelle, wenn auch nicht unbedingt exakte Rotation zur Lösung der Aufgabe von Vorteil war. Weiterhin ist anzumerken, dass alle vier Studien nur einfache, leicht zu erfassende Objekte bzw. 3D-Szenen verwendet haben. So wurde entweder das Modell eines einfachen Hauses (Abb. 8.5 a) bzw. eines simplifizierten Kopfes verwendet, bei denen die jeweilige Orientierung (oben, unten, rechts, links) explizit gegeben ist. Dies unterscheidet sich jedoch stark von medizinischen Anwendungen zum Beispiel für die Interventionsplanung und chirurgische Ausbildung, da hier meist Objekte (z.B. ein Tumor) keine eindeutigen Hinweise auf die Orientierung des Objektes beinhalten (siehe Abb. 8.5 b).

Da Jacob und Oliver [1995] Eignungsunterschiede der Rotationstechniken bei unterschiedlichen einfachen Aufgaben feststellen konnten und alle Studien nur einfache Objekte mit gegebener Orientierung verwendeten, ist die Eignung der jeweiligen Rotationstechniken für komplexe Aufgaben in der Interventionsplanung und chirurgischen Ausbildung trotz der bisherigen Evaluationsergebnisse offen. Entsprechend sind die vorgestellten Rotationstechniken für komplexere, anwendungsnahe Aufgaben empirisch zu evaluieren. Dies soll helfen, die Rotationstechniken im gegebenen Kontext entsprechend ihrer Eignung beurteilen und angemessen einsetzen zu können.



**Abbildung 8.5:** Einfache Objekte zur Evaluation der Rotationstechniken mit explizit gegebener Orientierung: (a) 3D-Haus wie von [Chen et al. 1988; Jacob und Oliver 1995; Hinckley et al. 1997] genutzt, (b) 3D-Kopfmodell genutzt von [Partala 1999] (modifiziert).

### 8.3.2 Evaluationskonzept

Das Ziel der Evaluationsstudie ist es, die in Abschnitt 8.3.1 diskutierten Rotationstechniken in Bezug auf intuitive Nutzbarkeit, Effizienz und Anwenderzufriedenheit bei komplexen „*scan-and-hit*“-Aufgaben mit komplex geformten Objekten zu untersuchen und zu vergleichen. Die Ergebnisse sollen entsprechend zur Identifikation einer geeigneten Rotationstechnik beitragen sowie Ansätze für weitere Untersuchungen und Entwicklungen aufzeigen.

Da sich die Evaluation unter anderem auf die intuitive Nutzbarkeit der Rotationstechniken konzentriert, werden den Probanden keine Instruktionen zur jeweiligen Benutzung der unterschiedlichen Rotationstechniken gegeben. Dafür können die Rotationstechniken aber vor der jeweils zu erfüllenden Aufgabe frei ausprobiert und erlernt werden. Informationen und Instruktionen zur Durchführung sowie das Evaluationsprogramm als Download wurden für die Probanden der Evaluation auf einer Webseite bereitgestellt.

## Versuchsaufbau

Die Evaluation wurde von jedem Probandem am ihr/ihm eigenen Computer durchgeführt. Entsprechend wurden die Versuchsteilnehmer nicht mit ungewohnten Eingabegeräten und fremden Einstellungen für die Eingabeperformanz konfrontiert. Für die Evaluation wurden alle zuvor näher diskutierten Rotationstechniken unter Verwendung der *world-in-hand* Metapher verwendet. Das Evaluationsprogramm mit allen Rotationstechniken und integriertem Fragebogen zur Evaluation der Nutzerzufriedenheit und Selbsteinschätzung wurde in Macromedia Director 8.5 [Adobe 2007c] implementiert und als selbstlauffähiges Windows-Programm exportiert.

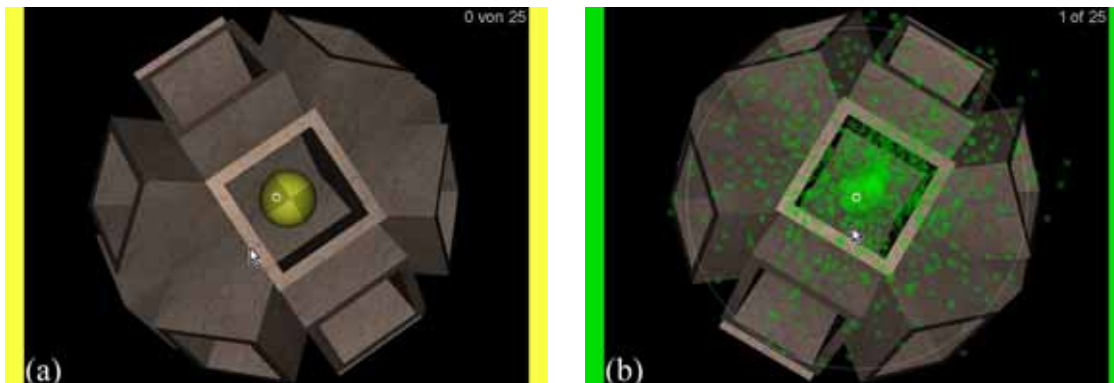
## Aufgabe

Für die Evaluation der Rotationstechniken stellt (1) die systematische Exploration einer komplexen 3D-Szene (zum Beispiel beim Suchen nach einem Zielobjekt) sowie (2) das Rotieren eines gefundenen Zieles in das Zentrum des 3D-Views (zum Beispiel zur weiteren Inspektion) in Kombination die wichtigste Aufgabe dar. Entsprechend hatten die Probanden eine Abfolge von *scan-and-hit* Aufgaben zu absolvieren.

Die Probanden wurden mit einem komplex geformten symmetrischen Objekt, das nur bedingt Schlüsse über die aktuelle Orientierung zuließ, konfrontiert (Abb. 8.6). Dieses Objekt musste durch Rotieren nach Zielen abgesucht werden. Wenn das Ziel gefunden wurde, musste es ins Zentrum des 3D-Views rotiert und dort durch Drücken der Leer- oder Entertaste abgeschossen werden (siehe Abb. 8.6). Weiterhin wurden die Probanden instruiert, dass die Geschwindigkeit entscheidend ist.

Ein Ziel konnte dabei nur abgeschossen werden, wenn sich dieses im Zentrum des 3D-Views unter einer kreisförmigen Markierung befand. Die erlaubte Fehlertoleranz vom Zentrum des 3D-Views zum Zentrum des Zieles betrug dabei  $13^\circ$ . Die Anzahl an Schüssen, die benötigt wurde, um alle Ziele abzuschießen, wurde als Maß für die Ungenauigkeit beim Erfüllen der Aufgabe mit erhoben.

Nach dem Abschießen eines Schusses wurde visuelles und auditives Feedback vom System gegeben, um das Treffen des Zieles (Explosion des Zieles) oder das Verfehlen desselbigen anzuzeigen. Wird das Ziel verfehlt, so wird ein Einschlag des Schusses auf dem zu rotierenden Objekt angezeigt, das Zielobjekt wird nicht entfernt und kann wiederholt anvisiert werden.



**Abbildung 8.6:** Screenshots vom Evaluationsprogramm. Die 3D-Szene wird durch die Probanden rotiert (z.B. mit (a) Bell's VT und (b) Shoemake's VT), um ein Ziel zu suchen, ins Zentrum zu rotieren und da abzuschießen. Das Zentrum des 3D-Views ist dabei mit einem kleinen Kreis markiert.

## Design

Die Evaluation wurde in Form eines Spieles mit unterschiedlichen Leveln konzipiert. In jedem Level wird jeweils eine andere der besprochenen Rotationstechniken (siehe Abschnitt 8.2) bereitgestellt, um das verwendete komplexe 3D-Szenenobjekt nach 25 Zielen abzusuchen, diese ins Zentrum zu rotieren und abzuschießen. Die Level werden dabei in randomisierter Reihenfolge dargeboten, um den Lerneffekt von einem Level zum anderen im gesamten Probandenkollektiv zu streuen.

Mit Ausnahme der verwendeten Rotationstechnik sind alle Level gleich aufgebaut. In jedem Level ist jeweils ein Ziel nach dem anderen abzuschießen. Dazu wird ein Ziel zufällig in einem der 18 Slots des abzusuchenden Objektes (siehe Abb. 8.6) platziert. Nachdem das Ziel gefunden und abgeschossen wurde, erscheint (mit kurzer Verzögerung) das nächste Ziel an einer zufälligen Position, bis alle Ziele abgearbeitet sind.

## Versuchsablauf

Zu Beginn des Evaluationsprozesses hatten alle Probanden ihr Erfahrungsniveau und ihre Vertrautheit mit der Maus als Eingabegerät zwischen 1 (niedrig) und 7 (hoch) einzuschätzen. Darüber hinaus hatten alle Probanden ihre Vorerfahrungen mit 3D-Szenen am Computer in vier Kategorien: *3D-Modellierungsanwendungen*, *3D-Spielen*, *3D-Navigation* und *3D-Rotation* zwischen 1 (niedrig) und 7 (hoch) einzuschätzen. Danach wurde allen Probanden eine generelle Beschreibung des Versuchsablaufs präsentiert. Daraufhin waren die einzelnen Level in randomisierter Reihenfolge abzuarbeiten.

Vor dem Betreten eines neuen Levels wurde jeweils eine Trainingsszene präsentiert, um sich mit der Rotationstechnik des aktuellen Levels vertraut machen zu können und deren Geschwindigkeit (mit Hilfe eines Sliders) den eigenen Bedürfnissen anzupassen, sofern dies die Rotationstechnik zuließ. Zusätzlich wurden die Probanden instruiert, dass sie durch Drücken der linken Maustaste und gleichzeitiges Verschieben der Maus die Szene rotieren können und dass mit der Leertaste und Entertaste geschossen werden kann. Das Training war weder zeitlich, noch durch die Anzahl der Ziele beschränkt und konnte von den Probanden selbstständig beendet werden, sobald sie sich mit der aktuellen Rotationstechnik vertraut genug fühlten. Daraufhin beginnt das eigentliche Level, in dem die Probanden 25 Ziele so schnell wie möglich zu finden und abzuschießen hatten. Um verlässliche Werte für die benötigte Zeit zur Erfüllung der Aufgabe zu erhalten, war es den Probanden nicht erlaubt, die Rotationsgeschwindigkeit während der Bearbeitung des Levels zu verändern.

Zur Kombination von objektiven und subjektiven *Usability*-Faktoren hatten nach jedem Level die Probanden zu bewerten, wie gut sie mit der Rotationstechnik des aktuellen Levels die Aufgabe erfüllen konnten und wie gut die Rotationstechnik den eigenen Erwartungen folgte. Darüber hinaus wurden die Probanden über die von ihnen benötigte Zeit informiert und es wurden die Top-Ten der schnellsten Probanden präsentiert. Die gleiche Prozedur wurde dann bei allen weiteren Leveln (Rotationstechniken) wiederholt. Der gesamte Versuchsablauf dauerte ca. 30 Minuten.

## Probandenkollektiv

An der Evaluation nahmen 42 unbezahlte Probanden (30 Männer, 12 Frauen) erfolgreich teil. (Um einen Einfluss von Spezialeingabegeräten zu vermeiden, schließt dies nur Probanden ein, die eine Computermouse verwendet haben.) Alle Probanden besaßen Langzeiterfahrungen mit der Computermousesteuerung. Auf einer Skala von 1 (niedrig) bis 7 (hoch) schätzten

bis auf vier Probanden alle ihre Erfahrung mit 7 ein, während die vier Probanden ihre Erfahrung jeweils mit 5 oder 6 bewerteten. Im Gegensatz dazu besaßen alle Probanden sehr unterschiedliche Vorerfahrungen bei der 3D-Navigation und Rotation. Zur Auswertung von Unterschieden zwischen Probanden auf unterschiedlichem 3D-Erfahrungsniveau wird das Probandenkollektiv in eine 3D-erfahrene und eine 3D-unerfahrene Gruppe aufgeteilt. Zu den 3D-erfahrenen Probanden werden dabei diejenigen gezählt, die bei mindestens zwei der vier Kategorien zur Vorerfahrung mit 3D-Szenen am Computer ihre Kenntnisse auf der Skala von 1 bis 7 mit 5 oder besser bewerteten ( $n = 23$ ). Alle anderen Probanden werden zur Gruppe der 3D-unerfahrenen Probanden ( $n = 19$ ) zusammengefasst.

### 8.3.3 Statistische Analyse

Zur statistischen Auswertung wurde zunächst der *Kolmogorov-Smirnov-Test* verwendet, um die gegebenen Daten auf eine Normalverteilung hin zu untersuchen. Die Messdaten für die benötigte Zeit pro Level erwies sich dabei als normalverteilt ( $p \geq 0,05$ ). Die Anzahl der benötigten Schüsse, die Bewertung der eigenen Vorerfahrung sowie die Bewertung der Rotationstechniken zeigt jedoch eine signifikant von der Normalverteilung abweichende Verteilung ( $p < 0,05$ ). Um Unterschiede zwischen den einzelnen Rotationstechniken zu untersuchen, wurden Tests für abhängige Daten (*paired samples tests*) verwendet, da alle Probanden alle Rotationstechniken (Level) nacheinander abarbeiten mussten.

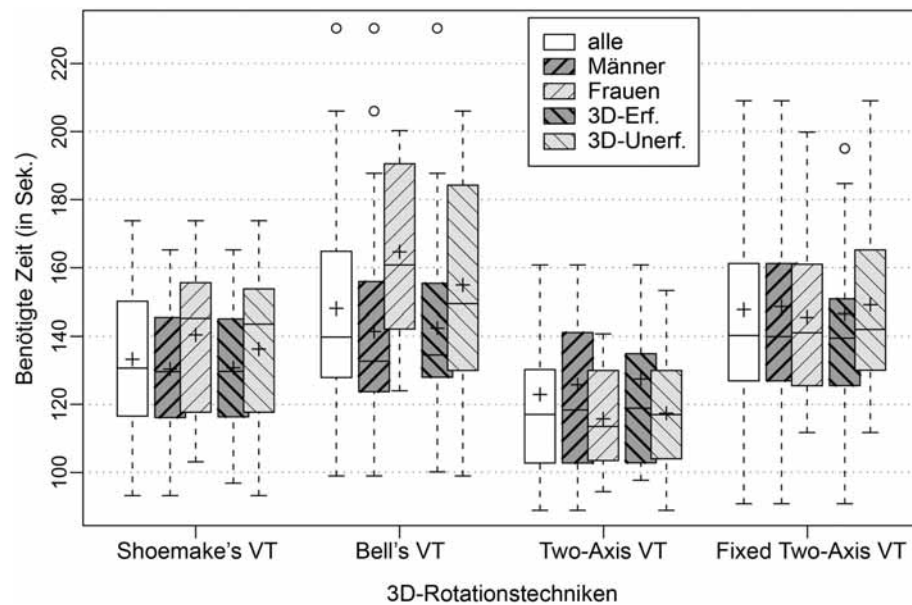
Um Unterschiede zwischen den Rotationstechniken bezüglich der benötigten Zeit zur Erfüllung der Aufgabe zu analysieren, wurde der *Paired Samples t-Test* verwendet. Um Unterschiede bezüglich der benötigten Schüsse zu analysieren, wurde der *Wilcoxon-Signed-Ranks-Test* angewendet. Zur Analyse der ordinal skalierten subjektiven Bewertungen der Rotationstechniken durch die Probanden wurde der *Sign-Test* angewendet.

### 8.3.4 Ergebnisse und Diskussion

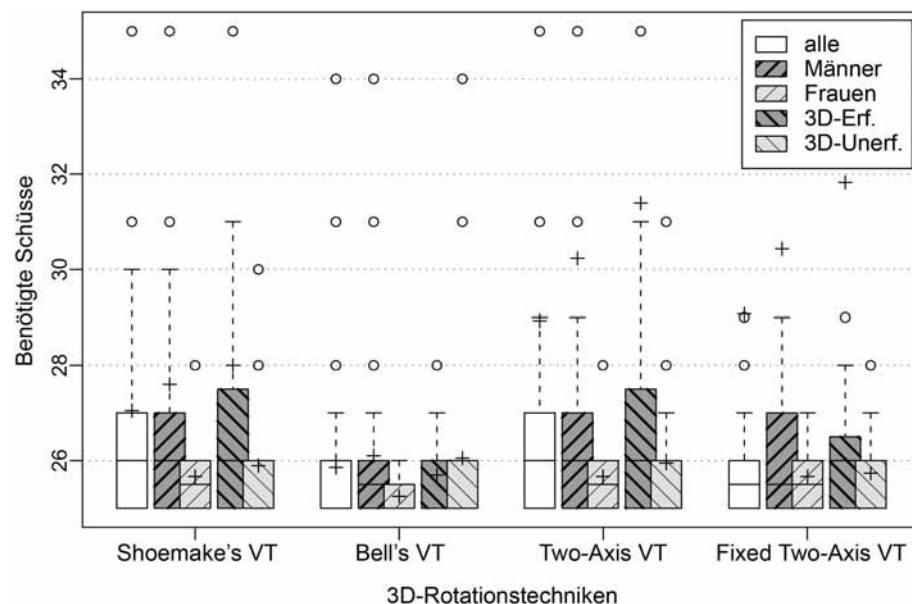
Tabelle 8.2 und Abbildung 8.7 stellen die Ergebnisse der Evaluation bezüglich der benötigten Zeit zur Erfüllung der gestellten „scan-and-hit“-Aufgaben dar. Im direkten Vergleich der Rotationstechniken stellt sich heraus, dass alle Probanden mit dem Two-Axis VT am schnellsten waren. Mit statistisch signifikantem Unterschied ( $p < 0,05$ ) zum Two-Axis VT erwies sich Shoemake’s VT als zweitschnellste Rotationstechnik. Mit beiden Techniken waren die Probanden dabei höchst signifikant ( $p < 0,001$ ) schneller als mit Bell’s VT und dem Fixed Two-Axis VT. Zwischen Bell’s VT und dem Fixed Two-Axis VT zeigte sich kein statistisch relevanter Unterschied bezüglich der benötigten Zeiten (Signifikanzwerte siehe Tabelle 8.3).

Mit Bell’s VT brauchten die Probanden signifikant ( $p < 0,05$ ) weniger Schüsse im Vergleich zu Shoemake’s VT und dem Two-Axis VT (Tab. 8.3, Abb. 8.8). Jedoch waren die Probanden mit Bell’s VT am langsamsten, so dass dies den Schluss zulässt, dass wegen der geringen zu erreichenden Geschwindigkeit kaum vorbeigeschossen wurde. Andere statistische Unterschiede bei den Rotationstechniken bezüglich der benötigten Schüsse konnten nicht beobachtet werden.

Abbildung 8.9 fasst die Ergebnisse der subjektiven Bewertung der Rotationstechniken durch die Probanden grafisch zusammen. Wie die Ergebnisse der Voranalyse (siehe Abschnitt 8.2) erwarten ließen, wurde der Two-Axis VT, der die meisten Anforderungen (siehe S. 110) erfüllt, am besten bewertet. Der Unterschied der subjektiven Bewertung, (1) wie komfortabel die Aufgabe mit der jeweiligen Rotationstechnik erfüllt werden konnte und (2) wie gut die Rotation den Erwartungen folgte, war dabei statistisch signifikant ( $p \leq$



**Abbildung 8.7:** Boxplots der benötigten Zeit zur Erfüllung der Aufgabe mit jeder Rotationstechnik. (Separate Gegenüberstellung von allen Probanden (n=42), den Männer (n=30), Frauen (n=12), 3D-erfahrenen (n=23) und 3D-unerfahren (n=19) Probanden.)



**Abbildung 8.8:** Boxplots der benötigten Schüsse zur Erfüllung der Aufgabe mit jeder Rotationstechnik. (Separate Gegenüberstellung von allen Probanden (n=42), den Männer (n=30), Frauen (n=12), 3D-erfahrenen (n=23) und 3D-unerfahren (n=19) Probanden.)

**Tabelle 8.2:** Durchschnittlich benötigte Zeit für jede Rotationstechnik aller Probanden ( $n = 42$ ), sowie der Männer ( $n = 30$ ), Frauen ( $n = 12$ ), 3D-erfahrenen ( $n = 23$ ) und 3D-unerfahrenen ( $n = 19$ ) Probanden.

Rotationstechnik	Durchschnittlich benötigte Zeit (in Sek.)				
	alle	Männer	Frauen	3D-Erfahr.	3D-Unerfahr.
Bell's VT	148,06	141,42	164,67	142,39	154,92
Shoemake's VT	133,24	130,37	140,42	130,75	136,26
Two-Axis VT	122,92	125,79	115,75	127,48	117,39*
Fixed Two-Axis VT	147,71	148,59	145,50	146,54	149,12

\*Geringe Unterschiede der durchschnittlich benötigten Zeiten sind statistisch nicht signifikant (siehe Abbildung 8.7 für einen besseren Überblick).

**Tabelle 8.3:** Signifikanzwerte beim Vergleich der Rotationstechniken bezüglich der benötigten Zeit und Schüsse.

Verglichene Rotationstechniken	Paired Samples T-Test benötigte Zeit		Wilcoxon Test benötigte Schüsse	
Bell's VT vs. Shoemake's VT	t = -4,240	p = ,000	Z = -2,083	p = ,037
Bell's VT vs. Two-Axis VT	t = 5,360	p = ,000	Z = -2,181	p = ,029
Bell's VT vs. Fixed Two-Axis VT	t = 0,080	p = ,937	Z = -0,796	p = ,426
Shoemake's VT vs. Two-Axis VT	t = 2,231	p = ,031	Z = -0,585	p = ,558
Shoemake's VT vs. Fixed Two-Axis VT	t = -3,053	p = ,004	Z = -0,766	p = ,444
Two-Axis VT vs. Fixed Two-Axis VT	t = 7,153	p = ,000	Z = -1,187	p = ,235

0,01) im Vergleich zu allen anderen Rotationstechniken. Bell's VT, Shoemake's VT und der Fixed Two-Axis VT wurden in Bezug auf Frage (1) sehr ähnlich bewertet. Hinsichtlich Frage (2) wurde der Fixed Two-Axis VT signifikant schlechter ( $p \leq 0,001$ ) als Bell's VT und Shoemake's VT bewertet.

Beim separaten Vergleich der Ergebnisse von Männern und Frauen konnten kaum geschlechtsspezifische Unterschiede beobachtet werden. Jedoch zeigte sich, dass bei der Verwendung von Bell's VT die Frauen signifikant ( $p < 0,05$ ) langsamer als die Männer waren. Die Auswertung der Vorerfahrungen aller Probanden attestiert jedoch allen weiblichen Probanden geringere Vorerfahrungen mit 3D-Navigation, 3D-Rotation, 3D-Modellierprogrammen und 3D-Spielen. Um zu überprüfen, ob dies den gefundenen signifikanten Unterschied zwischen Männern und Frauen mit Bell's VT erklärt, wurden die 3D-erfahrenen und 3D-unerfahrenen Probanden separat verglichen. Hierbei konnte kein signifikanter Unterschied festgestellt werden. Folglich weist der signifikante Unterschied der mit Bells VT benötigten Zeit zwischen Männern und Frauen darauf hin, dass das Geschlecht einen signifikanten Einfluss auf die erreichbare Geschwindigkeit von Bell's VT hat. Die 3D-Vorerfahrung hatte hingegen bei allen Rotationstechniken keinen statistisch signifikanten Einfluss auf die erreichten Ergebnisse.

Zusammenfassend konnten mit dem Two-Axis VT und Shoemake's VT die besten Leistungen erbracht werden. Bei beiden Techniken war jedoch eine höhere Anzahl von Schüssen registriert worden, wobei dennoch die Probanden die Aufgabe mit geforderter Genauigkeit

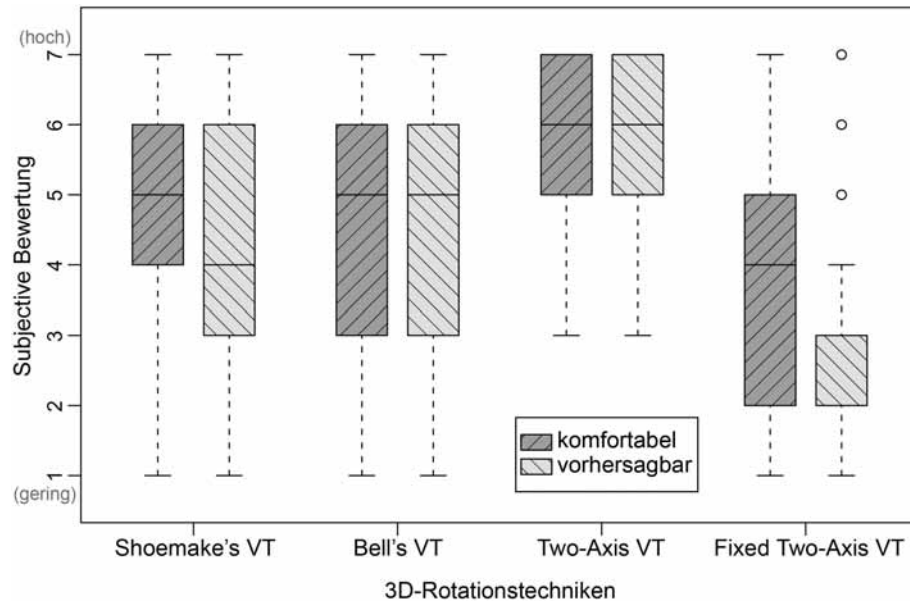


Abbildung 8.9: Boxplots der subjektiven Bewertung der Rotationstechniken.

(Ziele mussten getroffen werden) in geringerer Zeit erfüllten. Darüber hinaus wurde der Two-Axis VT von den Probanden als angenehmste und am besten steuerbare Rotationstechnik eingestuft. Diese Ergebnisse bestätigen damit auch die Anforderungsliste aus Abschnitt 8.1.2.

## 8.4 Kritische Anmerkungen und Ausblick

In Abschnitt 8.1.2 wurden vier grundlegende Anforderungen für komfortabel und intuitiv steuerbare Rotationstechniken identifiziert. Die Relevanz dieser Anforderungen konnte zusätzlich über die durchgeführte Evaluation bestätigt werden.

Den ermittelten Ergebnissen ist noch einmal hinzuzufügen, dass die unterschiedlichen Rotationstechniken mit der *world-in-hand* Metapher zum Explorieren von 3D-Szenen mit Blick von außen auf die Szene untersucht wurden. In Spezialfällen, wie der virtuellen Endoskopie, sind jedoch Ansichten von Innenräumen zu explorieren, für die eine *camera-in-hand* Metapher geeigneter ist [Partala 1999]. Während bei der *world-in-hand* Metapher die Kamera um die Szene rotiert wird, wird die Kamera bei der *camera-in-hand* Metapher um den eigenen Mittelpunkt rotiert. Entsprechend sind die zuvor präsentierten Ergebnisse hier im allgemeinen nicht übertragbar. Jedoch werden diese nicht grundlegend abweichen, da auch bei Verwendung der *camera-in-hand* Metapher die in Abschnitt 8.1.2 vorgestellten Anforderungen für komfortabel und intuitiv steuerbare Rotationstechniken gelten.

Eine bisher noch offen gelassene Frage ist jedoch, ob es die *ideale* Rotationstechnik gibt, die alle vier Anforderungen erfüllt. Eine besondere Herausforderung ist dabei die gleichzeitige Erfüllung von Anforderung 1 (*Gleiche Aktion - gleiche Reaktion*) und 3 (*transitive und kommutative Rotationsinteraktion*). Dieses Ziel kann mit 2D-Eingabegeräten jedoch nicht voll erfüllt werden. Bei 2D-Interaktionen muss, um transitive und kommutative 3D-Bewegungen zu garantieren (3), von einer überall gleich gerichteten Bewegung (1) abstrahiert werden. Das gleiche gilt auch andersherum, so dass bei garantierter *Gleiche Aktion - gleiche Reaktion* keine Transitivität und Kommutativität resultierender 3D-Rotationsbewegungen erreicht werden können.

Zu erfüllen sind alle vier Anforderungen nur durch 3D-Eingabegeräte. Ein Problem aktueller und auch zukünftiger 3D-Eingabegeräte ist jedoch, dass der einfache Zeigecharakter oder der flexible Ruhepunkt einer Maus verloren gehen. Ersteres trifft auf 3D-Eingabegeräte wie die SpaceMouse und den neueren SpaceNavigator<sup>2</sup> zu. Diese erlauben nur die Eingabe einer Richtung und eines Betrages (*rate-based input*). Letzteres trifft auf mittels 3D-Tracking verfolgte *Handles* als 3D-Eingabegeräte zu, welche in einer gewählten Position gehalten werden müssen und nicht einfach losgelassen bzw. abgelegt werden können, ohne die Ansicht zu verändern [Zhai 1998; Ware und Rose 1999]. Diese Nachteile sind bei konkreten Anwendungen gegenüber den Nachteilen der mausbasierten Interaktion abzuwiegen.

Weiterhin ist kritisch anzumerken, dass die in diesem Kapitel durchgeführte Betrachtung von mausbasierten Rotationstechniken die komplexe Navigation im virtuellen 3D-Raum vereinfacht untersucht. Das Verschieben und Zoomen in der Szene sind zusätzliche Freiheitsgrade, die bei komplexen 3D-Szenen gesteuert werden müssen. Untersuchungen zu beidhändiger Navigationen konnten hierbei einen eindeutigen Vorteil dieser gegenüber der einhändigen Navigation nachweisen. Prinzipiell gelten auch bei der simultanen beidhändigen Interaktion die in Abschnitt 8.1.2 identifizierten Anforderungen für komfortabel und intuitiv steuerbare Rotationstechniken, wodurch auch hier eine Übertragung der Evaluationsergebnisse plausibel ist. Wechselwirkungen, die bei beidhändiger Interaktion zu leicht anderen Ergebnissen führen können, sind jedoch nicht auszuschließen.

Zur 3D-Navigation mit Rotation, Zoomen und Verschieben kommt in komplexen Anwendungen zusätzlich auch die Navigation im Parameter- bzw. Informationsraum der Visualisierungen hinzu. Der Informationsraum wird dabei unter anderem durch die für eine Visualisierung zur Verfügung stehenden Oberflächenmodelle, die bereitgestellten Möglichkeiten zur Änderung von Farben, Transparenzen, Renderingstilen u.a. aufgespannt. Dieses Level an zusätzlichem Detail in der Interaktion stellt einen zusätzlich nötigen Aspekt für eine umfassende abschließende Evaluation von komplexen interaktiven Systemen dar, welche nicht Fokus dieser Arbeit ist.

Eine wie in Abschnitt 8.1.1 geforderte Unterstützung zielgerichteter dreidimensionaler Navigations- und Explorationsaufgaben kann dabei mit einer geeigneten Rotationstechnik allein nicht erreicht werden. In bestimmten Fällen kann dies aber durch Vorwissen über ähnliche und immer wiederkehrende Navigationsaufgaben, wie das Heranfahren/Zoomen an selektierte Objekte erreicht werden. Dabei kann das Vorwissen über die Aufgabe zur interaktionsfreien Führung des Anwenders verwendet werden. Die Führung kann hierbei über eine dynamische Visualisierung (Animation) vom aktuellen Zustand der Ansicht zum Zielzustand realisiert werden. Entsprechend wird bei der Betrachtung von dynamischen Visualisierungen im nächsten Kapitel auf die Unterstützung der Interaktion durch geführte Animationen eingegangen.

---

<sup>2</sup><http://www.3dconnexion.de/>



# Dynamische Visualisierung und Interaktion

---

Die interaktive Navigation in 3D-Visualisierungen kann allein keine ausreichende Unterstützung darstellen. Dies gilt zum Beispiel, wenn klar definierte Aufgaben durch einen zu hohen Freiheitsgrad der Interaktion (z.B. beim Ansteuern schwer erreichbarer bzw. einsehbarer Teile) verlangsamt werden. Im Besonderen gilt dies auch, wenn Interaktionsstrategien zur effizienten Exploration der 3D-Szene bei (ungeübten) Anwendern fehlen. Zusätzlich ist der erkundbare Visualisierungs- und Informationsraum (siehe auch [Ritter 2005]), aufgespannt durch eine Vielzahl an Visualisierungsstilen und -parametern (z.B. Transparenz, Farbe, Hatching, Stippling, ...), so immens groß, dass eine ungeleitete interaktive Exploration des Informationsraumes nicht praktikabel ist.

Hier kann geführt durch dynamische Visualisierungen bzw. Animationen der Informationsraum oder Teile dessen von aktuellem Interesse präsentiert werden. Gleichzeitig können dadurch Strategien zur Exploration des Visualisierungsraumes an Anfänger vermittelt werden.

In der (computerunterstützten) Ausbildung sind dynamische Visualisierungen (Lehrvideos) weit verbreitet. Bei der Interventionsplanung finden dynamische Visualisierungen unter anderem bei interdisziplinären Besprechungen wie Tumorboards Anwendung, um z.B. kurz und prägnant einen Überblick über die patientenspezifische Anatomie und die Lage von Tumoren zu vermitteln.

Ein wiederkehrendes Problem in der Interventionsplanung ist dabei die Erzeugung gleicher dynamischer Visualisierungen bzw. Animationen für unterschiedliche Patientendaten. Das manuelle Erstellen entsprechender Animationen ist für jeden Patientendatensatz zu aufwändig. Darüber hinaus sind manuell erstellte Animationen nicht vergleichbar und ihr Entstehungsprozess schlecht nachvollziehbar und dokumentierbar [Iserhardt-Bauer et al. 2001, 2002].

Für Lehr- und Ausbildungszwecke werden meist dynamische Animationen nur für einen Standard-Datensatz erstellt (vergleiche Kap. 2). In fallbasierten Trainingssystemen ist jedoch auch die dynamische Visualisierung einer Vielzahl fallspezifischer Daten sinnvoll. Hierauf wird jedoch aufgrund des manuellen Aufwandes meist verzichtet.

In diesem Kapitel werden deshalb zunächst die Visualisierungsaufgaben dynamischer Darstellungen für die chirurgische Ausbildung und Interventionsplanung erarbeitet (Abschnitt 9.1). Daraufhin wird die spezielle Problematik der dynamischen Visualisierung patientenspezifischer Daten dargelegt (Abschnitt 9.2) und deren Anwendung in aktuellen chirurgischen Ausbildungs- und Interventionsplanungssystemen zusammengefasst.

## 9.1 Aufgaben und Möglichkeiten dynamischer Visualisierungen

Klassisch werden dynamische Visualisierungen eingesetzt zur:

- Visualisierung dynamischer (zeitveränderlicher) Daten,
- Präsentation und Erläuterung komplexer räumlicher und funktionaler Zusammenhänge sowie auch zur
- Effekt- und Leistungsschau (*Eye-Catcher*).

Für die Unterstützung in der chirurgischen Ausbildung und Interventionsplanung ist hierbei vor allem die Präsentation und Erläuterung komplexer räumlicher Zusammenhänge von Interesse. Animationen können dabei gegenüber statischen Bildern die Position, Ausdehnung und den Verlauf wichtiger Strukturen besser veranschaulichen [Preim 1998].

Hinzu kommt die Unterstützung von Navigationsaufgaben, z.B. beim Abfliegen von Pfaden oder dem Heranfahren an schlecht sichtbare bzw. schwer erreichbare Strukturen (z.B. in der virtuellen Endoskopie).

Aufgaben, die dynamische Visualisierungen in diesem Zusammenhang erfüllen können, sind:

- Wiedergabe und Illustration didaktisch aufbereiteten Wissens (z.B. in Lehrvideos),
- Standardisierung von Visualisierungsprozessen (z.B. zur reproduzierbaren und vergleichbaren visuellen Analyse, medizinischer Volumendatensätze [Iserhardt-Bauer et al. 2001; Rößler et al. 2007]) und
- Effiziente Vermittlung eines 3D-Überblicks über die patientenspezifische Anatomie und Pathologie (z.B. bei interdisziplinären Tumorboard-Besprechungen).

Zusätzliche Aufgaben dynamischer Visualisierungen, die im Kontext komplexer interaktiver 3D-Anwendungen, vor allem für nicht professionelle Visualisierungsanwender (z.B.: Ärzte, Chirurgen) erfüllt werden sollten, sind [Bade et al. 2004a]:

- Heranführen an eine Visualisierung,
- nachvollziehbares Animieren/Illustrieren von Übergängen zwischen unterschiedlichen Visualisierungen sowie die
- Übernahme von kleineren, wiederkehrenden Interaktionsaufgaben zur Interaktionsunterstützung (z.B. Heranfahren und Hervorheben selektierter Objekte).

Das Heranführen an eine Visualisierung sowie die Animation von Übergängen zwischen einer aktuellen und einer neuen Darstellung der Szene ist zum Beispiel dann sinnvoll, wenn sich die dargestellte Szene zur nächsten Ansicht hin stark verändert bzw. sich eine vielleicht ungewohnte Ansicht der Szene ergeben kann. Ungewohnt kann hierbei die Blickrichtung und Skalierung aber auch der Darstellungsstil (z.B. Stipling, Hatching, mit Glyphen angereichert, ...) sein. Auch fehlende Kontextinformationen bei Nahaufnahmen können zu ungewohnten und schwer interpretierbaren Ansichten führen. In diesen Fällen kann das Begreifen und Interpretieren der ungewohnten Ansicht unterstützt werden, indem über eine Animation die Entstehung der Ansicht aus einer bekannten Ansicht verfolgt werden kann [Bade et al.

2004a]. Die Übernahme von kleineren Interaktionsaufgaben soll besonders aufwändige bzw. unnötig lästige Interaktionen vereinfachen.

Die identifizierten Aufgaben lassen sich hierbei in einem „*Story Telling Informed*“-Ansatz für Visualisierungen verallgemeinern [Hauser und Wohlfart 2008]. Die Aufgabe von statischen, interaktiven und dynamischen Visualisierungen ist in diesem Zusammenhang allgemein die effiziente Vermittlung von Erkenntnissen und Informationen durch die Kommunikation einer Visualisierungsstory.

Die Story einer Visualisierung soll dabei nach Hauser und Wohlfart [2008] zur besseren Verständlichkeit ihren eigenen animierten Entstehungsprozess (*Heranführen an eine Visualisierung*), verschiedene wichtige Zustände/Ansichten in der Visualisierungsstory (Story-Knoten) sowie animierte Übergänge zwischen den Story-Knoten (Animation von Übergängen zwischen unterschiedlichen Visualisierungen) enthalten. Wie entsprechende Storys und dynamische Visualisierungen für patientenspezifische Daten erreicht werden können, wird in den nächsten Abschnitten betrachtet.

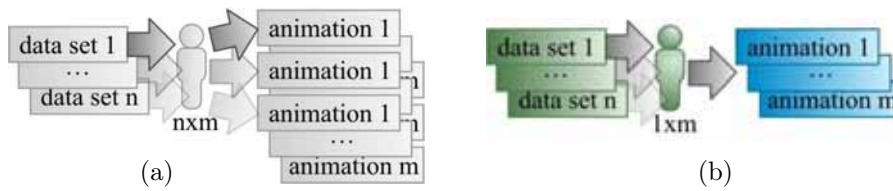
## 9.2 Dynamische Visualisierung patientenspezifischer Daten

Typische Szenarios dynamischer Visualisierungen von patientenspezifischen Volumendaten können dem *Information Seeking Mantra* (ISM) [Shneiderman 1996] folgen. Entsprechende dynamische Visualisierungen geben erst einen Überblick (*overview first*), um Betrachter mit den dargestellten Daten vertraut zu machen (z.B. durch eine 360 Grad Rotation um die Szene). Darauf folgt dann das Heranfahren an und Filtern von Daten im Fokus des Interesses (*zoom and filter*, z.B. durch Ausblenden irrelevanter Strukturen). Der letzte Schritt im *Information Seeking Mantra: details on demand* wird dann durch Nutzerinteraktion in der dargestellten Szene ermöglicht.

Als weitere allgemeine Szenariotypen dynamischer Visualisierungen beschreiben [Hauser und Wohlfart 2008] vergleichende Visualisierungen und sich iterativ wiederholende Visualisierungen. Bei vergleichenden Visualisierungen wird erst ein Datensatz dem *Information Seeking Mantra* folgend vorgestellt und dann ein Vergleichsdatsatz (z.B. prä-, postoperativ oder linke und rechte Schädelhälfte) eingeblendet sowie deren Unterschiede und Gemeinsamkeiten dargestellt. Sich iterativ wiederholende Visualisierungen eignen sich zur Präsentation einer größeren Anzahl ähnlicher Elemente (z.B. Lymphknoten bei der Planung von Neck Dissections). Hierbei werden die Elemente nacheinander über die grundlegend gleichen Visualisierungsschritte präsentiert (z.B. durch Heranfliegen, Hervorheben und 360° Rotation um diese) [Hauser und Wohlfart 2008].

Die besonderen Herausforderungen bei der dynamischen Visualisierung patientenspezifischer Daten sind die Fallzahlen sowie die Forderung nach reproduzierbaren Möglichkeiten zur visuellen Inspektion medizinischer Volumendaten.

Ein Ansatz zur einfachen Erzeugung von dynamischen Visualisierungen ist dabei das Aufzeichnen der Nutzerinteraktion bei der visuellen Analyse und beim Präsentieren der Analyseergebnisse. Diese können dann wiederholt abgespielt werden, wodurch die Dokumentation und Ergebnisweitergabe einer visuellen Analyse (z.B. von einem Radiologen an einen Chirurgen) effizient unterstützt werden kann [Hauser und Wohlfart 2008]. Die visuelle



**Abbildung 9.1:** Definition von  $m$  dynamischen Visualisierungen (Animationen) für  $n$  Datensätze durch einen Autor: (a) für jeden der  $n$  Datensätze müssen jeweils  $m$  Animationen definiert werden ( $\Rightarrow n \times m$  Definitionen), (b) über abstrakte, adaptive Definition müssten nur  $m$  Animationen definiert werden.

Analyse selbst kann hierdurch jedoch nicht unterstützt und standardisiert werden. Auch die simultane Änderung mehrerer Visualisierungsparameter ist dabei nicht möglich.

Ein anderer Ansatz ist die Verwendung von fest vorgegebenen Animationsabläufen, wie sie meist mit kommerziellen Animationssystemen erreicht werden können. Diese sind jedoch für jeden darzustellenden Datensatz jeweils neu zu definieren, da bereits bei ähnlichen Datensätzen und Problemstellungen in medizinischen Anwendungen eine hohe Variabilität von Lage, Ausdehnung und Anzahl von Objekten im Fokus des Interesses (Tumore, Lymphknoten, ...) möglich ist. Für jeden Fall manuell Animationen zu erzeugen, ist jedoch im Kontext von Interventionsplanung und fallbasierter chirurgischer Ausbildung nicht praktikabel.

Bei einer gegebenen Menge von  $n$  Datensätzen und einer gegebenen Menge an  $m$  zu erstellenden Animationen pro Datensatz (z.B.: Rotation um ganze Szene, Rotation um alle pathologischen Veränderungen) ist folglich eine Menge von  $n \times m$  Animationen durch einen Autor zu erstellen (Abb. 9.1 (a)).

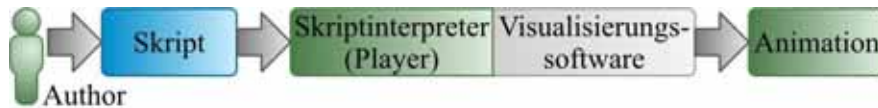
In diesem Zusammenhang sind Techniken wünschenswert, welche die Spezifikation von Animationen nicht in absoluten Angaben von Objektnamen, Entfernungen und Koordinaten, sondern durch höhere Visualisierungs- und Kommunikationsziele (ähnlich „erkläre alle Objekt der Kategorie X“) ermöglichen. Eine solche Form der abstrakten Spezifikation ist unabhängig von der konkreten Geometrie und Topologie und könnte adaptiv auf voneinander abweichende Datensätze (z.B. von unterschiedlichen Patienten) angewendet werden. Zusätzlich wird dadurch die Vermischung und nahtlose Verzahnung von interaktiver und geführter dynamischer Visualisierung möglich.

Bei einer gegebenen Menge von  $n$  Datensätzen und einer gegebenen Menge an  $m$  zu erstellenden Animationen pro Datensatz, bräuchten dank abstrakter, adaptiver Definition von Animationen nur noch einmalig  $m$  Animationen durch einen Autor erstellt zu werden (Abb. 9.1 (b)). Hierbei kann es zusätzlich zwischen den Animationen 1... $m$  gleiche Teile geben, bei denen bereits definierte Animationsteile wiederverwendet bzw. leicht über die Änderung einiger Parameter angepasst werden können.

**Anforderungen an die adaptive Animationsdefinition:** Ansätze, die eine Definition von einer dynamischen Visualisierung für mehrere Datensätze ermöglichen wollen, müssen mit:

- adaptiven/abstrakten Objekten und Objektbeschreibungen,
- adaptiven/abstrakten Positions-, Längen- und Richtungsangaben sowie mit
- adaptiven/abstrakten Befehlen umgehen können.

**Möglichkeiten der Animationsdefinition:** *Keyframing*<sup>1</sup> und das *Abfilmen* von bzw. *Aufzeichnen der Nutzerinteraktion* in interaktiven 3D-Szenen sind Möglichkeiten zur Animationsdefinition, die bereits eine genaue Geometrie im Definitionsprozess benötigen. Dadurch ist die Adaption an andere Objekte und räumliche Verhältnisse nicht möglich. Potentiell in der Lage zur Adaption sind nur skriptbasierte Ansätze, bei denen Animationsskripte erst zur Laufzeit mit den aktuell gegebenen Daten in Animationen umgesetzt werden (siehe Abb. 9.2).



**Abbildung 9.2:** Verarbeitung von Animationsskripten durch einen Skriptinterpreter für eine Visualisierungssoftware. (modifiziert aus [Muehler 2005])

**Anforderungen an Animationsskripte:** Ideale Animationsskripte sollten:

- ähnlich natürlichen Texten lesbar und verständlich sein,
- das Editieren auf unterschiedlichen Niveaus/Abstraktionsleveln ermöglichen sowie
- durch einen Autor erweiterbar und für spezielle Aufgaben spezialisierbar sein.

Im Folgenden werden überblickend Ansätze zur abstrakten Beschreibung und Definition dynamischer Visualisierungen besprochen.

## 9.3 Verwandte Arbeiten

Ansätze zur abstrakten Beschreibung von Kommunikationszielen zur automatischen Steuerung und Erzeugung informativer Grafiken kamen vor über einem Jahrzehnt mit einer Vielzahl von neuen Visualisierungsmöglichkeiten auf.

Frühe Konzepte auf dem Gebiet der automatischen Erzeugung von Animationen setzen auf die Deklaration von high-level Kommunikationszielen, welche dann von den Systemen in kleinere Animationsaktionen zerlegt werden. Bekannte Ansätze sind das WIP-Konzept (*Wissensbasierte-Informationen-Präsentation*) [Elisabeth André 1996; Butz 1994] und ESPLANADE (*Expert System for Planning Animation Design and Editing*) [Karp und Feiner 1993] welche beide eine Wissensbasis nutzen, mit der gegebene abstrakte Kommunikationsziele hierarchisch zu elementaren Animationsanweisungen zerlegt werden. Die Definition von Kommunikationszielen ist jedoch sehr komplex und starr. Erzeugte Animationen können auch nur durch Änderung der abstrakten Kommunikationsziele, aber nicht auf unteren Ebenen der Zerlegungshierarchie vorgenommen werden.

Während diese Ansätze die Erzeugung fertiger Animationsfilme verfolgen, konzentriert sich der Ansatz von Preim et al. [1996a] auf die Erzeugung kleinerer Animationsteile, die zum Beispiel ein vom Anwender selektiertes Objekt in der Szene mit der Kamera anfahren und dieses hervorheben. Hierbei werden abstrakte Befehle „*Betone Objekt X*“ oder „*Erkläre Objekt Y*“ über Skripte näher deklariert. Neben der textuellen Eingabe können diese auch aus der Interaktion erstellt werden. Die Skriptdeklaration sowie die Aufzeichnung von Skriptbefehlen aus der Interaktion ermöglichen eine gute Kontrolle der Animationen, jedoch sind

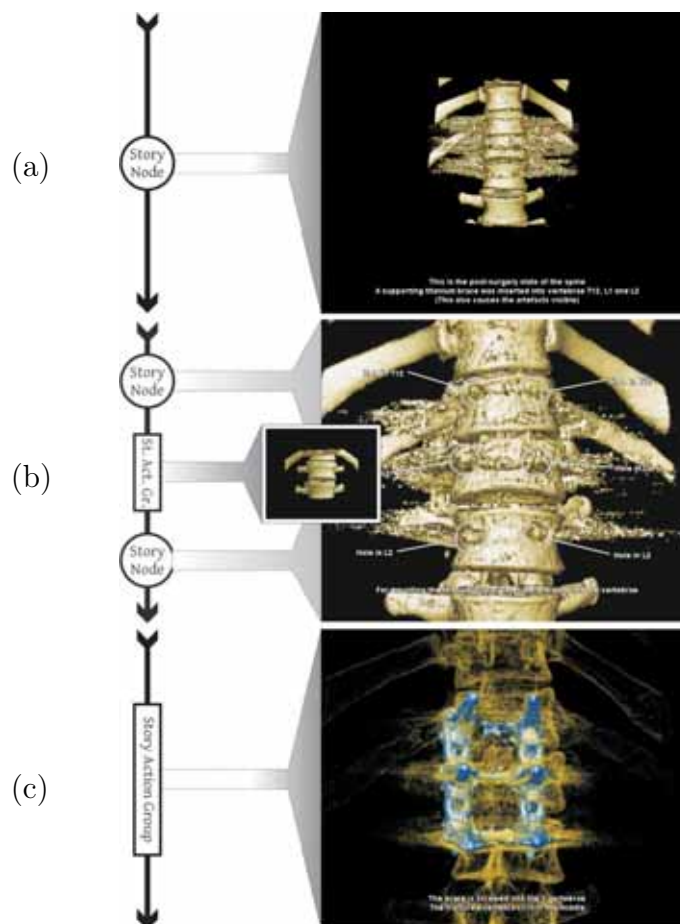
<sup>1</sup>Keyframing wird hier synonym für das Setzen von festen Schlüsselbildern und Übergängen zwischen diesen verwendet.

diese sehr an die aktuelle Szene gebunden. Anwendung findet das Skriptkonzept im ZOOMILLUSTRATOR [Preim und Strothotte 1996]. In diesem kommt nur eine begrenzte Anzahl an vordefinierten Modellen zum Einsatz, was den Einsatz dieses Skriptansatzes ermöglicht. Problematisch erweist sich auch die Lesbarkeit der erzeugten Skripte, da diese eine Reihe an fixen Koordinaten und Parametern enthalten.

### 9.3.1 Dynamische *Story Telling Informed* Visualisierungen

Ein generelles Konzept zur Definition von dynamischen Visualisierungen motivieren Hauser und Wohlfart [2008] über einen *Storytelling*-Ansatz. Hierbei wird die geplante Aussage einer dynamischen Visualisierung als Story aufgefasst. In der Tradition des Storytelling werden dabei *Visualisierungstories* über die generelle und sequentielle Beschreibung der grundlegenden Ereignisse in einem *Story Plot* beschrieben. Abbildung 9.3 illustriert eine solche *Visualisierungstory*.

Wichtige Zustände (Meilensteine) der Visualisierung werden jeweils als *story node* aufgefasst. Erreicht die dynamische Visualisierung einen *story node*, so wird hier die dynamische Animation angehalten, bis der Anwender diese fortsetzt. Zwischen aufeinander folgenden



**Abbildung 9.3:** Ausschnitt einer Visualisierungstory. (a) Postoperative Ansicht einer Wirbelfrakture. (b) Heranfahren an die drei beteiligten Wirbel gefolgt vom kurzzeitigen Ein- und wieder Ausblenden der präoperativen Ansicht zur Veranschaulichung von Details über die Schraublöcher der implantierten Titaniumbrücke. (c) Reduktion der Opazität der Knochendarstellung für eine bessere Sicht und Lagebeurteilung der Titaniumbrücke. [Hauser und Wohlfart 2008]

*story nodes* definieren sogenannte *story transitions* den Übergang von einer Visualisierungsansicht zur nächsten. Hierbei werden weiche und fließende Übergänge bevorzugt, da Diskontinuitäten und abrupte Sprünge die Interpretation und mentale Karte von Betrachtern einer Szene beeinträchtigen können [Hauser und Wohlfart 2008].

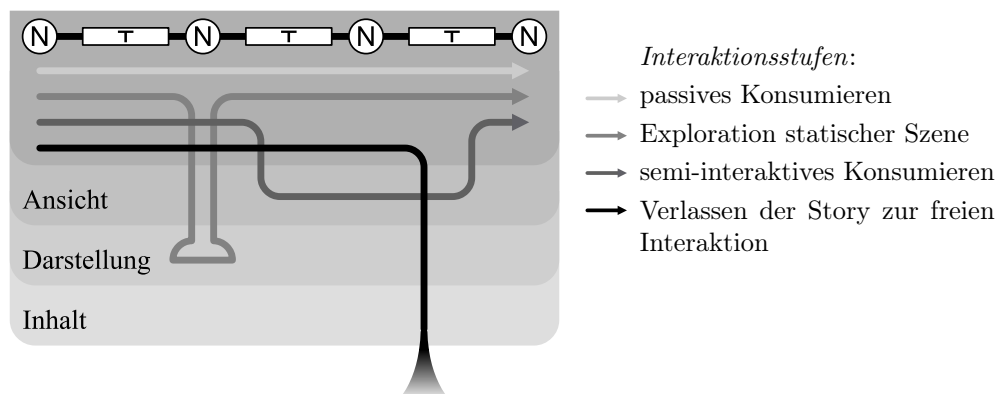
Eine *story transition* kann dabei verschiedene kleinere Übergänge enthalten, welche *story action groups* genannt werden. Diese werden nacheinander während einer *story transition* abgearbeitet. Jede *story action group* enthält dabei eine variable Anzahl an Änderungen von Visualisierungsparametern, welche parallel ausgeführt werden.

Über dieses allgemeine Konzept lassen sich komplexe dynamische Visualisierungen erzeugen, wobei kleine Animationsteile hierarchisch in *story transitions* und *story action groups* gekapselt werden können.

Neben der reinen dynamischen Visualisierung berücksichtigen Hauser und Wohlfart [2008] in ihrem Konzept auch die Kombination von dynamischer und interaktiver Visualisierung. Dabei unterscheiden sie vom *passiven Konsumieren* der dynamischen Visualisierung bis zur komplett *freien Interaktion* zwei weitere *Interaktionsstufen* (siehe Abb. 9.4).

Bei der *Exploration statischer Szenen* zwischen verschiedenen Animationsteilen kann eine aktuelle Sicht auf die Daten interaktiv verändert werden. Grundsätzlich können hierbei auf drei unterschiedlichen *Parametrisierungsstufen* Parameter der Ansicht (z.B. die Sichtrichtung), Parameter der Darstellung (Farbe, Transparenz, Renderingstil, ...) sowie der Inhalt einer Szene (Teile des Datensatzes die gezeigt bzw. nicht mit angezeigt werden) interaktiv manipuliert werden (siehe graue Facetten in Abb. 9.4). Nach der Interaktion kann dann wieder zur definierten dynamischen Visualisierung zurückgekehrt und diese in ihrem definierten Zustand weiter verfolgt werden.

Beim *semi-interaktiven Konsumieren* dynamischer Visualisierungen ist die Steuerung der Szene zwischen der dynamischen Visualisierung und dem Anwender aufzuteilen. Das Konzept von Hauser und Wohlfart [2008] sieht dabei vor, dass Anwender die Kontrolle der Szene bis zu einem gewissen Grad (beginnend bei der Manipulation von Parametern der Ansicht) selbst übernehmen können. Dies kann zum Beispiel bei einem zweiten Durchlauf einer dynamischen Visualisierung genutzt werden, um diese auch aus einer anderen Sichtrichtung verfolgen zu können.



**Abbildung 9.4:** Dynamische Visualisierung und Interaktion auf unterschiedlichen *Interaktionsstufen* mit Manipulation auf unterschiedlichen *Parametrisierungsstufen*: Ansicht, Darstellung, Inhalt. (modifiziert aus [Hauser und Wohlfart 2008])



Das Gesamtkonzept der dynamischen *Story Telling Informed* Visualisierung erlaubt damit die Definition von Visualisierungsstories unter Berücksichtigung und Einbindung von Nutzerinteraktion. Um in medizinischen Anwendungen nicht für jeden Datensatz eine neue Visualisierungsstory erstellen zu müssen, sind Umsetzungsansätze nötig, welche eine möglichst automatische Adaption definierter Visualisierungsstories an unterschiedliche Datensätze ermöglichen. Hierfür werden im Weiteren Ansätze adaptiver Animationsskripte weiter untersucht.

### 9.3.2 Ansätze adaptiver dynamischer Visualisierungen

Ein System, welches Animationen vereinfacht und sich mit relativen Positionsangaben definieren lässt, ist ALICE [Conway 1997; Cooper et al. 2003]. Alice stellt dabei ein Animationssystem für Computerneulinge und Kinder dar, wobei es auch zur Einführung in die OO-Programmierung von Programmierneulingen verwendet wird [Cooper et al. 2003]. Vereinfacht werden hier alle Animationen mit konstanter Geschwindigkeit ausgeführt. Zusätzlich werden Zeitangaben weitestgehend ausgespart, so dass nur die Reihenfolgen (`doinorder(...)`) oder Gleichzeitigkeit (`dotogether(...)`) von Animationsteilen bestimmt werden müssen. Positionen lassen sich im System relativ z.B. über `left`, `right`, ... und Bewegungen damit z.B. über `bunny.move(down, 1)` definieren (siehe Listing 9.1).

Grundlegend stellt Alice damit einen Ansatz für adaptive dynamische Visualisierung dar. Jedoch sind Objektbezeichner und Entfernungs- bzw. Positionsangaben statisch und eine Adaption an unterschiedliche Datensätze ist nicht vorgesehen.

**Listing 9.1:** Animationsskript in ALICE. Auffällig sind die relativen Bewegungsangaben sowie die stark an eine objektorientierte Programmiersprache (ähnlich Java) angelehnte Syntax. (modifiziert aus [Conway 1997] neu aufgelegt in [Muehler 2005])

---

```
doinorder (
    % recoil
    bunny.resize(toptobottom , 0.5, likerubber), -vs

    % leap
    dotogether(
        bunny.resize(toptobottom , 2, likerubber),
        bunny.move(up, 1)
    ),

    % fall
    bunny.move(down , 1),

    % bounce
    bunny.resize(toptobottom , 0.5, likerubber),
    bunny.resize(toptobottom , 2, likerubber)
)
```

---



### 9.3.3 Dynamische Visualisierungen in Anwendungen

Neben den bisher beschriebenen Ansätzen aus der wissenschaftlichen Forschung finden fortschrittliche Techniken zur adaptiven dynamischen Visualisierungen (noch) keine Anwendung in der chirurgischen Ausbildung und Interventionsplanung.

Bei computerunterstützten Lernsystemen in der Medizin kommen dynamische Visualisierungen meist nur als starre vorgerenderte Videos zum Einsatz (siehe Abschnitt 2). Zum Teil werden auch interaktive *object movies* (z.B. QuickTime VR) angeboten [Golland et al. 1999; Tiede et al. 2002; Pommert et al. 2006]. Hierbei werden von festen Kamerapositionen (z.B. auf einer die Szene umgebenden Kugel) Bilder der Szene vorgerendert, welche dann in speziellen Viewern interaktiv angesteuert werden können. Interaktive und adaptive dynamische Visualisierungen variabler Daten finden in kommerziellen Systemen keine Anwendung.

In der Interventionsplanung finden adaptive dynamische Visualisierungen häufiger Anwendung. Die Freiheitsgrade für generierte Animationen beschränken sich dabei jedoch meist nur auf das Rotieren der Kamera um ein Objekt [Iserhardt-Bauer et al. 2002; Preim et al. 2003; Rößler et al. 2007]. Oder es werden erweiterte Skriptfunktionalitäten angeboten, die aber nicht adaptiv auf unterschiedliche Datensätze übertragbar sind (siehe z.B. AMIRA [Mercury Computer Systems 2007]). Interaktiv kann dabei in die Animationen in den Systemen nicht eingegriffen werden. Als Ausnahmen erzeugen Tiede et al. [2002] und Rößler et al. [2007] interaktive, aber auch nur fest vorgerenderte *object movies*.

## 9.4 Skriptkonzept für adaptive dynamische Visualisierungen

Um die Möglichkeiten, die dynamische Visualisierungen eröffnen (siehe Abschnitt 9.1), auch in der chirurgischen Ausbildung und Interventionsplanung auf variablen Daten nutzen zu können, wurde eine adaptionsfähige Skriptsprache entwickelt [Muehler 2005; Muehler et al. 2006], deren Konzept im Folgendem vorgestellt wird.

Die Betrachtungen beschränken sich hierbei hauptsächlich auf die Aspekte des Konzeptes, die eine Adaption an unterschiedliche Datensätze erlauben und eine standardisierte dynamische Visualisierung von patientenspezifischen Daten fördern.

### 9.4.1 Hierarchisches Konzept

Der Strategie, abstrakte Kommunikationsziele bzw. Animationsanweisungen in elementare Befehle zu zerlegen folgend, wird ein Ansatz gewählt, der es ermöglicht, Skriptbefehle unterschiedlicher Abstraktionsstufen zu verwenden.

Auf dem untersten Abstraktionslevel (*LowLevel*-Skript) gibt es detaillierte atomare Befehle, welche die ganze Bandbreite an Funktionen der Animationsschnittstelle (Skriptinterpreter) abdecken. Auf diesen atomaren Befehlen setzen abstrakte Befehle für komplexere Aktionen auf (siehe Abb. 9.5). Die Abstraktion kann dabei auf alle Bestandteile eines Befehls (siehe Abschnitt 9.4.2) angewendet werden. So können mehrere Befehle zu einem abstrakten Befehl zusammengefasst, Objekte zu Objektgruppen aggregiert und Befehlsparameter abstrakter definiert werden. Skripte mit abstrakten Befehlen, Objekten oder Parametern werden als *HighLevel*-Skripte bezeichnet.

Da sich *HighLevel*-Befehle aus *LowLevel*-Befehlen zusammensetzen, sind *HighLevel*-Skripte über Ersetzungsregeln komplett in *LowLevel*-Skripte überführbar. Ersetzungsregeln

beinhalten dabei, welche *LowLevel*-Befehle einen *HighLevel*-Befehl ersetzen. Die Ersetzung der *HighLevel*-Skripte wird unter anderem bei der Interpretation bzw. beim Abspielen der Animation verwendet, um atomare Befehle zu erhalten, die vom Skriptinterpreter umgesetzt werden können.

Intern im Skriptinterpreter findet zusätzlich eine Umwandlung des *LowLevel*-Skripts in eine *Frameliste* statt. Diese enthält für jedes zu erzeugende Einzelbild (Frame) einer Animation alle zur Erzeugung dieses Bildes einzustellenden Parameter. Hierdurch wird eine einfache sequentielle Abarbeitung der Animation ermöglicht. Mit den *LowLevel*-Skripten ist dies noch nicht möglich, da hier durch die mögliche Angabe von Zeitintervallen eine sequentielle Reihenfolge der Anweisungen noch nicht garantiert ist [Muehler 2005; Muehler et al. 2006]. Die *Frameliste* stellt dabei nur eine interne Repräsentation zur vereinfachten sequentiellen Abarbeitung von Skripten dar, während die *LowLevel*- und *HighLevel*-Skripte die Schnittstelle zum Autor einer Animation repräsentieren.

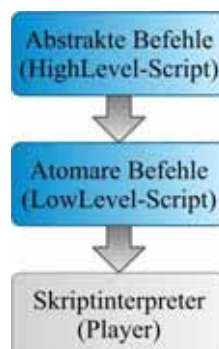
Der große Vorteil des beschriebenen Skriptkonzeptes liegt in der möglichen Kombination der unterschiedlichen Skriptlevel. So können in ein *HighLevel*-Skript *LowLevel*-Befehle eingearbeitet werden, die bei der Übersetzung in ein *LowLevel*-Skript erhalten bleiben. Dadurch ist es bei einem hohen Grad an Abstraktion in der Animationsdefinition dennoch möglich, Details zu kontrollieren.

Zusätzlich zur Erstellung von *LowLevel*- und *HighLevel*-Skripten können Autoren in sogenannten Ersetzungsregeln *HighLevel*-Befehle und ihre hierarchische Zerlegung zu *LowLevel*-Befehlen definieren.

### 9.4.2 Befehlssyntax und Aufbau der Skriptsprache

Die *HighLevel*- und *LowLevel*-Skripte setzen sich aus einem Initialisierungs- und dem Anweisungsabschnitt zusammen. Im Initialisierungsabschnitt werden Einstellungen vorgenommen, die Einfluss auf die Länge der Animation haben. Im Anweisungsabschnitt folgen die Anweisungen zur Animationssteuerung, wobei jede Anweisung in einer neuen Zeile steht.

Alle Anweisungen setzen sich aus vier Bereichen (Zeitangabe, Objekt, Befehl, Parameter) zusammen. Die Bereiche sind alle durch ein Leerzeichen voneinander abgegrenzt. Ein Beispiel wird in Listing 9.2 gegeben. Jeder der Bereiche kann bei der Umwandlung von einem *HighLevel*- in ein *LowLevel*-Skript durch spezialisiertere Angaben ersetzt werden.



**Abbildung 9.5:** Hierarchisches Skriptkonzept. Abstrakte *HighLevel*-Befehle bauen auf atomaren *LowLevel*-Befehlen auf. (modifiziert aus [Muehler 2005])

**Listing 9.2:** Beispiel einer Skriptanweisung, welche immer aus einer Zeitangabe, einem Objekt und dem darauf auszuführenden Befehl mit optionalen Parametern zusammengesetzt werden. (modifiziert aus [Muehler 2005])

---

```
%Time Object Command [Parameter]
[0,10] 'Liver' setColor brown
```

---

**Listing 9.3:** Möglichkeiten zur Angabe von Zeitpunkten und Zeiträumen für eine Anweisung. (modifiziert aus [Muehler 2005])

---

```
[5] ...           %Zeitpunkt
[0,10] ...        %Zeitraum
[start, 10] ...   %Zeitraum ab Animationsbeginn
[start, end] ...  %Zeitraum von Animationsbeginn bis Ende
```

---

## Zeitangaben

Die Zeitangabe erfolgt in eckigen Klammern, wobei Zeitpunkte als auch Zeitbereiche angegeben werden können (Listing 9.3). Bei der Angabe eines Zeitpunktes wird die Anweisung zu diesem Zeitpunkt ausgeführt. Bei Anweisungen mit Zeitbereichen wird versucht, die Anweisung über den angegebenen Zeitbereich zu interpolieren. Hierzu wird vom aktuellen Zustand der Szene ausgegangen und meist linear zum Parameterwert der aktuellen Anweisung interpoliert. Ausnahmen bilden Kamerabewegungen, welche auch über Bézier-Kurven für ein angenehmeres langsames Anfahren und Abbremsen interpoliert werden können [Muehler 2005]. Sollte eine Interpolation, zum Beispiel bei diskreten Anweisungen, nicht möglich sein, wird die Anweisung zu Beginn des Zeitbereiches ausgeführt.

Alle Zeiten werden hierbei in abstrakten Zeiteinheiten angegeben, da dies eine größere Flexibilität im Hinblick auf Länge und Framerate einer Animation bietet. Bei Angabe der Zeit in Sekunden müssten bei einer gewünschten Beschleunigung der Animation die Zeitangaben aller Anweisungen vom Autor entsprechend geändert werden. Bei einer Angabe in Frames ist die Wiedergabe auf langsameren Rechnern oder bei komplexeren Szenen, die nur eine geringere Framerate zulassen, verzögert. Bei der Angabe in Prozent der Gesamtlänge einer Animation müssten alle Werte angepasst werden, sobald die Animation um einige Anweisungen am Ende erweitert wird.

Die Umrechnung der abstrakten Zeiteinheiten in Sekunden erfolgt mit der durch den Autor im Initialisierungsabschnitt des Skriptes festgelegten Länge der Animation in abstrakten Zeiteinheiten (*LengthTimeUnit*) und in Sekunden (*LengthSeconds*) (siehe Listing 9.4). Dadurch kann die Geschwindigkeit einer Animation vom Autor, durch Anpassen der *LengthSeconds* ohne Veränderung anderer Zeitangaben im Skript, komfortabel angepasst werden. Zusätzlich kann genauso komfortabel durch gemeinsame Erhöhung von *LengthSeconds* und *LengthTimeUnit* eine Animation um weitere Anweisungen am Ende erweitert werden. Eine resultierende Framerate für die Wiedergabe wird erst im Skriptinterpreter berechnet, wodurch eine von der Rechenleistung unabhängige Definition der Animationen im Skript erreicht wird. Entsprechend sind die definierten Skripte auch auf heterogenen Rechnersystemen adäquat verwendbar.

Die Angabe der Zeiten für *HighLevel*- und *LowLevel*-Befehle ist dabei identisch. Für eine verbesserte Lesbarkeit und eine leicht verbesserte Anwendbarkeit ist es bei *HighLevel*-

**Listing 9.4:** Initialisierungsabschnitt eines Skriptes zur Definition der Animationslänge in abstrakten Zeiteinheiten und Sekunden. (modifiziert aus [Muehler 2005])

---

```
[Ini]      %Initialisierungsabschnitt
LengthTimeUnit=14
LengthSeconds=10

[Script]   %Anweisungsabschnitt
...
```

---

Befehlen zusätzlich möglich, die Schlüsselwörter **start** und **end** als Zeitangaben zu verwenden (siehe Listing 9.3). **start** und **end** werden dabei bei der Interpretation durch 0 bzw. *LengthTimeUnit* ersetzt.

## Objektbezeichnungen

An zweiter Stelle in einer Anweisung erfolgt die Bezeichnung eines Objektes, auf welches sich die Anweisung bezieht. Alle *HighLevel*- und *LowLevel*-Befehle beziehen sich dabei auf Objekte der Szene bzw. auf Systemobjekte. Verfügbare Systemobjekte sind **Cam** und **System**, welche zur Steuerung der Kamera bzw. zur Manipulation von Systemeigenschaften (z.B.: ändern der Hintergrundfarbe) verwendet werden.

Bezeichnungen von Objekten aus der Szene entsprechen bei *LowLevel*-Befehlen den Bezeichnungen, die den Objekten bei der Segmentierung zugewiesen wurden bzw. die sie innerhalb des Datensatzes haben. Diese Bezeichnungen stellen die sogenannten atomaren (nicht mehr teilbaren) Objektbezeichnungen dar. Von diesen atomaren Bezeichnungen können *HighLevel*-Objektbezeichnungen abstrahiert werden, welche durch Ersetzung im Skriptinterpreter in die atomaren Bezeichnungen überführt werden. *HighLevel*-Objektbezeichnungen können dabei in eine atomare Objektbezeichnung überführt werden (z.B.: kann die Bezeichnung 'Leber' durch die im Datensatz vorhandene Bezeichnung 'Leber, ROI#14' ersetzt werden). Zusätzlich können *HighLevel*-Objektbezeichnungen auch mehrere Objekte bündeln (z.B.: kann die Bezeichnung 'Lunge' durch 'Lungenfl. links' und 'Lungenfl. rechts' ersetzt werden).

Durch die Verwendung von *HighLevel*-Objektbezeichnungen können Animationsanweisungen auf ganze Objektgruppen oder gar alle Objekte der Szene angewendet werden. Zusätzlich ist dadurch die Adaption eines Skriptes an Datensätze mit unterschiedlich vielen Objekten und abweichenden Objektbezeichnungen möglich. Entsprechend sind Ersetzungsregeln für die *HighLevel*-Objektbezeichnungen nötig, welche näher in Abschnitt 9.4.3 beschrieben werden.

## Befehle

Der Befehl in einer Skriptanweisung gibt an, was zu der spezifizierten Zeit mit dem angegebenen Objekt in einer Anweisung geschehen soll. Hierzu stehen atomare Befehle sowohl zur Bewegung und Rotation von Objekten und der Kamera, als auch zur Manipulation von Eigenschaften der Darstellung der Objekte in einer Szene zur Verfügung.

Die Kamera kann zum Beispiel mit dem **move**-Befehl zu einer Position bewegt werden, welche sich relativ zu einem als Parameter übergebenen Objekt spezifizieren lässt. Verschiedene Rotationsbefehle ermöglichen die Rotation der Kamera oder eines Objektes um den

**Listing 9.5:** Beispiele für eine `move`-Anweisung. Durch die Angabe von Zeitbereichen wird die Kamera von ihrer aktuellen Position zur neuen Zielposition animiert. Abbildung 9.6 stellt die resultierende Kamerabewegung nach. (modifiziert aus [Muehler 2005])

---

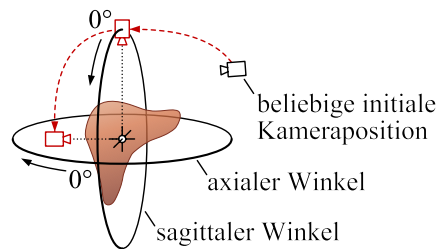
```
%[Zeit] 'Cam' move Objekt Winkel_sagittal Winkel_axial [Zoomfaktor]
```

---

```
[ 0,10] 'Cam' move 'Liver' 0 0
```

```
[10,20] 'Cam' move 'Liver' 90 90 2 %Zoomfaktor ist optional
```

---



**Abbildung 9.6:** Darstellung der Rotationswinkel zur relativen Positionierung der Kamera mit dem `move`-Befehl. Dabei wird zuerst um den sagittalen und dann um den axialen Winkel gedreht. Die Anweisungen für die dargestellten Kamerapositionen sind in Listing 9.5 zu finden. (modifiziert aus [Muehler 2005])

eigenen Mittelpunkt oder um ein spezifiziertes Objekt herum. Die Darstellung der Objekte kann z.B. mit `setVisible`, `setColor`, `setTransparency` und `setStyle` beeinflusst werden. Neben der Sichtbarkeit, Farbe und Transparenz eines Objektes kann damit auch der Visualisierungsstil, zum Beispiel zwischen Punkt- und Oberflächendarstellung, variiert werden (vergleiche auch *Parametrisierungsstufen* Abschnitt 9.3.1).

Die Spezifikation von Positionen und Bewegungen in Abhängigkeit von Objekten der Szene ermöglicht eine koordinatenfreie und adaptierbare Angabe dieser. Der `move`-Befehl (siehe Listing 9.5) erwartet zum Beispiel als Parameter ein Objekt, zwei Winkel und einen Zoomfaktor. Die Zielposition ergibt sich dabei aus den Drehungswinkeln in Sagittal- und Axialebene um den Mittelpunkt des angegebenen Objektes in einem dem Zoomfaktor entsprechendem Abstand. Abbildung 9.6 illustriert die aus Listing 9.5 resultierende Kamerabewegung. Ein Zoomfaktor von 1 bedeutet hierbei, dass das Objekt bzw. die *Bounding Sphere* des Objektes komplett in den Bildausschnitt passt, wobei größere Werte das Objekt größer darstellen. Ähnlich dem `move`-Befehl wird auch bei den Rotationsbefehlen verfahren und so eine Adaption der Animation an die Geometrie unterschiedlicher Szenen/Datensätze ermöglicht.

Zusätzlich zu den atomaren Befehlen können *HighLevel*-Befehle erzeugt werden, welche mehrere Befehle kapseln. Dadurch können abstraktere Anweisungen wie `'hebe Objekt X hervor'` realisiert werden. Die Umwandlung der *HighLevel*-Befehle in *LowLevel*-Befehle erfolgt ähnlich der Ersetzung von Objektbezeichnungen über definierte Ersetzungsregeln (siehe Abschnitt 9.4.3). Eine ausführliche Beschreibung aller atomarer Befehle kann in [Muehler 2005] gefunden werden.

## Parameter

Alle Angaben, die auf einen Befehl folgen, stellen Parameter dieses Befehls dar. Ein Befehl kann dabei mehrere Parameter besitzen, wobei hier auch zwischen *HighLevel*- und atomaren *LowLevel*-Parameterwerten unterschieden wird.

Zum Beispiel können abstrakte Farbangaben wie `rot` durch eine atomare Angabe in RGB-Werten `255,0,0` ersetzt werden. Als Parameter können auch Objektbezeichnungen (z.B.: wenn die Kamera ein Objekt umkreisen soll) und Geschwindigkeitsangaben (z.B. `slow`) nötig sein. Um diese auf atomare Parameter in atomaren Anweisungen zu überführen sind auch hier Ersetzungsregeln erforderlich, welche im nächsten Abschnitt näher besprochen werden.

### 9.4.3 Ersetzungsregeln

Ersetzungsregeln beschreiben, wie *HighLevel*-Objektbezeichnungen, -Befehle und -Parameter in atomare Anweisungen überführt werden können. In XML-Dateien werden diese Regeln für die Übersetzung der Skripte bereitgehalten. Vorteile der XML-Sprache sind hierbei, dass in XML strukturierte Inhalte zur maschinellen Verarbeitung, aber auch von Menschen lesbar repräsentiert werden können. Zusätzlich ist die XML-Struktur zur Repräsentation der Ersetzungsregeln geeignet, da sie eine hierarchische Schachtelung von Anweisungsblöcken ermöglicht.

Dadurch, dass die Ersetzungsregeln unabhängig vom Skriptinterpreter in XML-Dateien abgelegt werden, können diese Dateien um neue Anweisungen erweitert bzw. bestehende verändert und angepasst werden. Nachstehend werden Beispiele abstrakter *HighLevel*-Anweisungen und ihre Ersetzungsregeln vorgestellt.

#### Befehlsersetzungsregeln

Ein abstrakter *HighLevel*-Befehl ist der `view`-Befehl, welcher ein Objekt aus einer gewünschten Richtung zeigt. Auf diesem Befehl baut wiederum der `emphasize`-Befehl auf. Ein Objekt, auf das dieser Befehl angewendet wird, wird zunächst mit dem `view`-Befehl sichtbar gemacht und ins Zentrum der Ansicht gerückt. Danach wird das Objekt zusätzlich rot gefärbt und fast opak dargestellt. In Listing 9.6 sind ein Beispielskript und die Ersetzungsregeln zu sehen.

#### Parameterersetzungsregeln

Im Skript von Listing 9.6 wurden bisher nur die *HighLevel*-Befehle ersetzt. Da das Skript aber auch *HighLevel*-Parameter enthält, müssen diese auch in atomare Parameter überführt werden. Analog zu den Ersetzungsregeln für Befehle werden auch Regeln für die Ersetzung der Parameter definiert. Listing 9.7 zeigt die Ersetzung der Parameter aus Listing 9.6 mit den zugehörigen Ersetzungsregeln.

Abbildung 9.7 zeigt Ausschnitte aus der in Listing 9.6 definierten `emphasize`-Animation.

#### Objektersetzungsregeln

Die Ersetzung von abstrakten *HighLevel*-Objektbezeichnungen muss mit Rücksicht auf die in jedem Datensatz zur Verfügung stehenden Objekte und ihrer Bezeichnung in den Datensätzen angepasst werden. Bezieht sich eine Anweisung auf eine *HighLevel*-Objektbezeichnung, die durch mehrere Objekte ersetzt wird, so wird diese Anweisung für jedes der Objekte kopiert. Wird eine *HighLevel*-Objektbezeichnung als Parameter übergeben, so wird bei der Ersetzung eine Parameterliste mit allen atomaren Objektbezeichnungen erzeugt. In Listing 9.8 sind die Ersetzungsregeln beispielhaft für die Objektbezeichnung `'Lunge'` dargestellt.

**Listing 9.6:** *emphasize*-Befehl, Ersetzungsregeln und durch Ersetzung entstandene atomare Befehle. (modifiziert aus [Muehler 2005])

---

```

%Ausgangsanweisung
[0,10] 'Lunge' emphasize

%XML-Ersetzungsregeln
%Weitergabe der Anweisungsteile: T = Zeitangabe, O = Objekt, P = Parameter
<command commandStr = "emphasize">
    <command>T O view front 1</command>
    <command>T O setColor red</command>
</command>
<command commandStr = "view">
    <command>T O setVisible true</command>
    <command>T 'Cam' move O P</command>
</command>

%Ergebnis nach Ersetzung
[0,10] 'Lunge' setVisible true
[0,10] 'Cam' move 'Lunge' front 1
[0,10] 'Lunge' setColor red

```

---

**Listing 9.7:** Ersetzung der *HighLevel*-Parameter aus Listing 9.6.

---

```

%Ausgangsanweisung
[0,10] 'Lunge' setVisible true
[0,10] 'Cam' move 'Lunge' front 1
[0,10] 'Lunge' setColor red

%XML-Ersetzungsregeln
<command commandStr = "move">
    <parameter paramStr="front" singleValue="90 0" />
</command>
<command commandStr = "setColor">
    <parameter paramStr="red" singleValue="255,0,0" />
</command>

%Ergebnis nach Ersetzung
[0,10] 'Lunge' setVisible true
[0,10] 'Cam' move 'Lunge' 90 0 1
[0,10] 'Lunge' setColor 255,0,0

```

---





**Abbildung 9.7:** Ausschnitte aus der dynamischen Visualisierung der Lunge mit dem `emphasize`-Befehl aus Listing 9.6, 9.7 und 9.8. [Muehler 2005]

## Datensatzabhängigkeiten

Die Ersetzung von abstrakten Objektbezeichnungen durch reale Bezeichnungen von Objekten in einem gegebenen Datensatz erfordern, dass die Ersetzungsregeln für Objektbezeichnungen datensatzabhängig sind. Folglich muss für jeden Datensatz eine separate XML-Datei mit Ersetzungsregeln für die Objektbezeichnungen bereitgestellt werden. Diese können als zusätzliches Ergebnis der Bildanalyse automatisch generiert werden oder müssen für jeden neuen Datensatz manuell erzeugt werden.

**Listing 9.8:** Ersetzung der *HighLevel*-Objektbezeichnung 'Lunge' aus Listing 9.6.

---

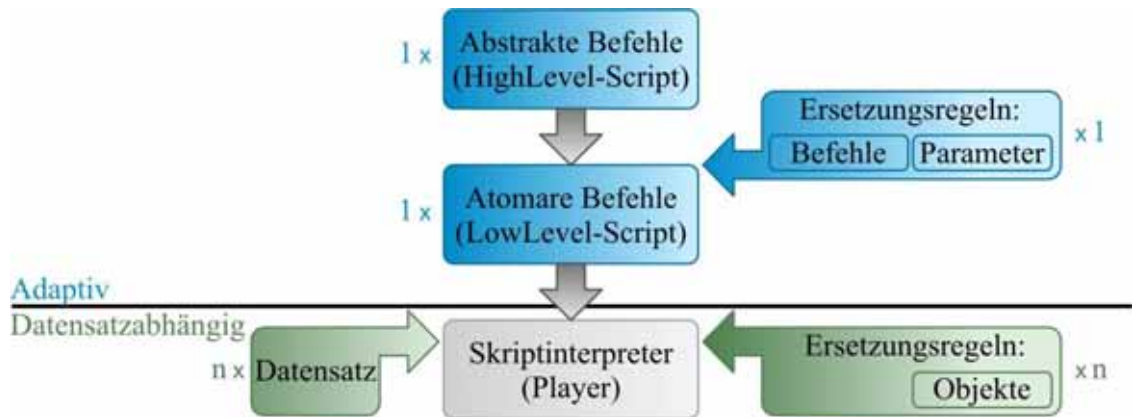
```
%Ausgangsweisung
[0,10] 'Lunge' setVisible true
[0,10] 'Cam' move 'Lunge' 90 0 1
[0,10] 'Lunge' setColor 255,0,0

%XML-Ersetzungsregeln
<object name = "Lunge">
    <object>Lungenfl. links</object>
    <object>Lungenfl. rechts</object>
</object>
<object name = "Lungenfl. links">
    <object>LiLungSeg, ROI1</object>
</object>
<object name = "Lungenfl. rechts">
    <object>ReLungSeg, ROI4</object>
</object>

%Ergebnis nach Ersetzung
[0,10] 'LiLungSeg, ROI1' setVisible true
[0,10] 'ReLungSeg, ROI4' setVisible true
[0,10] 'Cam' move 'LiLungSeg, ROI1' 'ReLungSeg, ROI4' 90 0 1
[0,10] 'LiLungSeg, ROI1' setColor 255,0,0
[0,10] 'ReLungSeg, ROI4' setColor 255,0,0
```

---





**Abbildung 9.8:** Hierarchische Ersetzung abstrakter *HighLevel*-Anweisungen zu *LowLevel*-Anweisungen durch animationsspezifische Ersetzungsregeln für Befehle und Parameter. Die Adaption der Skripte an beliebige Datensätze erfolgt dann über datensatzspezifische Ersetzungsregeln der Objektbezeichnungen.

Um die Generierung von adaptiven Animationen nicht durch diese Abhängigkeit von konkreten Datensätzen zu gefährden, wird die Ersetzung von Objektbezeichnungen erst bei der Erzeugung der *Frameliste* im Skriptinterpreter (siehe Abschnitt 9.4.1) vorgenommen.

Zusammenfassend wird für jede Animation ein Animationskript (mit *HighLevel*- und/oder *LowLevel*-Anweisungen) sowie ein XML-File mit den Ersetzungsregeln für Befehle und Parameter benötigt. Zusätzlich ist pro Datensatz ein XML-File mit den Ersetzungsregeln für die Objektbezeichnungen im Datensatz erforderlich. Das Zusammenspiel aller Komponenten wird zusammenfassend in Abbildung 9.8 dargestellt.

#### 9.4.4 Funktionalität des Skriptinterpreters

Mit dem vorgestellten Konzept können adaptive dynamische Visualisierungen definiert werden. Ein Skriptinterpreter bzw. Player für die Wiedergabe dieser dynamischen Visualisierung muss Skriptdateien der vorgestellten Syntax einlesen, *HighLevel*-Anweisungen in diesen mit Hilfe der XML-Ersetzungsregeln zu atomaren Anweisungen umformen sowie Positions- und Bewegungsanweisungen auf die Geometrie der Objekte eines Datensatzes anpassen können.

Folgende Minimalanforderungen sind dabei neben der gesamten Text und XML-Dateiverarbeitung an den Skriptinterpreter zu stellen:

- Bestimmung von Objektmittelpunkten und ihrer *Bounding Sphere* zur Auflösung der relativen Positions- und Bewegungsangaben.
- Interpolation von Parameterwerten und Kamerapfaden,
- Implementierung aller atomaren Anweisungen (siehe Tabelle 9.1).

Details der Implementierung seien hier bewusst ausgespart, da diese unabhängig vom Skriptkonzept sind.

## 9.5 Anwendungsszenarien des Skriptkonzeptes

Mit dem vorgestellten Skriptkonzept ist es möglich, dynamische Visualisierungen abstrahiert von den genauen zu visualisierenden Daten zu spezifizieren. Dadurch wird es möglich, einmal

**Tabelle 9.1:** Minimalumfang an *LowLevel*-Befehlen. (modifiziert aus [Muehler 2005])

[T]	<i>O</i> <b>move</b> <i>Objekt</i> $\angle$ - <i>sagi</i> $\angle$ - <i>axial</i> { <i>Zoom</i> }
	Bewegt Objekt <i>O</i> . Die Zielposition ergibt sich dabei aus der Drehung um die angegebenen Winkel in <i>axial</i> - und <i>sagittal</i> -Ebene zentriert am Mittelpunkt des angegebenen <i>Objekts</i> . Mit dem optionalen <i>Zoom</i> -Wert wird der Abstand zum Objekt bestimmt. Default: <i>Zoom</i> = 1 $\Rightarrow$ Objekt füllt die Ansicht voll aus.
[T]	<i>O</i> <b>rotate</b> <i>Achse</i> $\angle$ - <i>Winkel</i>
	Rotiert Objekt <i>O</i> . Dabei gibt <i>Achse</i> eine beliebige Rotationsachse und $\angle$ - <i>Winkel</i> den zu rotierenden Winkel um diese an.
[T]	<i>O</i> <b>rotate</b> { <i>X,Y,Z</i> } $\angle$ - <i>Winkel</i>
	Rotiert Objekt <i>O</i> um den angegebenen $\angle$ - <i>Winkel</i> um eine der Achsen des Weltkoordinatensystems.
[T]	<i>O</i> <b>rotateObj</b> { <i>X,Y,Z</i> } $\angle$ - <i>Winkel</i>
	Rotiert Objekt <i>O</i> um den angegebenen $\angle$ - <i>Winkel</i> um eine der Achsen des lokalen Koordinatensystems.
[T]	' <i>Cam</i> ' <b>rotateVirtual</b> { <i>X,Y,Z</i> } $\angle$ - <i>Winkel</i>
	Rotiert die Kamera an Achsenparallelen zur momentanen Sicht um den aktuellen <i>LookAt</i> -Punkt. Dabei zeigt die X-Achse immer nach rechts, die Y-Achse nach oben und die Z-Achse aus dem Bild heraus.
[T]	' <i>Cam</i> ' <b>setLookAt</b> <i>Objekt</i>
	Richtet die Kamera auf ein <i>Objekt</i> aus.
[T]	' <i>Cam</i> ' <b>setWatchUpVector</b> <i>true/false</i>
	Schaltet die automatische Korrektur der Kamera ein oder aus. Hierbei wird der <i>upVector</i> der Kamera immer so ausgerichtet, dass er in Richtung des Nordpols einer gedachten Kugel um das aktuelle <i>LookAt</i> -Objekt zeigt.
[T]	<i>O</i> <b>setVisible</b> <i>true/false</i>
	Setzt die Sichtbarkeitseigenschaft von Objekt <i>O</i> .
[T]	<i>O</i> <b>setColor</b> <i>R,G,B</i>
	Setzt die Farbe von Objekt <i>O</i> auf die angegebenen <i>R,G,B</i> -Werte $\in [0, 255]$ .
[T]	<i>O</i> <b>setTransparency</b> <i>Wert</i>
	Setzt die Transparenz von Objekt <i>O</i> auf den angegebenen <i>Wert</i> $\in [0, 1]$ .
[T]	<i>O</i> <b>setQuality</b> <i>Wert</i>
	Setzt die Darstellungsqualität von Objekt <i>O</i> auf den angegebenen <i>Wert</i> $\in [0, 1]$ .
[T]	<i>O</i> <b>setStyle</b> <i>Filled/Lines/Points</i>
	Setzt den Darstellungsstil von Objekt <i>O</i> auf eine gefüllte, eine Gitternetz- oder eine gepunktete Darstellung.
[T]	<i>O</i> <b>setSelected</b> <i>true/false</i>
	Selektiert oder deselektiert Objekt <i>O</i> .
[T]	' <i>System</i> ' <b>setBackground</b> <i>R,G,B</i> { <i>R,G,B</i> }
	Setzt die Hintergrundfarbe auf einen angegebenen <i>R,G,B</i> -Wert bzw. auf einen Verlauf vom ersten zum zweiten Farbwert.

erstellte Animationsskripte flexibel wiederzuverwenden. Dadurch sind, bei einer gegebenen Menge von  $n$  Datensätzen und einer geforderten Menge von  $m$  Animationen pro Datensatz, nur noch einmalig  $m$  Animationen durch einen Autor zu erstellen (vergleiche Abschnitt 9.2 und Abb. 9.1). Gleichzeitig können alle  $m$  Animationen von einmal definierten abstrakten *HighLevel*-Befehlen profitieren.

Für die Nutzung der Skripte ergeben sich folgende drei Hauptanwendungsfelder:

- Automatische Generierung von standardisierten Animationen für Datensätze in der Diagnose, Interventionsplanung und Ausbildung.
- Erzeugung von standardisierten statischen Ansichten unterschiedlicher Datensätze in der Diagnose, Interventionsplanung und Ausbildung.
- Integrierte Erzeugung von Animationssequenzen in interaktiven 3D-Visualisierungen.

Die hier genannten Anwendungsfelder vereinen dabei mehrere der in Abschnitt 9.1 erwähnten Visualisierungsaufgaben.

Nachstehend werden die drei genannten Anwendungsfelder näher erläutert. Aufgrund der Ähnlichkeit der Erzeugung standardisierter Animationen und statischer Ansichten werden diese gemeinsam besprochen. Auf die Integration des Skriptkonzeptes in interaktiven 3D-Visualisierungen wird mit dem Ziel der Navigationsunterstützung näher eingegangen.

### 9.5.1 Generierung standardisierter Ansichten und Animationen

Als Beispiel für die Erzeugung standardisierter dynamischer Visualisierungen sei aus [Tomandl et al. 2003] folgendes Szenario entliehen. Zur standardisierten Beurteilung von intrakraniellen Aneurysmen<sup>2</sup> generieren Tomandl et al. [2003] 3D-Videosequenzen von volumetrischen CT-Angiographieaufnahmen. Hierbei wird zunächst eine Videosequenz erstellt, bei der die gesamte Szene von der Kamera umfahren wird. Danach wird ein automatisch detektiertes Aneurysma in Nahaufnahmen mit der Kamera umfahren, wobei die Visualisierungsparameter der umliegenden Strukturen so angepasst werden, dass das Aneurysma gut zu erkennen ist. Mit dem vorgestellten Skriptkonzept können identische Animationen erreicht werden. In Listing 9.9 wird eine ähnliche Animation für onkologische Leberdatensätze definiert. Abbildung 9.9 zeigt für zwei unterschiedliche Datensätze Ausschnitte vom Heranfahren an den Tumor und beginnender Rotation um diesen.

Ähnlich kann bei der Erzeugung standardisierter statischer Ansichten verfahren werden. Hierzu sind alle Befehle, die die Kamera in die gewünschte Position bringen und Objekte der Szene nach Wunsch anpassen, zu einem Zeitpunkt auszuführen. Die ersten zwei Befehlszeilen in Listing 9.9 stellen entsprechende Anweisungen dar. In aufeinander folgenden Zeitschritten können damit auch unterschiedliche Ansichten der Szene erzeugt und ausgegeben werden.

Wie die einfachen Beispiele zeigen, können mit dem vorgestellten Skriptkonzept komfortabel dynamische Visualisierungen und statische Ansichten einer Szene erzeugt werden. Dadurch, dass die Skripte eine Adaption auf andere Szenen/Datensätze zulassen, können diese zur Standardisierung der 3D-Visualisierungen für Diagnose und Interventionsplanung beitragen.

Zusammenfassend können Visualisierungssysteme für patientenspezifische Daten davon profitieren, ihre statischen und dynamischen Ansichten über das vorgestellte Skriptkonzept zu

---

<sup>2</sup>Intrakranielle Aneurysmen sind krankhafte Ausweitungen von arteriellen Gefäßen im Hirn.

**Listing 9.9:** Beispielskript für die Erzeugung standardisierter Animationen für die Leberchirurgie. Hierbei wird zunächst durch Rotation der Kamera um die Szene ein Überblick vermittelt, um dann an den Tumor heranzufahren und dessen räumliche Lage durch eine erneute Rotation näher zu veranschaulichen.

---

```

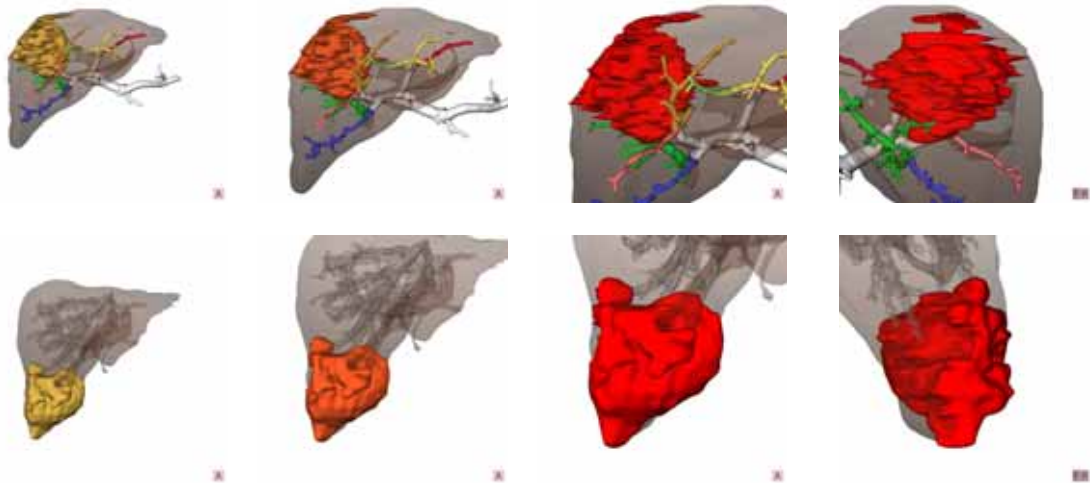
%HighLevel-Anweisung
[0] 'Leber' view front           %Initiale Sicht auf Leber
[0] 'Leber' setTransparency 0.3  %leichte Transparenz der Leber
[0,20] 'Leber' fullXRot          %360° Rotation um die Leber
[20,23] 'Leber' setTransparency 0.8 %starke Transparenz der Leber
[20,25] 'Tumor' emphasize front  %Zoom auf den Tumor
[25,35] 'Tumor' fullXRot         %360° Rotation um den Tumor

%XML-Ersetzungsregeln
<command commandStr = "emphasize">
    <command>T 0 view P</command>
    <command>T 0 setColor 255,0,0</command>
</command>
<command commandStr = "view">
    <command>T 0 setVisible true</command>
    <command>T 'Cam' move 0 P</command>
</command>
<command commandStr = "fullXRot">
    <command>T 'Cam' rotateX 360</command>
</command>
<command commandStr = "move">
    <parameter paramStr="front" singleValue="90 0" />
</command>
<object name = "Tumor">
    <object>T-HCC hem.lob.</object>
</object>

%Ergebnis nach Ersetzung
[0] 'Leber' setVisible true      %Initiale Sicht...
[0] 'Cam' move 'Leber' 90 0      %   -"-
[0] 'Leber' setTransparency 0.3  %leicht transparent
[0,20] 'Cam' rotateX 'Leber' 360 %360° Rotation...
[20,23] 'Leber' setTransparency 0.8 %stark transparent
[20,25] 'T-HCC hem.lob.' setVisible true %Zoom auf den Tumor
[20,25] 'Cam' move 'T-HCC hem.lob.' 90 0 %   -"-
[20,25] 'T-HCC hem.lob.' setColor 255,0,0 %Zoom auf den Tumor
[25,35] 'Cam' rotateX 'T-HCC hem.lob.' 360 %360° Rotation...

```

---



**Abbildung 9.9:** Heranfahren an einen Lebertumor mit gleichzeitiger farblicher Hervorhebung und anschließend beginnender Rotation in zwei unterschiedlichen Datensätzen. Ausschnitt aus der Animation nach Listing 9.9.

definieren. In solchen Systemen ist es dann ohne bzw. bei moderatem Mehraufwand möglich, dynamische Animationen zur Heranführung an eine Visualisierung bzw. zur Visualisierung von Übergängen zwischen einer aktuellen und einer neuen Darstellung der Szene zu generieren. Potentiell sollte davon die Akzeptanz und Erlernbarkeit von interaktiven Visualisierungsanwendungen profitieren. Insbesondere gilt dies für Anwendungen mit unerfahrenen Visualisierungsnutzern (z.B. Chirurgen) und für Anwendungen in Ausbildung und Lehre.

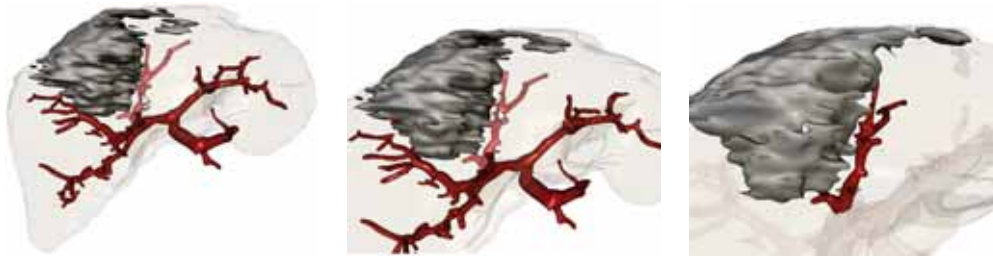
### 9.5.2 Dynamische Visualisierung im Interaktionskontext

Die Übernahme von bzw. Unterstützung bei kleineren Interaktionsaufgaben durch dynamische Visualisierungen soll aufwändige bzw. unnötig lästige Interaktionen vereinfachen.

Eine Standard-Interaktionsaufgabe ist das Navigieren an ein Objekt heran, welches dann näher untersucht wird. Unter anderem ist dies bei der Beurteilung von Lymphknoten sowie Tumoren und Blutgefäßen bei der Planung von Neck Dissections bzw. Lebertumoroperationen erforderlich (siehe Kapitel 2).

Diese Navigationsaufgabe kann durch ein geführtes Heranfahren in einer dynamischen Animation komfortabel unterstützt werden. Hierzu wird kurzzeitig die Führung von der Animation übernommen und der Anwender dynamisch ans Ziel geleitet. Ein einfaches Szenario ist, dass ein Anwender per Mausklick ein Objekt in der Szene selektieren kann, welches dann angeflogen wird. Im vorgestellten Skriptkonzept ist dies über die atomare `move`-Anweisung möglich. Die `move`-Anweisung kann dann von der aktuellen Sicht auf die Szene ausgehend mit dem selektierten Objekt im Skriptinterpreter aufgerufen werden.

Unter Nutzung des Skriptkonzeptes können auch komplexere Interaktionsaufgaben unterstützt werden. Abbildung 9.10 zeigt Ausschnitte einer komplexen Interaktionsunterstützung, bei der durch Anklicken mit der Maus ein Blutgefäßast ausgewählt wurde. An diesen wird, ausgehend von der aktuellen Anwendersicht, automatisch herangefahren. Gleichzeitig wird die Transparenz der anderen Gefäßäste sowie der sie umgebenden Leber stark erhöht. Damit wird automatisch eine Szene erzeugt, die eine gute Inspektion und Beurteilung des Gefäßverlaufs und Abstandes zum ebenfalls in der Szene enthaltenen Tumor ermöglicht. Durch das Auswählen eines anderen Gefäßes wird automatisch dieses angefahren und entsprechend



**Abbildung 9.10:** Dynamische Visualisierung zur Interaktionsunterstützung: (vlnr) Ein Gefäßast wird mit dem Mauszeiger ausgewählt, an diesen wird herangeflogen und alle anderen Gefäßäste werden stark transparent dargestellt.

die Szene verändert. Zusätzlich kann von dieser genauen Analyse des Gefäßverlaufs wieder automatisiert in eine Ansicht zurück gefahren werden, in der die ganze Leber wieder betrachtet und eine eventuelle Resektionsebene eingezeichnet werden kann. Alle Einstellungen und Veränderungen der Szene sowie das Heranfahren an das selektierte Objekt können bei rein manueller Steuerung unangenehm komplex und frustrierend sein.

Solche und ähnliche Animationsszenen können komfortabel mit dem vorgestellten Skriptkonzept definiert und an bestimmte Aufgaben angepasst werden. Hierdurch kann die Interaktion und Navigation in komplexen dreidimensionalen Visualisierungen zielgerichtet vereinfacht und auf wesentliche Aktionen beschränkt werden.

## 9.6 Zusammenfassung und kritische Anmerkungen

In diesem Kapitel wurden die Möglichkeiten und Aufgaben von dynamischen Visualisierungen betrachtet. Für die chirurgische Ausbildung und Interventionsplanung wurde dabei der Nutzen von adaptiven dynamischen Visualisierungen, vor allem für die Standardisierung des 3D-Präsentationsprozesses sowie zur Unterstützung in interaktiven Visualisierungen berücksichtigt.

In Abschnitt 9.4 wurde ein Skriptkonzept vorgestellt, welches die Definition von abstrakten Animationsbeschreibungen ermöglicht. Diese werden zur Wiedergabe an unterschiedliche Datensätze adaptiert, wodurch ein und dasselbe Skript für eine Vielzahl an Datensätzen genutzt werden kann. Hierdurch kann auf die manuelle Erstellung von dynamischen Visualisierungen für jeden Datensatz verzichtet und eine Standardisierung erreicht werden (Abschnitt 9.5.1).

Im Kontext interaktiver 3D-Visualisierungen erweist sich das adaptive Skriptkonzept zusätzlich als praktikabel. Die Angabe rein relativer Positions- und Bewegungsangaben macht hierbei eine Adaption von Animationen an den aktuellen Zustand der interaktiven Szene möglich, wodurch über die Skripte komplexe Interaktionsaufgaben unterstützt werden können (Abschnitt 9.5.2).

Vergleichend kann das vorgestellte Skriptkonzept als konkrete Umsetzung des Konzeptes der „Story Telling Informed“ Visualisierungen von Hauser und Wohlfart [2008] verstanden werden. Dabei wird dieses um die Adaption von Animationsbeschreibungen (Visualisierungsstories) an unterschiedliche Datensätze erweitert, wodurch einmal definierte Animationsbeschreibungen wiederverwendet und der manuelle Aufwand dramatisch reduziert werden können.

Beim Skriptkonzept ist kritisch sowie ausblickend anzumerken, dass dessen hier vorgestellte Fähigkeiten mit den definierten atomaren Befehlen noch sehr begrenzt sind. Dies trifft unter anderem auf die Verwendung der Objektmittelpunkte und *Bounding Volumes* (Kugeln, Boxen) zur internen Auflösung relativer Positionsangaben zu, welche bei elongierten und verzweigten Strukturen zu suboptimalen Ergebnissen führen können. Darüber hinaus kann über die in dieser Arbeit beschriebene Grundfunktionalität nicht garantiert werden, dass Objekte, die mit der Kamera anvisiert werden auch wirklich sichtbar sind und nicht durch andere Objekte verdeckt werden. Weiterführende Arbeiten beschäftigen sich deshalb mit der Integration intelligenter Verfahren zur Sichtpositions- und Sichtbarkeitsbestimmung [Mühler et al. 2007]. Darüber hinaus ist die Integration und Entwicklung von Methoden essenziell, welche die Sichtbarkeit verdeckter Strukturen im Kontext anderer zum Teil verdeckender Strukturen gewährleisten und Objekte im Fokus der Visualisierung geeignet hervorheben können (siehe [Krüger et al. 2005b; Bruckner et al. 2005; Viola et al. 2006]).

Die Definition von dynamischen Visualisierungen (Authoring) wird im vorgestellten Konzept durch die Definition abstrakter Befehle und die Wiederverwendbarkeit von definierten Animationsteilen unterstützt. Der kreative Authoringprozess von Animationen wird jedoch durch die Möglichkeit Skripte zu verfassen nicht ausreichend unterstützt. Zahlreiche Ansätze, die ein visuelles und komfortables Authoring ermöglichen, stehen zur Verfügung. Die Beschreibung abstrakter adaptierbarer Animationen stellt diese jedoch vor Herausforderungen, die über diese Arbeit hinausgehen und in zukünftigen Arbeiten zu betrachten sind.

Allgemein ist bei der Verwendung dynamischer Visualisierungen immer kritisch anzumerken, dass der Vorteil bzw. Nutzen von dynamischen Darstellungen gegenüber statischen Ansichten kontrovers diskutiert wird. Für die Anatomieausbildung von Medizinstudenten fanden Khalil et al. [2005a, b] heraus, dass Animationen gegenüber mehreren statischen Ansichten Probanden mit guter bis sehr guter räumlicher Vorstellungskraft verlangsamen. Im selben Vergleich verschlechterte sich die Lernleistung bei Probanden mit schlechter räumlicher Vorstellungskraft durch die ausschließliche Verwendung von Animationen.

Die dreidimensionale Interventionsplanung unterscheidet sich jedoch von der Anatomieausbildung insoweit, dass die grundlegende Anatomie bekannt ist und genaue räumliche Abstände der patientenindividuellen Anatomie beurteilt und meist auch interaktiv manipuliert werden müssen. Ein weiterer Aspekt ist, dass interaktive Visualisierungssysteme sowohl statische als auch dynamische Ansichten präsentieren. So wird eine statische Ansicht zum Beispiel durch das Rotieren der Szene zu einer dynamischen. Inwiefern und in welchem Verhältnis statische und/oder dynamische Visualisierungen hierbei Lernerfolge und räumliches Verständnis fördern bzw. fördern können ist unklar.





# Computerunterstütztes Training der Leberoperationsplanung

---

In diesem Kapitel wird die Anwendung und das Zusammenspiel der in dieser Arbeit besprochenen Aspekte in der chirurgischen Ausbildung und Interventionsplanung besprochen. Dies erfolgt beispielhaft bei der Vorstellung des LIVER SURGERY TRAINER, einem System für Ausbildung und Training in der Leberoperationsplanung. Hierzu wird das im Rahmen dieser Arbeit entwickelte Konzept des LIVER SURGERY TRAINER vorgestellt. In Abschnitt 10.1 wird zunächst die Entwicklung des LIVER SURGERY TRAINER motiviert. Daraufhin werden die Anforderungen und Ziele dargelegt (Abschnitt 10.2) sowie das entwickelte Konzept vorgestellt (Abschnitt 10.3 und 10.4). Vertiefend wird hierbei auf die Anwendung der in dieser Arbeit behandelten Aspekte der interaktiven und dynamischen Visualisierung eingegangen (Abschnitt 10.5).

## 10.1 Motivation

Die chirurgische Weiterbildung ist durch eine starke Abhängigkeit von chirurgischen Experten einerseits und von aktuell verfügbaren Fällen andererseits charakterisiert. Lernende sind dabei auf das Erlernen der Fähigkeiten örtlich verfügbarer Experten sowie auf das lokale Fallspektrum beschränkt. Lernsysteme mit einem repräsentativen Fallspektrum können diese Abhängigkeit verringern. Durch diese können alle drei Eckpfeiler der chirurgischen Ausbildung: *theoretische Chirurgie*<sup>1</sup>, *klinische Chirurgie*<sup>2</sup> und *Operationstechniken*<sup>3</sup> unterstützt werden [Mehrabi et al. 2000]. Neben dem gefahrlosen Trainieren von Operationstechniken ist die Auswahl der optimalen Operations- und Therapiestrategie von essentiellm Interesse. Mit den rasanten Entwicklungen neuer Operationstechnik und -techniken sowie mit den aktuellen Fortschritten in der computerunterstützten Interventionsplanung steigt der Trainings- und Ausbildungsbedarf auf diesem Gebiet.

Für das gefahrlose Trainieren der motorischen Fertigkeiten und physischen Durchführung von Eingriffen kommen Chirurgesimulatoren zum Einsatz (Abschnitt 2.7). Das Erlernen und Trainieren computerunterstützter Interventionsplanung gehört derzeit nicht zur chirurgischen Ausbildung (vergleiche Abschnitt 2). Entsprechend konzentriert sich die vorliegende Arbeit in diesem Kapitel auf das computerunterstützte Erlernen und Trainieren komplexer Interventionsplanung.

---

<sup>1</sup>Die *theoretische Chirurgie* beinhaltet alles theoretische Faktenwissen in der Chirurgie, z.B. über die Anatomie, OP-Vorbereitung und Nahtmaterial.

<sup>2</sup>Die *klinische Chirurgie* behandelt Wissen über Differentialdiagnose, Therapiemöglichkeiten und Behandlungsalternativen sowie prozedurales Wissen über die Therapieentscheidungsfindung und Interventionsplanung.

<sup>3</sup>Die *Operationstechniken* bilden die Gesamtheit aller überwiegend motorischen Techniken und Fähigkeiten zur Durchführung der unterschiedlichen Operationen.

### 10.1.1 Trainings- und Ausbildungsbedarf bei der Interventionsplanung

Die Operationsplanung besteht grundlegend aus der Identifikation und Abwägung verschiedener Therapiemöglichkeiten und der konkreten Planung eines durchzuführenden Eingriffs. Beide Teile der Planung sind dabei nicht disjunkt, so wird meist die als Goldstandard anerkannte Therapieform zuerst betrachtet und zum Teil konkret geplant. Zeigt die Planung jedoch Probleme auf, so müssen alternative Strategien herangezogen werden.

In der Leberchirurgie stellt die Resektion eines Tumors den Goldstandard dar. Ergibt die Resektionsplanung dass der Tumor zum Beispiel aufgrund eines zu geringem Abstands zu großen Blutgefäßen nicht reseziert werden kann, so muss über eine Chemotherapie bzw. Ablation des Tumors entschieden werden.

Bei der Interventionsplanung sind dabei meist komplexe räumliche Lagebeziehungen zu berücksichtigen. So sind Zugangswege, Abstände zu Risikogebieten sowie das durch die Intervention zu erwartende Schädigungsausmaß abzuschätzen und in der Planung zu berücksichtigen. Strategien zur Entscheidungsfindung bei unterschiedlichsten Erkrankungen unter Berücksichtigung aller nötigen Parameter und Alternativen müssen dabei erst erlernt und in ein mentales Modell überführt werden.

Computerunterstützte Interventionsplanungssysteme können komplexe Entscheidungsfindungs- und Planungsprozesse zum Beispiel durch eine detaillierte Risikoanalyse unterstützen (vergl. Abschnitt 2.2). Der effiziente Umgang und erfolgreiche Einsatz dieser Systeme muss jedoch auch erlernt und trainiert werden.

Durch detaillierte Planungssysteme wurden speziell in der Leberchirurgie bereits neue Therapie- und Planungsstrategien entwickelt [Lang et al. 2005b]. Diese bauen auf der computergestützten Berücksichtigung der genauen Patientenanatomie und Simulation von Therapieergebnissen auf und können dadurch, vor allem bei komplexen Interventionen (z.B. wiederholten Tumorresektionen in der Leber) [Lang et al. 2005b], ein schonenderes Vorgehen sowie die Reduktion von möglichen intra- und postoperativen Komplikationen ermöglichen. Zusätzlich stehen durch Fortschritte in der Operationstechnik vermehrt unterschiedliche Therapiemöglichkeiten zur Verfügung. Ihre Anwendbarkeit und ihr Nutzen im Vergleich zu anderen Techniken muss erlernt und ihre Planung trainiert werden. Neben Chirurgiesimulatoren für neue Operationstechniken sind zur Steigerung der breiten Anwendung neuer Operationstechniken und Planungsmöglichkeiten auch Ausbildungssysteme für die Interventionsplanung nötig.

### 10.1.2 Herausforderungen der Interventionsplanung in der Leberchirurgie

Die Resezierbarkeit (*resectability*) von Lebertumoren ist teilweise bei zentraler Tumorage, räumlich entfernten bzw. in beiden Leberlappen liegenden Metastasen sowie bei wiederholten Eingriffen (Rezidivoperationen) schwer zu beurteilen. Wichtigstes Kriterium für die Resezierbarkeit ist dabei, ob ausreichend gesundes und von allen Gefäßsystemen gut versorgt und entsorgtes Lebergewebe erhalten werden kann (Abschnitt 2.2).

Weitere Planungsentscheidungen betreffen die Anwendbarkeit von Ablationen zur lokalen thermischen oder chemischen Zerstörung von Tumoren sowie den Zugang und die Platzierung der dazu nötigen Applikatoren. Zu beurteilende Kriterien sind hierbei unter anderem die Größe des bzw. der Tumore und deren Lage in Bezug zu Blutgefäßen. Letztere besitzen

eine starke kühlende Wirkung, welche bei der thermischen Ablation zu lokal stark unterschiedlichen Schädigungsvolumina führen kann.

Für die Planung von Resektionen und Ablationen ist daher immer die patientenindividuelle Gefäßanatomie sowie Tumorlokalisation und -morphologie zu berücksichtigen. Diese muss geeignet aus den radiologischen Bilddaten extrahiert und visualisiert werden. Zusätzlich müssen Möglichkeiten zur Exploration der 3D-Daten sowie zur interaktiven Manipulation angeboten werden. Weitere Computerunterstützung ist bei der Bestimmung des funktionalen Restvolumens sowie bei der Simulation der Ablationswirkung erforderlich.

Herausforderungen, die über den Rahmen des LIVER SURGERY TRAINER hinaus gehen, sind unter anderem Entscheidungen bezüglich möglicher Chemotherapien zur Verkleinerung der Tumore vor Operationen. Entsprechend kombinierte Therapien werden in interdisziplinären Tumorboards besprochen.

Die im Rahmen dieser Arbeit relevanten Herausforderungen der computerunterstützten Interventionsplanung betreffen die Rekonstruktion und Aufbereitung patientenindividueller Oberflächenmodelle sowie Möglichkeiten zur effizienten (semi-)interaktiven Exploration, Navigation und dynamischen Visualisierung (vergl. Abschnitt 2.8.1).

## 10.2 Anforderungen und Ziele

Um die Entscheidungsfindung und Interventionsplanung in der Leberchirurgie bei Lebertumoren praxisnah trainierbar zu machen, werden nachfolgend die Ziele für ein entsprechendes Lernsystem formuliert und Anforderungen an das System abgeleitet.

### 10.2.1 Ziele

Ziel des LIVER SURGERY TRAINER ist das praxisorientierte Vermitteln und Trainieren von Therapieentscheidung und Interventionsplanung in der Leberchirurgie.

Hierbei soll es unter anderem Viszeralchirurgen und Chirurgen in der Ausbildung zum Viszeralchirurgen ermöglicht werden, praxisorientiert an virtuellen Fällen die Interventionsplanung durchzuführen. Unterstützt werden soll dies durch die Vermittlung von Planungsstrategien, Expertenvorschlägen und Kritik am eigenen Vorgehen. Hierdurch soll die Lernkurve bei der Interventionsplanung deutlich beschleunigt, aber auch die Weiterbildung von Experten komfortabler werden. Ein entsprechendes Lernsystem soll auch zur Vor- und Nachbereitung von Operationen verwendet werden können (siehe *blended learning* [Tang und Byrne 2007]). Ein langfristiges Ziel ist dabei die bessere Verbreitung von neuen Operationstechniken und Interventionsplanungsmöglichkeiten.

Aus den aufgestellten Zielen ergeben sich die in den folgenden Abschnitten erläuterten Anforderungen an das Konzept des LIVER SURGERY TRAINER.

### 10.2.2 Zielgruppenspezifische Anforderungen

Die Zielgruppe des LIVER SURGERY TRAINER sind Viszeralchirurgen sowie Fachärzte (Chirurgen) in der Weiterbildung zum Viszeralchirurgen. Bei dieser Zielgruppe ist bereits von einer hohen Lernstufe bei fachspezifischem Faktenwissen auszugehen bzw. wird dieses im Rahmen der klassischen Ausbildung erworben. Entsprechend ist Spezialwissen über die Interventionsplanung und Planungsstrategien zu vermitteln. Weiterhin kann davon ausgegangen werden,

dass in der Zielgruppe selbstständiges analytisches und abstraktes Denken und Handeln beherrscht wird.

Die Medienkompetenz und Erfahrung in der Computerbedienung kann jedoch über die Zielgruppe sehr heterogen verstreut sein. Entsprechend muss auch Computerneulingen der Einstieg ermöglicht werden.

Zielgruppenspezifisch werden Lernort und Lernsituation stark vom individuellen Selbststudium geprägt sein. Hierbei wird meist einzeln an Rechnern in der Klinik und privat gelernt. Für den Einsatz in der Klinik muss gewährleistet werden, dass auch ohne Audionutzung die Lerninhalte erfassbar sind. Darüber hinaus ist neben Spezialrechnern allgemein bei der Rechnerausstattung in der Zielgruppe von Standardsystemen auszugehen, die über keine erweiterten Grafikfähigkeiten und Eingabegeräte verfügen.

Durch das zielgruppenspezifische Selbststudium ist weiterhin ein selbsterklärendes System mit automatischer Einweisung, kontextabhängiger Hilfe und Schritt-für-Schritt Anleitung erforderlich. Als automatische Einweisung können komplexe Bedienungshandlungen anfänglich vom System vorgeführt und erklärt werden. Durch den klinischen Alltag wird die Lernzeit, die der Zielgruppe zur Verfügung steht, relativ gering sein. Dies erfordert schnell und gezielt zugängliche Informationen im System sowie eine zügige Wissensvermittlung.

Die wichtigsten Anforderungen, die sich folglich aus der Zielgruppenspezifität ableiten, sind:

- Möglichkeiten zur zielgerichteten Auswahl von zu bearbeitenden Fällen bzw. Problemen,
- logisch strukturierte Informationseinheiten in den Fällen für einen leichten Zugang zu gesuchten Informationen,
- unterschiedliche Hilfestellungen für unterschiedliche Benutzergruppen bezüglich dem domänenspezifischen Spezialwissen und den Erfahrungen in der Bedienung der computerunterstützten Leberoperationsplanung und des Lernsystems,
- Auswahl von Trainingseinheiten nach Zeitaufwand und Schwierigkeitsgrad sowie
- Standard-PC-fähige 3D-Visualisierungs- und Interaktionskonzepte.

Die Anforderungen lassen sich dabei hauptsächlich auf eine geeignete Fallauswahl und Fallstrukturierung sowie auf ein komplexes Hilfesystem übertragen.

### 10.2.3 Anforderungen aus dem medizinischen Lerninhalt

Die wichtigsten sich aus dem medizinischen Lerninhalt ergebenden Anforderungen betreffen die Fallbasis und die interaktive dreidimensionale Präsentation.

**Klinische, patientenspezifische Fälle:** Bei der Interventionsplanung in der Leberchirurgie ist besonders die patientenspezifische Anatomie und Morphologie der unterschiedlichen Krankheiten zu berücksichtigen. Durch die starken Variationen der gesunden und pathologischen Leberstrukturen gibt es keinen sogenannten „Standardfall“, an dem alles nötige Wissen vermittelt und trainiert werden kann. Trainiert werden muss hier vor allem die Anwendung von Planungsstrategien auf ganz unterschiedliche patientenspezifische Fälle. Diese Fälle sollten dabei zum Training nicht die Möglichkeiten klinischer Daten überschreiten.

**Interaktive 3D-Visualisierung:** Die Interventionsplanung bezieht sich auf patientenspezifische Bilddaten, in denen die gegebenen räumlichen Beziehungen in der Planung zu berücksichtigen sind. Zum Erlernen und Trainieren der Interventionsplanung sind deshalb, wie auch bei der computerunterstützten Planung selbst, interaktive dreidimensionale Visualisierungen einzusetzen.

**Adaptivität:** Zum effizienten Umgang mit den geforderten klinischen Fällen und der interaktiven Visualisierungen dieser muss das Lernsystem adaptiv auf die patientenindividuellen Daten eingehen können.

### 10.2.4 Lerntheoretische Anforderungen

Aus der Zielgruppenspezifik und dem Lerninhalt lassen sich die nachstehenden lerntheoretischen Eigenschaften ableiten:

- Es findet ein überwiegend selbstorganisiertes Lernen statt.
- Es müssen komplexe Lerninhalte und Problemlösestrategien erlernt werden.
- Es muss der Praxisbezug zur Klinik gewahrt werden (z.B. keine idealisierte Anatomie möglich).

Berücksichtigung können diese geforderten Eigenschaften vor allem in den Lernparadigmen Kognitivismus und Konstruktivismus finden. Hierbei wird in der Synthese beider Paradigmen realitätsnahes problemorientiertes Lernen ermöglicht [Issing und Klimsa 1997]. Hierbei werden komplexe Lerninhalte strukturiert, aber nicht vereinfacht. Dies führt zu den lerntheoretischen Anforderungen:

- Es ist eine problemorientierte Lernsituation zu erschaffen.
- Das System muss das Entwickeln eigener Lösungsstrategien und -wege zulassen (keine starr vorgegebenen Interaktionen und Abläufe).
- Komplexe Lerninhalte werden strukturiert bereitgestellt und die Lernenden werden bei der Konstruktion eigenen Wissens unterstützt (z.B. durch tutorielle Steuerung).
- Fehler der Lernenden sind durch Feedback aufzugreifen und zu korrigieren (auch zeitversetzt).

Diese Anforderungen werden im nachstehenden Konzept des LIVER SURGERY TRAINER aufgegriffen, wobei konkrete Systemstrukturen zur Umsetzung der Anforderungen erläutert werden.

## 10.3 Konzept des LIVER SURGERY TRAINER

Der LIVER SURGERY TRAINER wird entsprechend den Anforderungen als fallbasiertes Lernsystem zum problemorientierten Lernen entworfen. Konzeptionell besteht dabei der LIVER SURGERY TRAINER aus verschiedenen Lernmodulen, die sich linear und hierarchisch in kleinere Module unterteilen. Auf höchster Ebene besteht der LIVER SURGERY TRAINER dabei aus einer Lehrbuchkomponente und einer Fallbearbeitungskomponente. Die Lehrbuchkomponente stellt Faktenwissen ähnlich einem Lehrbuch zur Verfügung. Dieses ist einzeln oder

während der Fallbearbeitung über Querverweise zugänglich. Im Vergleich zur Fallbearbeitungskomponente stellt die Lehrbuchkomponente einen sehr kleinen Teil des LIVER SURGERY TRAINER dar, so dass diese hier nicht weiter betrachtet wird.

Für die Fallbearbeitungskomponente stellt eine Falldatenbank alle fallspezifischen Daten bereit. Über eine Fallauswahl können je nach Problemstellung gezielt Fälle herausgesucht werden. Hierbei wird unterstützt, dass man Fälle einem gegebenen Problem entsprechend (z.B. zentrale Tumorlage) bzw. in Ähnlichkeit zu realen Patienten wählen kann. Anhand der Fälle kann dann eine Interventionsplanung durchgeführt werden, wobei das System auf unterschiedlichen Niveaus Unterstützung anbietet. Die Planung im Lernsystem erfolgt dabei identisch dem in Abschnitt 2.2 beschriebenen Vorgehen unterstützt durch die gleichen Planungssysteme.

Im Folgenden werden die unterschiedlichen Komponenten des LIVER SURGERY TRAINER kurz erläutert.

### 10.3.1 Falldatenbank und Datenbasis

Der LIVER SURGERY TRAINER basiert auf Daten aus aktuellen Interventionsplanungsprozessen. Dabei handelt es sich um klinische Daten, welche für die Planung bereits aufgearbeitet (u.a. segmentiert und Oberflächenmodelle extrahiert) wurden. Zu diesen Daten werden zusätzlich Informationen der operierenden Experten zu Planungsdetails und Interventionsverlauf erhoben sowie Planungsalternativen und Expertenvorschläge für eine gute Planung hinzugefügt. Zusätzlich werden vom Operateur kommentierte Videoaufzeichnungen der durchgeführten Intervention angefertigt und bereitgestellt.

Für die Falldatenbank werden dabei Fälle aus der Praxis so ausgewählt, dass ein repräsentatives Spektrum unterschiedlicher Erkrankungen, erforderlicher Therapien, Tumorlokalisierung und -anzahl sowie verschiedener Schwierigkeitsgrade abgedeckt wird. Videoaufzeichnungen werden dabei nur bei einigen wenigen, medizinisch interessanten oder klinisch allgemeinen Fällen angefertigt, so dass für ähnliche Fälle auf diese Aufzeichnungen verwiesen werden kann bzw. die Besonderheiten bestimmter Fälle besser herausgearbeitet werden können.

Die wichtigste Schnittstelle für Lernende zur Falldatenbank im LIVER SURGERY TRAINER ist dabei die Fallauswahl. Das Visualisierungs- und Interaktionskonzept dieser wird in Abschnitt 10.5.1 vorgestellt.

Zusammenfassend unterstützt das beschriebene Konzept der Falldatenbank die Verwendung patientenspezifischer klinischer Daten. Auf dieser Grundlage kann das geforderte realitätsnahe und problemorientierte Trainieren aufgebaut werden.

### 10.3.2 Fallbearbeitung

Das Erlernen und Trainieren von Interventionsplanung und assoziierten Entscheidungsprozessen erfolgt im Lernsystem durch die Bearbeitung von Fällen. Fälle für die Bearbeitung können dabei in der Fallauswahl problemorientiert ausgewählt werden. Die anschließende Fallbearbeitung erfolgt in logischen Schritten, welche in Abbildung 10.1 skizziert sind. Die einzelnen Schritte sollten dabei in ihrer logischen Reihenfolge abgearbeitet werden, wobei es auch möglich ist, Schritte zu überspringen. Letzteres ermöglicht dabei das zügige Anwählen gewünschter Informationen (z.B. Expertenvorschläge).

In den Schritten *Fallvorstellung* und *durchgeführte Therapie* werden hauptsächlich einführende Informationen bzw. Informationen über die in echt angewendete Therapie für den aktuellen Fall präsentiert.

Der Schritt *Interventionsplanung* stellt die interaktive Hauptkomponente der Fallbearbeitung dar. Diese teilt sich konzeptionell in die nachstehenden Sub-Schritte:

**Therapieentscheidung:** In diesem Schritt wird initial eine erste Entscheidung hinsichtlich der zu planenden Therapie getroffen (z.B. Resektion). Diese kann dann im weiteren Planungsschritt wieder verworfen werden, z.B. wenn sich bei der Planung der Resektion ergibt, dass dabei nicht genügend funktionsfähiges Gewebe erhalten werden kann.

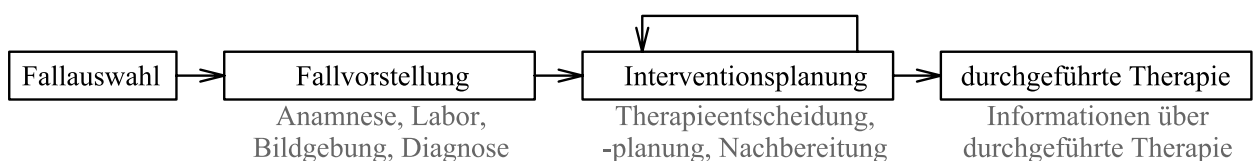
**Therapieplanung:** Hier wird die konkrete Planung der Therapie mit Definition von Resektionsebene oder Applikatorplatzierung durchgeführt. In Spezialfällen kann es dabei auch nötig sein, mehrere Resektionsebenen zu definieren oder Kombinationen aus Resektion und Ablation zu planen. Grundlegend können hier Lernende interaktiv verschiedene Strategien testen und gegenüberstellen. Dies erfolgt selbstgesteuert oder mit einstellbarer Unterstützung durch das Lernsystem. Weiterführend wird der Sub-Schritt in weitere kleinere Lernziele unterteilt, auf deren Aufzählung hier verzichtet wird. In [Mirschel 2004; Schmidt 2005; Riedel 2005] können diese nachgelesen werden.

**Nachbereitung:** Zur Nachbereitung können Lernende ihre aufgestellte Planung bewerten lassen sowie Expertenvorschläge und Kommentare zum aktuellen Fall einsehen (Abb. 10.2). In medizinisch besonders relevanten Fällen können hier auch Videoaufzeichnung der durchgeführten Intervention und andere Zusatzinformationen abgerufen werden.

### 10.3.3 Interaktive Patientenakte

Die interaktive Patientenakte unterstützt die Lernenden bei der Bearbeitung eines Falls in Form eines elektronischen Notizbuches. Sie kann jederzeit und in jedem Schritt der Fallbearbeitung aufgerufen werden. In ihr können fallspezifische Dokumente und Informationen von den Lernenden „abgelegt“ werden. Beim Auslassen bzw. Überspringen einzelner Schritte (z.B. bei gezielter Informationssuche oder gezieltem Training von Teilaspekten) werden automatisch elementare Informationen (z.B.: Name und Alter des Patienten, diagnostisch relevante radiologische Bilder u.ä.) aus den übersprungenen Schritten in die Patientenakte aufgenommen.

Beim interaktiven Füllen der Patientenakte können aktuell betrachtete Schichtbilder und Ansichten der 3D-Szene in die Patientenakte aufgenommen sowie grafisch annotiert und beschriftet werden. Entsprechend können in der Patientenakte für den Planungsprozess von den Lernenden wichtige Informationen verwaltet, erweitert und bei Bedarf wieder abgerufen werden.



**Abbildung 10.1:** Fallablaufschemata im LIVER SURGERY TRAINER. (modifiziert aus [Mirschel 2004])

Abbildung 10.3 zeigt den Zustand einer Patientenakte von zwei aufeinanderfolgenden Schritten der Fallbearbeitung. Die Möglichkeit, interaktiv Informationen in der Patientenakte abzulegen, unterstützt dabei Lernende bei der Organisation und Konstruktion eigener mentaler Repräsentationen des aktuellen Falls. Über die Patientenakte kann zusätzlich effizient auf die hinterlegten Informationen aus den bereits bearbeiteten Fallbearbeitungsschritten zugegriffen werden, ohne den aktuellen Schritt zu verlassen. Darüber hinaus ist es in der Patientenakte möglich, eine geplante Therapie abzulegen, um alternative Therapievarianten ausprobieren, verwerfen, verfeinern und letztendlich die beste identifizieren zu können.

### 10.3.4 Hilfekonzzept

Alle Schritte im Lernsystem werden von einer mehrstufigen Hilfe begleitet. Grundlegend wird bei der Unterstützung der Lernenden zwischen *inhaltlichen Hilfen* und *Bedienungshilfen* unterschieden. Erstere beinhalten Informationen zu Planungsstrategien und bewerten das bisher Geleistete, um Korrekturvorschläge und Hinweise für das weitere Vorgehen geben zu können. Ein Beispiel für eine inhaltliche Hilfe ist die Analyse einer geplanten Resektionsebene in Bezug auf den Abstand zum Tumor und das Volumen des resezierten und verbleibenden Gewebes mit einer qualitativen Einschätzung des Planungsergebnisses. Auch Vergleiche zum hinterlegten Expertenvorschlag mit Angabe der Unterschiede können zur inhaltlichen Unterstützung dienen.

Neben den inhaltlichen Hilfen werden auch Hilfen zur Bedienung des Lernsystems und der Planungskomponenten angeboten. Hauptsächlich führen diese Hilfen in die komplexe Bedienung und Steuerung der Interventionsplanungskomponenten ein und unterstützen diese auf unterschiedlichen Hilfestufen. Grundlegende Bedienungshilfen unterstützen zusätzlich Lernende mit geringer Computererfahrung bei der allgemeinen Programmbedienung.

Inhaltliche, wie auch Bedienungshilfen erfolgen dabei konzeptionell in unterschiedlicher Granularität auf unterschiedlichen Unterstützungsstufen. Realisiert wird dies über die Hilfemodi [Riedel 2005]:

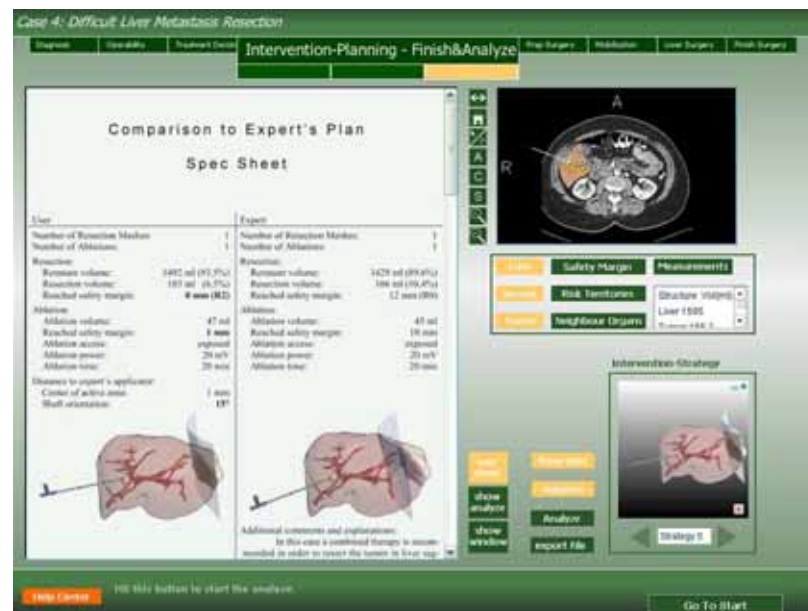
- Präsentation
- Geführte Tour
- Automatische Hilfe
- Hilfe auf Anfrage

**Präsentation:** Bei der Präsentation wird die Steuerung komplett vom System übernommen und die Interaktion ähnlich einem Video vorgeführt.

**Geführte Tour:** Bei der geführten Tour werden die Interaktionsabläufe Schritt-für-Schritt (geführt) durchlaufen. Hierbei werden Lernende sukzessiv instruiert einen Interaktionsschritt durchzuführen.

**Automatische Hilfe:** Bei der automatischen Hilfe werden kontextabhängig automatisch Hilfen angezeigt. So werden beim Definieren einer Resektionsebene die einzelnen nötigen Interaktionsschritte textuell beschrieben angezeigt.

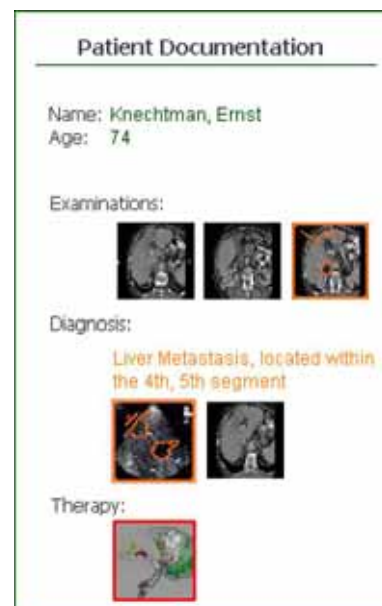




**Abbildung 10.2:** Bewertung einer im LIVER SURGERY TRAINER geplanten Intervention mit Gegenüberstellung des Expertenvorschlags.

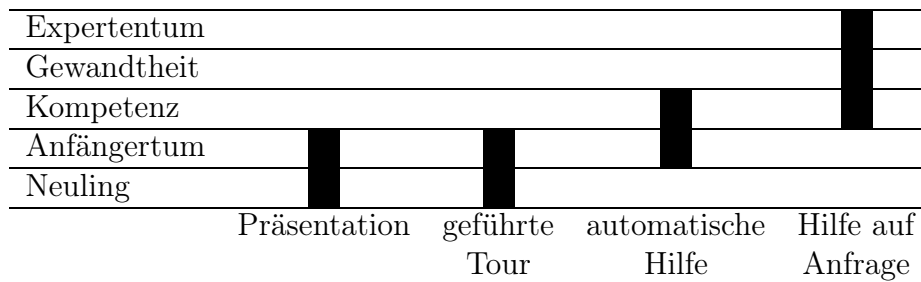


(a)



(b)

**Abbildung 10.3:** Interaktive Patientenakte zur Aufbewahrung relevanter Informationen: (a) Mit annotierten Schichtbildern gefüllte Patientenakte nach Einsicht der Diagnose des aktuellen Falls, (b) Patientenakte weiter gefüllt mit einer 3D-Ansicht der aktuellen Therapieplanung.



**Abbildung 10.4:** Adäquate Unterstützung auf den unterschiedlichen Lernstufen: *Neuling*, *Anfängertum*, *Kompetenz*, *Gewandtheit* und *Expertentum* nach [Baumgartner und Payr 1999] durch die Hilfemodi: *Präsentation*, *geführte Tour*, *automatische Hilfe* und *Hilfe auf Anfrage* im LIVERSURGERYTRAINER.

**Hilfe auf Anfrage:** Bei Hilfe auf Anfrage werden Hilfen nur bei direkter Anforderung z.B. durch Drücken der F1-Taste angezeigt.

Alle Hilfemodi sind hierbei untereinander verknüpft, so dass man sich z.B. in der geführten Hilfe bei Bedarf einzelne Schritte im Präsentationsmodus ansehen kann. Durch das Konzept der unterschiedlichen Hilfemodi können Lernende auf unterschiedlichen Lernstufen effizient unterstützt werden [Mader und Stöckl 1999].

Baumgartner und Payr [1999] unterscheiden fünf Lernstufen: *Neuling*, *Anfängertum*, *Kompetenz*, *Gewandtheit* und *Expertentum*. Ein Neuling ist mit der zu lernenden Sache noch nicht vertraut, während auf der Lernstufe des Expertentums sowohl die Wahrnehmung der aktuellen Situation, als auch Problemlösungsentscheidungen bereits so verinnerlicht wurden, dass diese ohne bewusste Aufmerksamkeit ablaufen. Abbildung 10.4 illustriert hierbei die Eignung der verwendeten Hilfemodi für die unterschiedlichen Lernstufen. Hierbei eignet sich die *Präsentation* vorrangig für Neulinge, während auf der Stufe der Gewandtheit und dem Expertentum die *Hilfe auf Anfrage* von Relevanz ist.

### 10.3.5 Implementierungsaspekte

Eine Herausforderung bei der Implementierung des LIVERSURGERYTRAINER liegt in der nötigen Integration unterschiedlicher Medien:

- medizinische Bild-/Volumendaten,
- rekonstruierte 3D-Modelle,
- Videoaufzeichnung von Operationen,
- animierte Lersequenzen sowie
- anderes Bild-, Text- und Tonmaterial.

Eine zweite Herausforderung besteht darin, das Ausprobieren und Trainieren der computerunterstützten Interventionsplanung im Rahmen des Lernsystems realitätsnah zu ermöglichen. Darüber hinaus müssen Navigations- und Hilfestrukturen bereitgestellt werden, welche die Lernenden im System unterstützen.

## 10.4 Implementierungsvarianten

Grundlegend können Lernsysteme für die computerunterstützte Interventionsplanung in drei unterschiedlichen Programmkonstellationen umgesetzt werden:

- Umsetzung in Autorensoftware für Lernsysteme und/oder Multimedia-Anwendungen,
- Umsetzung im Interventionsplanungssystem sowie
- Kombinierte Umsetzung in Autorensoftware und Interventionsplanungssystem, wobei Lernsystemkomponenten in der Autorensoftware implementiert und Komponenten zur Interventionsplanung aus bestehenden Systemen zur computerunterstützten Planung integriert werden.

Alle drei Implementierungsvarianten bieten dabei unterschiedliche Vorteile, aber auch Fallstricke. Im Folgenden werden diese kurz erläutert und zusammenfassend mit Blick auf die Entwicklung von Lernsystemen für die Interventionsplanung diskutiert.

### Umsetzung in Autorensoftware

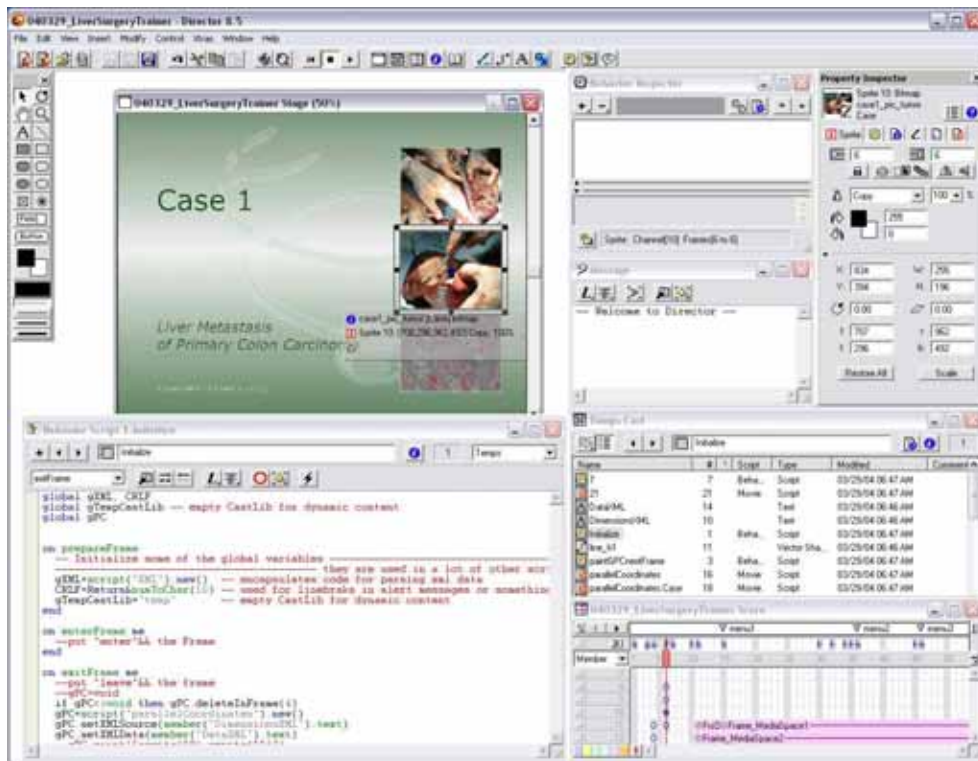
Bei der Umsetzung eines Lernsystems in spezialisierten Autorensystemen können einfache Lernanwendungen bereits über Templates und Wizards ohne Programmieraufwand effizient umgesetzt werden (siehe zum Beispiel AUTHORWARE [Adobe 2007a] und TOOLBOOK [SumTotal 2007]). Zusätzliche Skriptfunktionalität ermöglicht darüber hinaus komplexe Steuerungs- und Programmabläufe, wobei meist jedoch von einem starren bildschirm- bzw. seitenbasierten Ablauf ausgegangen wird (vergleiche Abschnitt 2.6). Durch vordefinierte Funktionalitäten für *Drill & Practice* Aufgaben, Multiple Choice Fragen und die automatische Auswertung der Lernergebnisse stellen diese spezialisierten Entwicklungsumgebungen eine effiziente Ideallösung für Standard-CBT-Systeme dar.

Entwicklungsumgebungen für interaktive Multimedia-Anwendungen lassen mehr Freiräume im Programmablauf in der Gestaltung von Benutzeroberflächen und der Realisierung komplexer Benutzerinteraktion (siehe Flash [Adobe 2007b] und Macromedia Director [Adobe 2007c]).

Die Integration von medizinischen Bild-/Volumendaten wird dabei in allen Systemen nicht unterstützt. Die Integration interaktiver 3D-Szenen wird meistens nur bedingt über vorgerenderte *object movies* ermöglicht (vergleiche Abschnitt 9.3.3). Hingegen wird die Integration aller anderen Medien umfangreich unterstützt.

Um in diesen Entwicklungsumgebungen und den resultierenden Lernsystemen komplexe Interventionsplanung (z.B. Einzeichnen einer Resektionsebene) zu ermöglichen, müsste die gesamte Funktionalität dafür in der gewählten Entwicklungsumgebung re-implementiert werden.

Nützlich ist diese Variante der Implementierung vor allem in frühen Stadien der Entwicklung. Da Autorensysteme die Gestaltung der graphischen Benutzerschnittstelle (GUI) sowie die Strukturierung von Programmabläufen effizient unterstützen, können erste Konzepte und Prototypen in entsprechenden Systemen schneller umgesetzt und getestet werden. Der erste Prototyp des LIVER SURGERY TRAINER wurde zur Erprobung erster Konzepte diesen Beobachtungen folgend mit Macromedia Director [Adobe 2007c] (siehe Abbildung 10.5) umgesetzt [Bade et al. 2004a, b]. Hierdurch konnten vor allem das Fallpräsentations- und Interaktionskonzept sowie die Strukturierung des Lerninhaltes konzeptionell unterstützt und erprobt werden.



**Abbildung 10.5:** Entwicklung des LIVER SURGERY TRAINER unter Macromedia Director zur Erprobung der Fallpräsentations- und Interaktionskonzepte. [Mirschel 2004]

### Umsetzung im Interventionsplanungssystem

Bei der Umsetzung des Lernsystems im Interventionsplanungssystem steht die volle Funktionalität der computerunterstützten Interventionsplanung zur Verfügung. Entsprechend kann diese im Lernsystem angeboten und trainiert werden. Zusätzlich ist die Integration von medizinischen Bild-/Volumendaten und rekonstruierten 3D-Modellen durch das Interventionsplanungssystem gegeben.

Die Integration von Medien wie Videos, Bilder, Text und Ton ist jedoch meist problematisch. Darüber hinaus ist oft die Gestaltung und Umsetzung von Lerninhalten, der Hilfe und Navigation in diesen sowie von Trainingsaufgaben und deren Bewertung problematisch. Entsprechende Funktionalitäten müssen dann im Interventionsplanungssystem implementiert werden. Diese zusätzlichen Entwicklungen im Planungssystem können dabei aber auch zur Verbesserung der Interaktionsgestaltung, der Hilfe und des Anwenderfeedbacks im Planungssystem selbst genutzt werden. Analog profitiert auch das Lernsystem von Neu- und Weiterentwicklungen der computerunterstützten Planung, welche dann gleich zum Trainieren bereit stehen.

### Kombinierte Umsetzung in Autorensoftware und Interventionsplanungssystem

Eine Möglichkeit, die Fähigkeiten beider Entwicklungsumgebungen zu verbinden, besteht dabei darin, diese kombiniert über adäquate Schnittstellen einzusetzen. Dabei ist es für den Lernkontext nicht ausreichend, wenn sich Bildschirmseiten der Lernsoftware mit Anzeigen der Interventionsplanung abwechseln. Vielmehr müssen Lernmedien, Trainingskonzepte und Lernhilfen über entsprechende Schnittstellen nahtlos in die Planungskomponenten und umgekehrt integriert werden können. Zusätzlich muss dafür gesorgt werden, dass die Lernsys-

temkomponente Aktionen in der Planungskomponente steuern kann und über Aktionen in der Planungskomponente informiert wird. Beides ist dabei zum Beispiel für die Präsentation, geführte Tour und automatische Hilfe (siehe Abschnitt 10.3.4) erforderlich.

Bei der Entwicklung des LIVERSURGERYTRAINER konnte die kombinierte Umsetzung geeignet angewendet werden, um integriert in dem mit Autorenssoftware entwickelten ersten Prototypen die Trainingskomponenten für die Resektionsplanung und Ablationsplanung prototypisch entwickeln und erproben zu können [Bade et al. 2006b].

Problematisch beim kombinierten Ansatz ist neben der zu realisierenden Schnittstelle auch die Handhabung bei der Entwicklung. So sind für die verschiedenen Komponenten unterschiedliche Programmierumgebungen notwendig. Problematisch ist dies vor allem bei der Detektion und Behebung von Programmfehlern, da sich diese in jeder der Komponenten, in der Schnittstelle und in der Kombination aller ergeben können und dies von gängigen Debug-Systemen nicht unterstützt wird.

Bei der Beteiligung komplexer Interventionsplanungssysteme sind Variante 2 und 3 zu bevorzugen, um die komplexe Planungsfunktionalität unmittelbar im Lernsystem nutzen zu können. Variante 2 und 3 machen damit aber das Lernsystem stark von einer existierenden Software und deren Konzept zur Interventionsplanung abhängig, wodurch die Allgemeingültigkeit der Lerninhalte beeinträchtigt werden kann.

Die Wahl der idealen Variante für ein spezielles System hängt dabei von Kosten-Nutzen-Rechnungen ab, welche die in diesem Abschnitt erwähnten Vor- und Nachteile der vorgestellten Implementierungsvarianten berücksichtigen sollten. Hierbei kann es im Entwicklungsprozess durchaus sinnvoll sein, auf unterschiedlichen Entwicklungsstufen unterschiedliche Implementierungsvarianten zu verwenden.

## 10.5 Interaktions- und Visualisierungsaspekte des LIVERSURGERYTRAINER

Wichtige Interaktions- und Visualisierungsaspekte des LIVERSURGERYTRAINER sind hier die Fallauswahl, die 3D-Visualisierung patientenspezifischer Daten, das Heranführen an komplexe 3D-Visualisierungen, das dynamische Visualisieren von Hervorhebungswechseln sowie die Interaktion in den 3D-Szenen. Im Folgenden werden diese Aspekte des LIVERSURGERYTRAINER näher beschrieben und daran die Anwendung der in den vorherigen Kapiteln behandelten Themen erläutert.

### 10.5.1 Fallauswahl

Eine Anforderung aus Abschnitt 10.2.2 ist die zielgerichtete Auswahl von zu bearbeitenden Fällen bzw. Problemen. In Lernsystemen werden zur Fallauswahl üblicherweise entweder einfache Listen mit Fallnamen ohne Bezug zum Inhalt (z.B.: Fall 1, Fall 2, ..., Fall  $n$ ) oder erweiterte Listen mit Angaben zu bestimmten Eigenschaften der Fälle angeboten. Diese Präsentationsformen sind jedoch für eine zielgerichtete Auswahl ungeeignet, da sie die Charakteristik der einzelnen Fälle im Gesamtkollektiv nicht ausreichend erkennbar repräsentieren können. Darüber hinaus ist aus einfachen Listen oder auch Tabellen die Gesamtverteilung der zur Verfügung stehenden Fälle nicht ersichtlich.

Eine Fallauswahl zur zielgerichteten Wahl von zu trainierenden Fällen in Lernsystemen muss jedoch:

- Überblick über die Falldatenbank und die enthaltenen Fälle vermitteln,
- Auswahl von Fällen anhand ihrer Eigenschaften ermöglichen sowie
- Fälle vergleichbar präsentieren.

Im Folgenden werden zunächst Fallkriterien für die Fallauswahl im LIVER SURGERY TRAINER erarbeitet. Darauf aufbauend werden erweiterte Visualisierungsmethoden zur Realisierung der Fallauswahl betrachtet.

### Kriterien für die Fallauswahl

Wichtige Kriterien für die Auswahl von Fällen in Lernsystemen stellen sowohl *domänenspezifische*, als auch *lernsystemspezifische Metadaten* dar. Entsprechend dieser Daten werden Fälle von Lernenden ausgewählt.

*Domänenspezifische Metadaten* beziehen sich dabei auf Eigenschaften der Falldaten, die sich direkt aus dem Lerninhalt bzw. Anwendungsgebiet des Lernsystems ergeben. Beispiele sind unter anderem die Art der Erkrankung, deren Lokalisation und die Form der Therapie. Im LIVER SURGERY TRAINER können Lernende nach diesen Auswahlkriterien zum Beispiel ein Leberhämangiom (Art der Erkrankung) mit durchzuführender Resektion (Form der Therapie) zum Trainieren auswählen.

*Lernsystemspezifische Kriterien* zur Fallauswahl stellen zum Beispiel der Schwierigkeitsgrad und der Bearbeitungsstatus dar. Über diese können Lernende ihren Fähigkeiten entsprechende Fälle auf unterschiedlichen Herausforderungsstufen auswählen bzw. verfolgen welche Fälle im Lernsystem sie schon bearbeitet haben. Unter Berücksichtigung der stark begrenzten Lernzeit, die der Zielgruppe des LIVER SURGERY TRAINER zur Verfügung steht, wird als lernsystemspezifisches Kriterium zusätzlich die zu erwartende Bearbeitungsdauer für die Fälle angegebenen. Über diese können Lernende dem aktuellen Zeitfenster angepasst Fälle auswählen und trainieren.

Im LIVER SURGERY TRAINER werden zusätzlich Falleigenschaften hinzugenommen, welche die mentale Einordnung (Erinnerung) eines Falls unterstützen bzw. die Kommunikation zwischen Lernenden über bearbeitete Fälle erleichtern. Beispiele für entsprechende Metadaten zur Fallidentifikation sind unter anderem Alter und Geschlecht sowie fiktive Namen und Passbilder der virtuellen Patienten. Diese ermöglichen der Zielgruppe die schablonengerechte mentale Einordnung von Lernfällen ähnlich realen Patienten. Dadurch ergibt sich zusätzlich die Möglichkeit der gewohnten Vorstellung von „Patienten“ in Seminaren und Lerngruppen.

Aus allen drei Kategorien ergibt sich damit eine umfangreiche Anzahl an Kriterien für die anwendungsorientierte und zielgerichtete Fallauswahl.

### Typische Anfrageszenarien bei der Fallauswahl

Die Auswahl von Trainingsfällen erfolgt zielgerichtet durch die Lernenden anhand spezifischer Eigenschaften des gesuchten Falles. Die Suche erfolgt z.B. nach einem bestimmten Krankheitsbild, einem noch nicht bearbeiteten Fall, einem Fall der dem zur Verfügung stehenden Zeitfenster entspricht oder einem Fall der möglichst ähnlich einem realen Fall ist.

Grundlegend ergeben sich bei der Fallauswahl die Anfrageszenarien:

- *Einfaches Finden und Auswählen* von Fällen mit 2-3 bekannten Eigenschaften,

- *Komplexes Finden und Auswählen* von Fällen anhand mehrerer Parameter und Relationen zu anderen Fällen in der Datenbank sowie
- *Exploration der Falldatenbank* für ein tieferes Verständnis der verfügbaren Fälle.

Ein Beispiel für *einfaches Finden und Auswählen* ist die Suche nach einem Fall mit Hepatozellulärem Karzinom (HCC), bei dem eine Resektion geplant werden soll oder bei dem eine Resektion des linken Leberlappens zu planen ist.

*Komplexes Finden und Auswählen* ergibt sich, wenn sehr viele Eigenschaften der Fälle zu berücksichtigen sind und/oder Fälle miteinander verglichen werden müssen. Nötig ist dies zum Beispiel, wenn ein Fall gesucht wird, der einem realen Fall möglichst ähnlich ist.

Die *Exploration der Falldatenbank* erfolgt, wenn Lernende sich einen Überblick über die in einem Lernsystem vorhandenen Fälle verschaffen wollen oder seltene Fälle herausuchen und trainieren möchten. Auch durch Lehrende und Systemautoren muss die Falldatenbank exploriert werden, um zum Beispiel Bereiche im Fallspektrum zu identifizieren, die noch durch Fälle ergänzt werden sollten oder um unterschiedliche Fälle für ein Trainingsseminar oder eine Prüfung auswählen zu können.

Geeignete Konzepte für die Fallauswahl sollten dabei alle drei Anfrageszenarien unterstützen.

### Typischer Fallumfang

Die Fallanzahl in einigen Lernsystemen ist oft mit 10 bis 20 Fällen eher gering. Umfangreiche Falldatenbanken umfassen durchaus mehrere hundert Fälle. Entsprechend große Fallzahlen stehen zum Beispiel in einigen CASUS- und CAMPUS-Anwendungen zur Verfügung (siehe Abschnitt 2.6). Viel größere Fallzahlen sind jedoch nicht zu erwarten.

Jedoch können bereits bei 10 bis 20 Fällen einfache Listen mit Fallnamen oder auch mit Freitextumschreibungen der Fälle nicht mehr effizient alle Konzepte zur Fallauswahl unterstützen.

### Erweiterte Techniken zur Fallauswahl

Um eine zielgerichtete und effiziente Fallauswahl im LIVER SURGERY TRAINER zu ermöglichen, wird auf erweiterte interaktive Visualisierungstechniken zurückgegriffen, welche die Exploration und gezielte Suche in multivariaten Daten ermöglichen.

Im LIVER SURGERY TRAINER-Prototypen wurden zur Unterstützung komplexer Auswahlzenarien zunächst *Parallele Koordinaten* zur Fallauswahl realisiert [Bade et al. 2004b]. *Parallele Koordinaten* sind aus der Informationsvisualisierung bekannt und bilden multivariate Daten auf Streckenzüge ab [Inselberg 1985; Hauser et al. 2002]. Diese Streckenzüge werden zwischen parallel angeordneten Achsen abgetragen, welche jeweils den Wertebereich einer Variablen in den Daten aufspannen. Der Verlauf der Streckenzüge gibt dann Aufschluss über einen Fall und seine Eigenschaften. Abbildung 10.6 zeigt eine entsprechende Darstellung der Fälle für den LIVER SURGERY TRAINER.

In den *Parallelen Koordinaten* können Fälle über die Selektion ihres Streckenzuges oder auch Fallgruppen durch Selektion mehrerer Streckenzüge ausgewählt werden. Durch Selektion von Wertebereichen auf den Achsen werden Fälle, die in diesem Bereich liegen, hervorgehoben. Dadurch können die Fälle in der Datenbank interaktiv gefiltert werden, wodurch die zielgerichtete Fallauswahl erleichtert wird. Dabei sind die Ausprägungen aller Fälle in der



**Abbildung 10.6:** Fallauswahl über *Parallele Koordinaten* für die zielgerichtete Informationssuche

Datenbank erkennbar. Zusätzlich können Fälle sehr einfach untereinander verglichen werden, um zum Beispiel weitere ähnliche Fälle auszuwählen.

Als Alternative zu normalen Tabellen werden in einer aktuellen Diplomarbeit zusätzlich *InfoZoom* [Spence und Beilken 1999; Spence 2001] und *Table Lens* [Rao und Card 1994] als weitere potentiell geeignete Techniken aus der Informationsvisualisierung für die Fallauswahl zusammengetragen und analysiert [Mewes 2007]. Hierbei wird dargelegt, dass die grafische Darstellung der Daten kombiniert mit interaktiven Filtermöglichkeiten, wie von *InfoZoom* und *Parallelen Koordinaten* bereitgestellt, am besten für die zuvor beschriebenen typischen Anfrageszenarien geeignet sein sollten.

## Evaluation der Fallauswahl

Um Nutzerakzeptanz sowie einen möglichen Komfort- und Effizienzgewinn im Vergleich zu vertrauten Tabellendarstellungen zu überprüfen, wurde eine empirische Evaluation der erweiterten Techniken zur Fallauswahl durchgeführt. Hierzu werden in [Mewes 2007] Tabellen, *Table Lens*, *InfoZoom* und *Parallele Koordinaten* (siehe Abb. 10.7) zur Bearbeitung von fünf Aufgaben zur Fallauswahl evaluiert. Hierbei wurden drei Aufgaben zum *Einfachen Finden und Auswählen* sowie zwei zum *Komplexen Finden und Auswählen* durch 42 Probanden (5 Ärzte, 20 Informatiker, 17 weder Ärzte noch Informatiker) bearbeitet.

Die Studie konnte dabei eine Überlegenheit der *Parallelen Koordinaten* und der *InfoZoom*-Technik gegenüber einer Tabellenansicht, vor allem bei Aufgaben zum *Komplexen Finden und Auswählen* bestätigen [Mewes 2007]. Hierbei wurde deutlich, dass durch diese Präsentationsformen eine Fallauswahl realisiert werden kann, welche die zielgerichtete Auswahl von Fällen anhand ihrer Eigenschaften ermöglicht und den Vergleich von Fällen sowie die Einordnung eines Falles in Relation zum gesamten Fallbestand zulässt. Hierdurch werden die speziellen Anforderungen an die Fallauswahl erfüllt und es wird eine zielgerichtete Informationssuche, wie in Abschnitt 10.2.2 gefordert, ermöglicht.

Da die Fallauswahl mittels *Paralleler Koordinaten* und *InfoZoom* jedoch für die Zielgruppe unbekannt und ungewohnt ist, könnte der erforderliche Einarbeitungsaufwand die Vorteile



CaseID	Age	BMI	Difficulty	Height	First Name	Last Name	Metastasis	Therapy	Type	Weight	Operability
MD-88632078	-	-	3 (high)	-	-	-	-	-	LDLT	-	-
MD-83594017	63	22	3 (high)	193	Sxxx	Gxxx	4	Ablation(Kryo)	Onco	84	no
MD-85559331	66	36	3 (high)	176	Lxxx	Axxx	0	Ablation(MW)	Onco	114	no
MD-24390960	51	16	3 (high)	188	Uxxx	Oxxx	1	Ablation(MW)	Onco	58	no
MD-63389521	18	18	3 (high)	187	Dxxx	Txxx	0	Ablation(RF)	Onco	64	no
MD-13068478	32	25	3 (high)	195	Jxxx	Hxxx	1	Resection	Onco	96	yes
MD-98119950	40	35	3 (high)	179	Oxxx	Lxxx	3	Resection	Onco	114	yes
MD-46133695	12	29	3 (high)	164	Sxxx	Rxxx	2	Resection	Onco	78	yes
MD-4277700	39	32	3 (high)	198	Zxxx	Rxxx	3	Resection	Onco	129	yes
MD-71779504	-	-	2 (medium)	-	-	-	-	-	LDLT	-	-
MD-29602980	-	-	2 (medium)	-	-	-	-	-	LDLT	-	-
MD-13144624	-	-	2 (medium)	-	-	-	-	-	LDLT	-	-

(a)

Ansicht zurücksetzen

	BMI	Diffic	Disse	First	Graf	Heig	Last	MVH	Age	Metastasis	Operability	Posit	Posit	Sex	Stat	The	Type	Vess	Wei
MD-80044048	39	1	...	W...	-	-	Sx...	-	41	2	no	V...	ce...	Fe...	3	A...	O...	-	-
MD-52762346	41	2	...	Tx...	-	-	Px...	-	51	1	yes	V...	ce...	Male	5	R...	O...	-	-

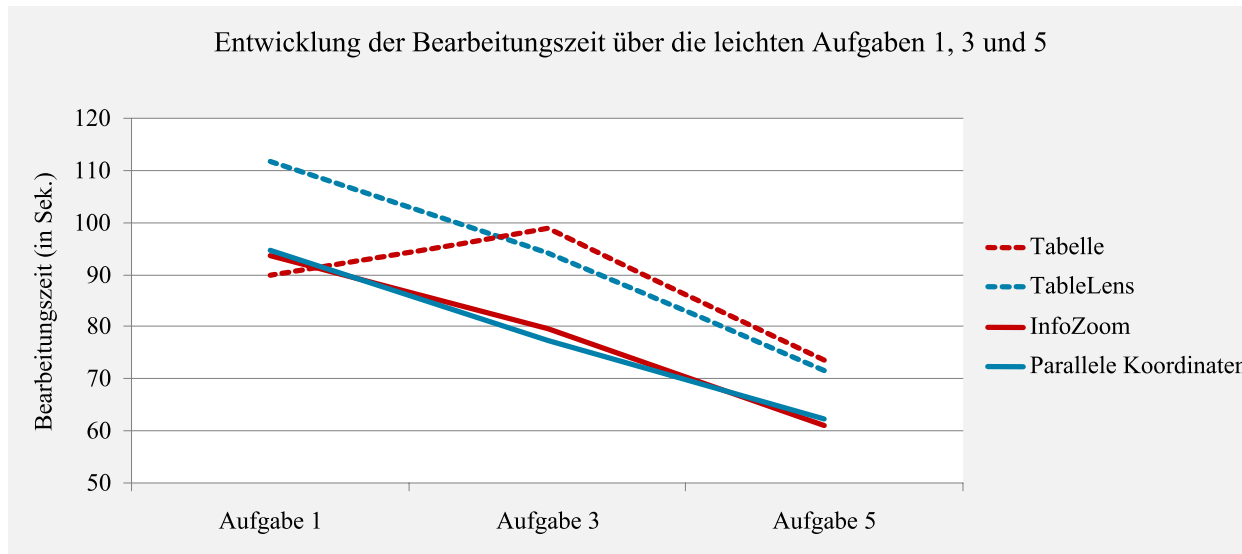
(b)

<- Zoom(Zurück) Zoom (Vor)-> Ansicht zurücksetzen

Age	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BMI	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Difficulty	...	...	...	1	...	3 (	2	...	1 (L	...	...	3	...	...	1	...	3	2	...
Dissection	yes	no	...	yes	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
First Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Graft Segments	-	-	-	-	S...	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Height	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Last Name	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MVH remains	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Metastasis	-	-	-	-	-	1	3	2	3	4	0	1	4	1	0	4	1	2	0
Operability	-	-	-	-	-	no	...	...	...	...	...	...	...	...	...	...	...	...	...
Position(Liver)	-	-	-	-	-	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Position(Tumor)	-	-	-	-	-	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Sex	-	-	-	-	-	F...	M...	Fe...	Fe...	M...	...	Male	Female	Female	...	...	...	...	...
Therapy	-	-	-	-	-	Ablation(Kryo)	Ablation(M...	Ablation(RF)	...	...	...	...	...	...	...	...	...	...	...
Type	-	-	-	-	-	LDLT	...	...	...	...	...	...	...	...	...	...	...	...	...
Vessel abnormalities	-	-	-	-	-	no	...	...	...	...	...	...	...	...	...	...	...	...	...
Weight	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

(c)

**Abbildung 10.7:** Screenshots der evaluierten Techniken zur Fallauswahl: (a) einfache Tabelle, (b) *Table Lens* und (c) *InfoZoom*. Zusätzlich wurden *Parallele Koordinaten*, wie in Abbildung 10.6 dargestellt, evaluiert. (modifiziert aus [Mewes 2007])



**Abbildung 10.8:** Verlauf der mittleren Bearbeitungszeit mit jeder der untersuchten Visualisierungstechnik für drei zeitlich versetzt dargebotene Aufgaben zum *einfachen Finden und Auswählen*. Die Bearbeitungszeiten für die parallelen Koordinaten und InfoZoom sinken dabei rasch unter die Bearbeitungszeiten der Tabelle. [Mewes 2007]

dieser Techniken gegenüber einfacher tabellarischer Auflistungen konterkarieren. Um dies zu überprüfen, wurden in der Evaluationsstudie drei einfache Selektionsaufgaben (*Einfaches Finden und Auswählen*) zeitlich versetzt dargeboten, so dass mit steigender Anzahl bearbeiteter Aufgaben der Lerneffekt evaluiert werden kann. Während bei der ersten einfachen Selektionsaufgabe die mittlere Bearbeitungszeit mit den *Parallelen Koordinaten* und *InfoZoom* noch knapp über der Fallauswahl in Tabellenform lag, sanken die Bearbeitungszeiten mit den erweiterten Techniken bei den nachfolgenden Aufgaben unter die der Tabellen (siehe Abbildung 10.8). Diese Ergebnisse bestätigen einen nötigen Einarbeitungsaufwand, der jedoch sehr gering ausfällt und bereits nach kurzer Anwendung verschwindet. Dies legt nahe, dass die Vorteile der erweiterten Techniken (*Parallele Koordinaten* und *InfoZoom*) zur Fallauswahl effizient genutzt werden können.

Die Evaluationsergebnisse weisen weiterhin auf eine bessere Auswahlgeschwindigkeit und -korrektheit mit den *Parallelen Koordinaten* und der *InfoZoom*-Technik hin. Diese besondere Eignung lässt sich, aus den gesammelten Erfahrungen während der Evaluationsstudie, auf Möglichkeiten zur Filterung der angezeigten Fälle in beiden Techniken zurückführen.

Zusammenfassend wird über den Ansatz *Paralleler Koordinaten* im LIVER SURGERY TRAINER sowie über die *InfoZoom*-Technik die zielgerichtete Fallauswahl bzw. allgemeiner die Informationssuche effizient ermöglicht. Die *InfoZoom*-Technik schneidet dabei zusätzlich in der subjektiven Beurteilung durch die Probanden am besten ab. Durch die geringe Anzahl an potentiellen Anwendern des Lernsystems in der Studie (5 Ärzte) stehen jedoch konkrete Aussagen zur besseren Eignung der erweiterten Techniken noch aus.

### 10.5.2 3D-Visualisierung patientenspezifischer Daten

Parallel zur Planungssoftware von Leberinterventionen werden die radiologischen Bilddaten über Oberflächennetze dargestellt. Diese werden mit den in dieser Arbeit beschriebenen Verfahren zur Reduktion von Treppenstufen- und Blockartefakten behandelt (Kap. 6–7).

Zur Erzeugung von standardisierten Ansichten, zum Beispiel für die Falleinführung oder Inspektion von Gefäßästen, wird der vorgestellte Ansatz für adaptive dynamische Visualisierungen auch für statische Ansichten eingesetzt (vergleiche Abschnitt 9.5.1). Über den gleichen Ansatz werden zusätzlich standardisierte Animationssequenzen zum Heranführen an komplexe 3D-Visualisierungen und zur dynamischen Visualisierung von Hervorhebungswechseln erzeugt.

**Aufbereitung der Oberflächenmodelle:** In Kapitel 4 und 5 wurden allgemeine Strategien und Glättungsverfahren zur Reduktion von Artefakten in medizinischen Oberflächenmodellen besprochen und evaluiert. Den Evaluierungsergebnissen aus Kapitel 5 folgend, eignen sich im LIVER SURGERY TRAINER Taubin's  $\lambda$ - $\mu$ - und die Laplace+ $\mathcal{HC}$ -Glättung zur Reduktion von Block- und Terrassenartefakten in den Oberflächenmodellen der Leber. Entsprechend kompakte und sehr fein aufgelöste Modelle können über diese Verfahren bei sehr guter Genauigkeit geglättet werden (siehe Abschnitt 5.7).

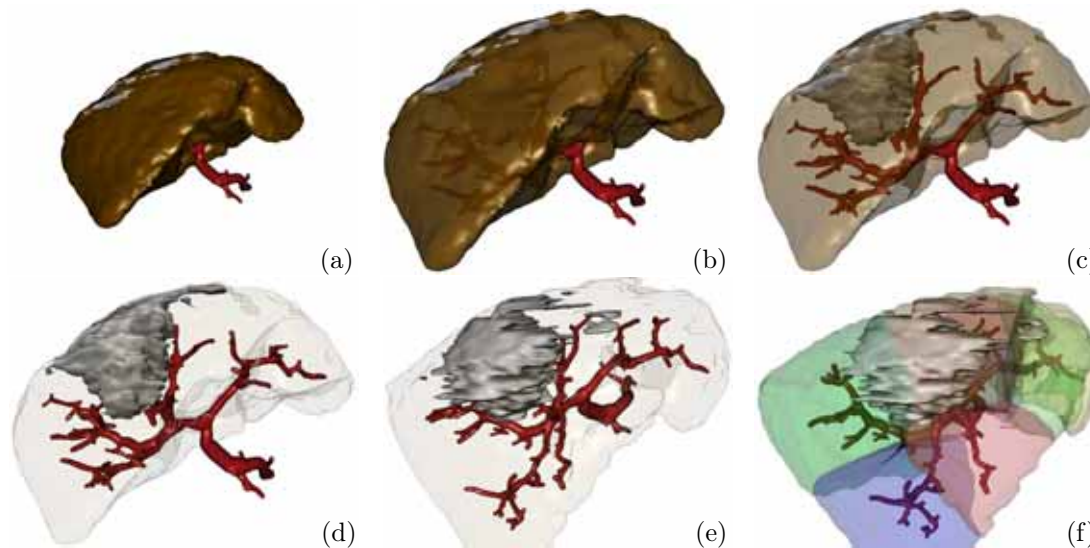
Tumore stellen ähnlich der Leber kompakte Objekte dar, welche auch gut mit Taubin's  $\lambda$ - $\mu$ - und der Laplace+ $\mathcal{HC}$ -Glättung behandelt werden können. Diese Verfahren garantieren aber keine exakte Repräsentation der segmentierten Volumendaten, so dass sich in der Interventionsplanung Verfahren zur voxelbegrenzten Oberflächennetzglättung für Tumore besser eignen (*Diamond Constraint* Glättung, siehe Abschnitt 6.2.4). In der Umgebung eines Lernsystems ist dieser zusätzliche Grad an Exaktheit nicht unbedingt erforderlich. Jedoch darf die Behandlung mit einem anderen Glättungsverfahren die Tumorform nicht therapierelevant verändern.

Zur Darstellung von Blutgefäßen im LIVER SURGERY TRAINER eignen sich die in Kapitel 7 vorgestellten MPUIs sowie modellbasierte Darstellungsverfahren (z.B. *Convolution Surfaces*, siehe Abschnitt 7.2.2). Da im Umfeld der Interventionsplanung in der Leberchirurgie die exakte Blutgefäßform, wie sie durch MPUIs dargestellt werden kann, nicht zwingend relevant ist und Lage sowie Verzweigungsverhalten der Gefäße im Mittelpunkt der Planung stehen, können auch *Convolution Surfaces* geeignet eingesetzt werden.

Für die nachfolgenden Bilder, aus einer im Rahmen der vorliegenden Arbeit entwickelten Umsetzung des LIVER SURGERY TRAINER, wurden für die Blutgefäße *Convolution Surfaces* und für Tumore und Leberoberfläche Taubin's  $\lambda$ - $\mu$ -Glättung verwendet.

**Heranführen an komplexe 3D-Visualisierungen:** Um leicht verständliche 3D-Visualisierungen auch für Neulinge der computerunterstützten Interventionsplanung zu erzeugen, werden komplexe Ansichten über dynamische Visualisierungen eingeführt (Abschnitt 9.1). So wird in Abbildung 10.9 von einer frontalen Ansicht auf die opake Leber ausgegangen (a). Über nachvollziehbare Änderungen der Transparenz (a-d), der Sichtrichtung (d-f) und dem Einblenden der Versorgungsgebiete (f) wird diese neue Visualisierung erklärend eingeführt. Hierbei können Anwender den Änderungen der Szene durch einfaches Beobachten folgen, während einfaches Umschalten von Abbildung 10.9 (a) zu (f) dazu führt, dass Anwender die neue Ansicht aufwändig interpretieren und mental in Relation zur vorherigen setzen müssen. Vor allem dann, wenn die aktuelle Szene noch unbekannt ist, kann dies zu Interpretationsproblemen führen.

**Dynamische Visualisierungen von Hervorhebungswechseln:** Ergänzend zum Heranführen an 3D-Visualisierungen werden dynamische Visualisierungen eingesetzt, um die Änderungen in der 3D-Darstellung beim Wechsel des Fokus von einem auf ein anderes Ob-



**Abbildung 10.9:** Dynamisches Heranführen aus vertrauter Ansicht auf die Leber (a) zur Ansicht der Versorgungsgebiete um den Tumor (f).

jekt nachvollziehbar zu machen (Abschnitt 9.1). Abbildung 10.10 zeigt Ausschnitte eines entsprechend visualisierten Hervorhebungswechsels bei der sukzessiven Untersuchung zweier Gefäßäste in der direkten Nachbarschaft des Tumors.

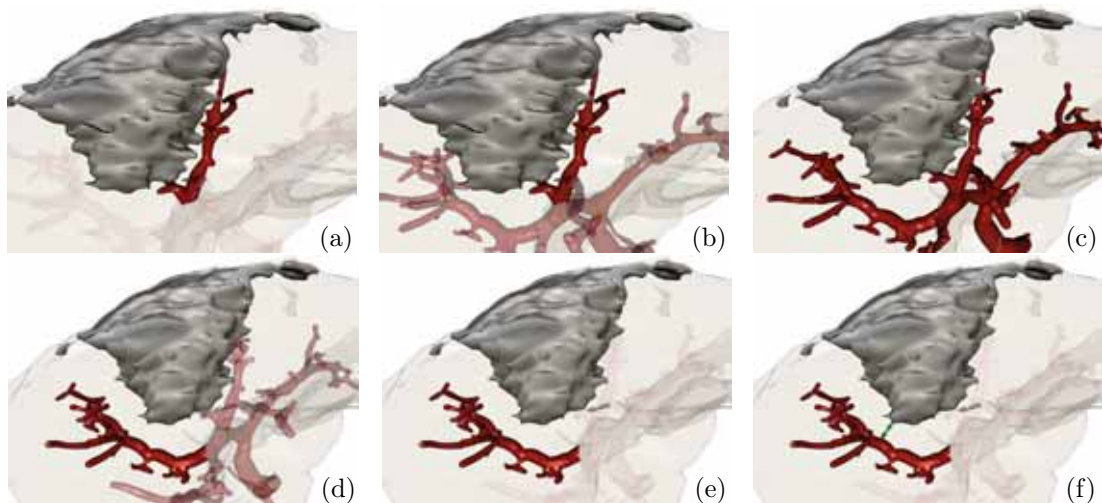
### 10.5.3 Interaktion in 3D-Szenen

Zur Unterstützung komplexer Interaktionen in den 3D-Visualisierungen werden standardisierte Interaktionsaufgaben über das skriptbasierte Konzept für adaptive dynamische Visualisierungen automatisiert (Abschnitt 9.4). Abbildung 10.10 sowie Abbildung 9.10 auf Seite 146 stellen beispielhaft Ausschnitte aus dem automatisierten Heranfliegen an einen Gefäßast in Tumornähe da. Hierbei werden gleichzeitig sowohl komplexe Navigationsaufgaben als auch Interaktionen zur Einstellung geeigneter Visualisierungsparameter übernommen. Dies erleichtert die nötigen Interaktionen.

Für die freie Navigation und Exploration der 3D-Szenen im LIVERSURGERYTRAINER wird auf die in Kapitel 8 behandelten Rotationstechniken eingegangen. Der Einsatz des Two-Axis Valuator Trackball (Two-Axis VT) zur mausbasierten 3D-Rotation bietet sich hier an. Die mausbasierte Interaktion ist im Umfeld des LIVERSURGERYTRAINER wichtig, da davon auszugehen ist, dass 3D-Eingabegeräte in den meisten Lernsituationen nicht verfügbar sind (Abschnitt 10.2.2).

## 10.6 Zusammenfassung

Der LIVERSURGERYTRAINER soll das problemorientierte Erlernen und Trainieren computerunterstützter Interventionsplanung in der Leberchirurgie ermöglichen. Unter Berücksichtigung zielgruppenspezifischer, domänenspezifischer und lerntheoretischer Anforderungen wurde ein Lernsystemkonzept entwickelt. Besondere Aspekte des Konzeptes sind die Fallbasis aus klinischen Daten, ein mehrstufiges Hilfesystem sowie eine verstärkte Unterstützung bei der Interaktion und Navigation in 3D-Visualisierungen.



**Abbildung 10.10:** Hervorhebungswechsel von einem selektierten Gefäßast (a) zu einem anderen Gefäßast (f). Hierbei werden der Darstellungsstil der beteiligten Gefäße sowie die aktuelle Ansicht auf die Szene dynamisch geändert.

Die gezielte Informationssuche im Lernsystem wird über die Fallauswahl mit parallelen Koordinaten sowie über die Unterteilung der Fallbearbeitung in logische Abschnitte unterstützt.

Zur Unterstützung des Lernprozesses bietet das Hilfesystem verschiedene Modi, welche mit unterschiedlicher Granularität die verschiedenen Lernstufen: Neuling (*Präsentation* des Falls und der Interventionsplanung), bis hin zu Gewandtheit und Expertentum (Hilfe auf Anfrage) unterstützen.

Die grafische Darstellung der Oberflächenmodelle aus klinischen Daten wird im Lernsystem ohne Block- und Terrassenartefakte über die in Kapitel 6 und 7 vorgestellten Methoden zur Extraktion und Glättung von Oberflächenmodellen erreicht.

Standardisierte Ansichten und dynamische Visualisierungen patientenspezifischer Daten werden über ein skriptbasiertes Konzept erreicht. Dieses ermöglicht die Adaption von abstrakten Ansichts- und Animationsbeschreibungen an die patientenindividuellen Daten. Hierdurch wird der Erstellungsaufwand standardisierter Darstellungen automatisierbar und erfordert beim Einpflegen neuer Fälle in das System keinen zusätzlichen Aufwand. Darüber hinaus unterstützen die erreichten standardisierten Ansichten die Lernenden beim Vergleichen und Ziehen von Schlüssen zwischen unterschiedlichen Fällen. Ferner werden die Lernenden durch den Einsatz der adaptiven dynamischen Visualisierungen bei der 3D-Navigation unterstützt.





# Fazit und Ausblick

---

Die vorliegende Arbeit zielt darauf ab, Techniken der interaktiven und dynamischen Visualisierung für die chirurgische Ausbildung und Interventionsplanung zu entwickeln. Die chirurgische Ausbildung und Interventionsplanung stellen dabei besondere Anforderungen an die Genauigkeit und Plausibilität von Visualisierungen radiologischer Daten. Block- und Terrassenartefakte aus binär segmentierten Volumendaten gefährden dabei die Plausibilität und Akzeptanz generierter Oberflächendarstellungen anatomischer Strukturen. Die unkontrollierte Reduktion dieser Artefakte führt jedoch zu Darstellungen, welche keine verlässlichen Aussagen mehr über Lagebeziehungen, Form und Größe der anatomischen Strukturen erlauben.

Trotz der beschriebenen Probleme bergen dreidimensionale Visualisierungen ein großes Potential für die chirurgische Ausbildung und Interventionsplanung. Aktuelle Fortschritte in der computerunterstützten Interventionsplanung belegen diesen Trend. Dennoch zeigt sich, dass die Nutzung fortgeschrittener 3D-Visualisierungen in der Praxis derzeit hinter ihren Möglichkeiten bleibt. Eine Hypothese, der in dieser Arbeit Rechnung getragen wird, führt dies auf die schlechte Unterstützung bei der Navigation und Exploration der Visualisierungen zurück. Während beim traditionellen Betrachten von Schichtbildern ein zielgerichtetes und strukturiertes Vorgehen von einem Schichtbild zum nächsten möglich ist, führt das große Maß an Freiheitsgraden in interaktiven 3D-Visualisierungen zu einem Interaktionsoverhead. Die in dieser Arbeit entwickelten Verfahren und Konzepte nehmen sich dabei den speziellen Problemen an.

## 11.1 Beitrag zur Reduktion von Artefakten in medizinischen Oberflächenmodellen

Das visuelle System des Menschen ist sehr empfindlich gegenüber Diskontinuitäten in Oberflächen, da diese normalerweise wichtige Merkmale bei der Objekterkennung und Objektwahrnehmung darstellen. Zusätzlich werden bei merkmalsbetonenden bzw. illustrativen Darstellungstechniken (Silhouetten, Schraffur, ...) z.B. zur Darstellung von Kontextinformationen in medizinischen Visualisierungen diese Artefakte unvermeidlich betont. Dementsprechend können Artefakte bzw. Diskontinuitäten, wie sie durch die Extraktion aus binären Segmentierungsmasken entstehen, unnötig die visuelle Aufmerksamkeit binden und die Objektwahrnehmung übermäßig behindern.

Im Umfeld der chirurgischen Ausbildung und Interventionsplanung sind jedoch Objektdarstellungen wünschenswert, welche so einfach, effizient und genau wie möglich von Betrachtern erkannt und beurteilt werden können. Darüber hinaus werden auch für die Simulation von physikalischen oder physiologischen Prozessen (z.B. der Strömung in Blutgefäßen) artefaktfreie Objektrepräsentationen benötigt. Artefakte und Diskontinuitäten in den Modellen können sonst die Simulation verfälschen, oder gar unmöglich machen.

Mögliche Strategien zur Artefaktreduktion sind dabei grundlegend den drei Gruppen: Glättung der Segmentierungsmaske, Artefaktreduktion bei der Oberflächenextraktion sowie Glättung extrahierter Oberflächen zuzuordnen (Kap. 4). Die empirische Evaluation von Verfahren zur Glättung von Oberflächenmodellen (Kap. 5) zeigt auf, dass Artefakte durch die

Extraktion aus binären Segmentierungsmasken in kompakten und gut aufgelösten Oberflächen mit den untersuchten Verfahren effizient reduziert werden können. Zusätzlich konnte gezeigt werden, dass schlecht aufgelöste und elongierte Objekte, wie Blutgefäße, nicht adäquat behandelt werden können. Entsprechend werden Glättungsempfehlungen zur Glättung von medizinischen Oberflächenmodellen unterschiedlicher Objektklassen gegeben (Abschnitt 5.7).

Um eine korrekte Innen-Außen Repräsentation der ursprünglichen Voxeldaten auch bei der Glättung der Oberflächenmodelle zu erhalten, wurde das *Diamond Constraint* entwickelt (Kap. 6). Damit kann eine korrekte Repräsentation für *Marching Cubes* (MC) extrahierte Oberflächenmodelle auch nach der Glättung garantiert werden. Für eine optimale Repräsentation von Blutgefäßen wurde ein auf die spezielle Problematik dünner nur ein Voxel breiter Strukturen angepasstes MPUIs Verfahren vorgestellt (Kap. 7).

Die Glättungsempfehlungen aus Abschnitt 5.7 können entsprechend auf die neuen Techniken erweitert werden, so dass allgemein für die Glättung von Oberflächenmodellen aus Volumendaten eine voxelbegrenzte Glättung (z.B. mit dem *Diamond Constraint*) empfohlen wird. Im Spezialfall von schmalen Gefäßstrukturen wird die Verwendung des MPUIs Ansatzes empfohlen. Beide Strategien ermöglichen dabei eine effiziente Reduktion der Artefakte bei gleichzeitig guter bis sehr guter Genauigkeit. Dabei kommt vor allem die Genauigkeit der Visualisierungen der Interventionsplanung zugute. Vorangetrieben wird die praktische Nutzung der untersuchten Glättungsverfahren sowie der neu entwickelten Verfahren dadurch, dass diese alle in MEVISLAB [MeVis Research 2007] implementiert wurden und nun zahlreichen Anwendungen in der medizinischen Visualisierung zugute kommen können.

## 11.2 Beitrag zur effizienten 3D-Navigation

Die Attraktivität und das Potential der Computerunterstützung in der Ausbildung und Interventionsplanung ist unter anderem darin begründet, dass durch die Computerunterstützung mit den dargestellten Inhalten interagiert werden kann. Dies macht das gefahrlose Erproben von Therapieentscheidungen sowie eine intensivere Auseinandersetzung mit dargestellten Informationsräumen möglich [Ritter 2005]. Gleichzeitig stehen die komplexen Navigations- und Interaktionsaufgaben der effizienten und strukturierten Nutzung von 3D-Visualisierungen entgegen.

Die wichtigste Interaktionstechnik in dreidimensionalen Visualisierungen für die chirurgische Ausbildung und Interventionsplanung ist die mausbasierte Rotation der 3D-Szenen. Hierbei konnten unter anderem die Transitivität und Kommutativität von erzeugten Rotationsbewegungen sowie die Kontinuität zwischen Eingabe und Systemreaktion als wichtige Anforderungen für effizient und komfortabel steuerbare Rotationstechniken identifiziert werden.

In Bezug auf diese Anforderungen wurden existierende Rotationstechniken analysiert. Hierbei konnte ein Überblick über zur Verfügung stehende Rotationstechniken und ihre Eigenschaften gegeben werden. Zusätzlich zur Analyse wurde die Eignung der verschiedenen Verfahren für die Exploration komplexer Objekte empirisch evaluiert. Hierbei konnten signifikante Unterschiede zwischen den Techniken festgestellt werden, welche die identifizierte Anforderungsliste bestätigen. Zusätzlich wird durch die festgestellten statistisch relevanten Unterschiede belegt, dass die Art und Weise mit der komplexe dreidimensionale Visualisierungen interaktiv erkundet werden können, ein wichtiger Faktor für die Effizienz und Akzeptanz entsprechender Visualisierungsanwendungen ist.



Basierend auf der detaillierten Analyse der Interaktionstechniken und ihrer Schwachstellen wird für Teilaspekte der 3D-Navigation eine animierte Führung vorgeschlagen, welche als Interaktionsaspekt in ein Konzept zur interaktiven und dynamischen Visualisierung übernommen wird.

Zusammenfassend rechtfertigen die gewonnenen Erkenntnisse einen entsprechend hohen Aufwand zur Wahl geeigneter Rotationstechniken in der chirurgischen Ausbildung und Interventionsplanung sowie in 3D-Visualisierungsanwendungen über dieses Gebiet hinaus.

## 11.3 Beitrag zur interaktiven dynamischen Visualisierung

Dynamische Visualisierungen finden bei der (computerunterstützten) Ausbildung als Lehrvideos und bei der Interventionsplanung unter anderem zur Präsentation der Patientendaten in interdisziplinären Tumorboard-Besprechungen Anwendung.

Ein Beitrag dieser Arbeit liegt darin, Anwendungsgebiete bzw. Aufgaben der dynamischen Visualisierung im Kontext der chirurgischen Ausbildung und Interventionsplanung aufzuzeigen. Das größte Problem dynamischer Visualisierungen im medizinischen Kontext besteht dabei in der aufwändigen Erzeugung dieser für patientenindividuelle Volumendaten. Zur Vereinfachung wird deshalb derzeit in der chirurgischen Ausbildung auf vorberechnete Animationen eines meist idealisierten Datensatzes zurückgegriffen. In der Interventionsplanung werden keine oder nur simple Animationen (z.B. einfache Drehungen an verschiedenen Achsen um die Szene) verwendet.

In dieser Arbeit wurde ein skriptbasiertes Konzept zur adaptiven dynamischen Visualisierung von Patientendaten vorgestellt. Dieses ermöglicht die Definition von abstrakten Animationsbeschreibungen, welche bei der Wiedergabe der Animationen an die aktuelle Patienten-anatomie angepasst werden. Dadurch wird es möglich, effizient standardisierte dynamische Visualisierungen aus unterschiedlichen Patientendatensätzen zu erzeugen. Zusätzlich können über das Konzept standardisierte statische Ansichten erzeugt werden. Beides kann dabei zur Unterstützung der standardisierten Befundung, Interventionsplanung und Dokumentation im klinischen Umfeld beitragen.

Wie Kapitel 8 darstellt, ist die Art und Weise mit der komplexe dreidimensionale Visualisierungen interaktiv erkundet werden können, ein wichtiger Faktor für die Effizienz und Akzeptanz entsprechender Visualisierungsanwendungen. Allgemeine Ziele einer Visualisierung sind dabei: 1. die explorative Analyse von weitgehend unbekannten Daten, 2. die konfirmative Analyse von inhaltlich grob bekannten Daten sowie 3. die Präsentation von ausführlich bekannten Daten und Ergebnissen. Nur Visualisierungen mit dem Ziel der explorativen Analyse benötigen dabei umfassende Explorationsfreiheit. Die konfirmative Analyse sowie die Präsentation bekannter Daten kommen jedoch mit weniger Interaktion aus. In der chirurgischen Ausbildung und Interventionsplanung sind durch Diagnose und Bildanalyse die Daten genauer bekannt. Dies macht automatische und semi-automatische Verfahren zur Unterstützung der Navigation und Interaktion in den Visualisierungen möglich. Im Rahmen dieser Arbeit wird die Verwendung dynamischer Visualisierungen zur Unterstützung und Erleichterung der Interaktion und Navigation in 3D-Visualisierungen motiviert. Die Realisierung von unterstützenden dynamischen Visualisierungen wird dabei mit Hilfe des vorgestellten Ansatzes zur Deklaration von adaptiven Skripten für dynamische Visualisierungen erreicht.

Hierdurch wird ein Beitrag zur verbesserten Mensch-Computer-Interaktion in der klinischen Anwendung dreidimensionaler Visualisierungen geleistet.

Das Gesamtkonzept aus einführender dynamischer Visualisierung, interaktivem Rotieren und interaktionsunterstützender dynamischer Visualisierung kann hierbei als Analogon zum *Information Seeking Mantra*<sup>1</sup> [Shneiderman 1996] verstanden werden. Unter Berücksichtigung der Bildanalyse sind zusätzlich Parallelen zum neueren *Visual Analytics Mantra*<sup>2</sup> von Keim [2005] zu erkennen.

Als zusammenfassender Beitrag der Arbeit zur interaktiven und dynamischen Visualisierung für die chirurgische Ausbildung und Interventionsplanung wird hier das *Medical Interactive Visualization Mantra* abgeleitet:

- Image analysis first, show context (overview), zoom, rotate and details on demand guided by further analysis of the 3D-scene.

Wie das vorgestellte Konzept zur dynamischen Visualisierung erweitert werden muss, um eine weiterführende Analyse der 3D-Szene durchführen und nutzen zu können, stellt eine Herausforderung kommender Arbeiten dar.

## 11.4 Beitrag zur computerunterstützten chirurgischen Ausbildung

Zunehmende Fortschritte in der Interventionsplanung ermöglichen eine verbesserte Qualität von geplanten Interventionen. Zusätzlich können durch sorgfältige Planung Interventionen ermöglicht werden, welche anders nicht durchführbar wären. Die computerunterstützte Interventionsplanung muss jedoch erlernt und trainiert werden können. Ein Beitrag dieser Arbeit auf diesem Gebiet liegt in der Konzeption und prototypischen Umsetzung des LIVER SURGE-RY TRAINER für das Erlernen und Trainieren der Interventionsplanung in der Leberchirurgie. Über diese Arbeiten zum LIVER SURGERY TRAINER konnte eine BMBF-Förderung für die weitere Entwicklung des Lernsystems erreicht werden.

Die Integration dynamischer Visualisierungen, interaktiver und semi-automatischer Navigation sowie qualitativ hochwertiger Oberflächenmodelle aus Patientendaten kann hierbei als Template für die Entwicklung ähnlicher Lern- und Interventionsplanungssysteme verstanden werden. Das *Medical Interactive Visualization Mantra* kann dabei als grobe Richtlinie für Visualisierungen in der chirurgischen Ausbildung und Interventionsplanung dienen.

## 11.5 Ausblick

Diese Arbeit hat viele Aspekte und Fragestellungen der interaktiven und dynamischen Visualisierung für die chirurgische Ausbildung und Interventionsplanung behandelt, aber auch viele offen gelassen. Die sich ergebenden offenen und weiterführenden Fragestellungen in allen behandelten Themenbereichen werden nachfolgend dargelegt.

---

<sup>1</sup>„Overview first, zoom and filter, then details on demand.“ [Shneiderman 1996]

<sup>2</sup>„Analyse first, show important, zoom, filter and analyse further, details on demand“ [Keim 2005]

### 11.5.1 Extraktion von qualitativ hochwertigen Oberflächenmodellen

Um das allgemeine Problem von Artefakten durch binäre Segmentierung zu vermeiden, sollte die Forschung im Bereich der Bildanalyse und Segmentierung stärker auf die Erzeugung von (fuzzy) Wahrscheinlichkeitskarten für die Objektzugehörigkeit von Voxeln ausgerichtet werden.

Weiterführende Arbeiten auf dem Gebiet der Oberflächenextraktion können sich neben der Reduktion von Artefakten in extrahierten Oberflächenmodellen mit der gleichzeitigen Verbesserung der Dreiecksqualität befassen. Dabei wäre zu untersuchen, inwiefern existierende Verfahren zur Oberflächenextraktion mit gleichzeitiger Verbesserung der Dreiecksqualität auf binär segmentierte Daten angepasst werden können bzw. wie diese sinnvoll zu erweitern sind. Eine Herausforderung zukünftiger Extraktionsverfahren für die Simulation kann darüber hinaus die direkte Erzeugung von Volumengittern aus den segmentierten Volumendaten darstellen. Ein direkter Schritt aus den segmentierten Volumendaten zu einem Volumengitter für die Simulation, ohne Umweg über eine Oberflächenrepräsentation, könnte dabei Fehler und Probleme aktueller Zwischenschritte minimieren.

### 11.5.2 Interaktion und Navigation in 3D-Visualisierungen

Nach der Identifikation von Schwächen und Stärken unterschiedlicher Rotationstechniken können diese ihren Fähigkeiten entsprechend eingesetzt werden. Zukünftige Arbeiten können sich dabei mit der Spezialisierung der Techniken auf spezifische Anwendungsfelder befassen. Überaus interessant ist auch die Erforschung alternativer Möglichkeiten zur Exploration von 3D-Szenen, welche zum Beispiel von Analysen der 3D-Szene und abgeleiteten Vorschlägen für potentiell relevante Blickpunkte auf die Szene profitieren können.

### 11.5.3 Dynamische Visualisierung patientenspezifischer Daten

Das entwickelte skriptbasierte Konzept zur dynamischen Visualisierung wird derzeit sowohl im LIVERSURGERYTRAINER für die chirurgische Ausbildung als auch im ENT-SURGERYPLANNER zur Planung von Neck Dissections eingesetzt. Weiterentwicklungen fokussieren sich hierbei unter anderem auf die Analyse der 3D-Szene. Hierbei sind aktuell die automatische Detektion von optimalen Sichtpositionen sowie die Sichtbarkeitsbestimmung und Detektion verdeckender Objekte in 3D-Szenen Mittelpunkt einschlägiger Forschung [Mühler et al. 2007]. Entsprechende Informationen sollen zu optimalen Kamerapfaden für überblickende Animationen, aber auch zur Verbesserung der Sicht auf verdeckte Objekte führen. Zusammenfassend wird dies über die Analyse der 3D-Szene helfen, 3D-Visualisierungen in Richtung des *Medical Interactive Visualization Mantra* zu verbessern.

### 11.5.4 LIVERSURGERYTRAINER

Die Entwicklung des LIVERSURGERYTRAINER wird derzeit in einem BMBF-geförderten Projekt fortgesetzt. Hierbei konnte das Anwendungsspektrum neben der Tumorresektion und -ablation zusätzlich auf Leber-Lebend-Spenden ausgeweitet werden. Die didaktische Basis des Lernsystems wird anhand neuer Lernmodelle, wie dem Vier-Komponenten-Instruktionsdesign-Modell (4C/ID-Modell) von Merriënboer et al. [2002], weiter verfeinert [Cordes et al. 2006b]. Zur strukturellen Weiterentwicklung werden szenariobasierte Ansätze

nach Carroll [2000]; Rosson und Carroll [2001] verfolgt [Cordes et al. 2007b]. Weiterführend ist die Konzeption des LIVER SURGERY TRAINER ausführlich mit der Zielgruppe zu evaluieren. Erste Evaluationsanstrengungen werden hierbei aktuell von Cordes et al. [2007a] unternommen.

Eine Herausforderung zukünftiger Arbeiten ist vor allem die Evaluation des erreichten Lernerfolges in Konkurrenz und in Kombination zu alternativen Lernmethoden. Dies könnte einen neuen Entwicklungszyklus antreiben, der mögliche identifizierte Nachteile auszugleichen versucht. Untersuchungen der Kombination mit anderen Lernmethoden könnten zusätzlich Synergieeffekte unterschiedlicher Lernmethoden identifizieren, welche das Konzept des LIVER SURGERY TRAINER auf geeignet kombinierte Lernszenarios optimieren helfen.

Ausblickend ist bei der weiterführenden Entwicklung des LIVER SURGERY TRAINER die Untersuchung von kooperativem Lernen in der Interventionsplanung von Interesse. Arbeiten in diese Richtung könnten die Tür zu einem allgemein interessanten Forschungsfeld in Richtung kooperativem Planen unter Beteiligung mehrerer (interdisziplinärer) Experten aufstoßen.

---

# Literaturverzeichnis

---

- [Adobe 2007a] ADOBE: *Authorware*. <http://www.adobe.com/products/authorware/>, 2007. – [letzter Besuch 12.12.2007]
- [Adobe 2007b] ADOBE: *Flash*. <http://www.adobe.com/products/flash/>. Version: 2007. – [letzter Besuch 12.12.2007]
- [Adobe 2007c] ADOBE: *Macromedia Director*. <http://www.adobe.com/products/director>, 2007. – [letzter Besuch 02.03.2007]
- [Albu et al. 2003] ALBU, Alexandra B.; LAURENDEAU, Denis; HÉBERT, Luc; MOFFET, Helene; MOISAN, Christian: Three-Dimensional Reconstruction of the Bony Structures Involved in the Articular Complex of the Human Shoulder Using Shape-based Interpolation and Contour-Based Extrapolation. In: *International Conference on 3D Digital Imaging and Modeling (3DIM'03)*, IEEE Computer Society, Oktober 2003, S. 370–377
- [Allamandri et al. 1998] ALLAMANDRI, Fabio; CIGNONI, Paolo; MONTANI, Claudio; SCOPIGNO, Roberto: Adaptively adjusting Marching Cubes output to fit a trilinear reconstruction filter. In: *Eurographics Workshop on Visualization in Scientific Computing (VisSci'98)*, 1998, S. 25–34
- [Aspert et al. 2002] ASPERT, Nicolas; SANTA-CRUZ, Diego; EBRAHIMI, Touradj: MESH: Measuring Errors between Surfaces using the Hausdorff Distance. In: *IEEE Intern. Conf. on Multimedia and Expo* Bd. 1, 2002, S. 705–708. – <http://mesh.epfl.ch>
- [Autodesk 2007] AUTODESK: *Autodesk 3ds Max*. <http://www.autodesk.com/3dsmax>. Version: 2007. – [letzter Besuch 27.02.2007]
- [Bade et al. 2006a] BADE, Ragnar; HAASE, Jens; PREIM, Bernhard: Comparison of Fundamental Mesh Smoothing Algorithms for Medical Surface Models. In: *Simulation und Visualisierung (SimVis'06)*, SCS-Verlag, 2006, S. 289–304
- [Bade et al. 2007a] BADE, Ragnar; KONRAD, Olaf; PREIM, Bernhard: Reducing Artifacts in Surface Meshes Extracted from Binary Volumes. In: *Journal of WSCG* 15 (2007), Nr. 1–3, S. 67–74
- [Bade et al. 2004a] BADE, Ragnar; MIRSCHEL, Sebastian; HAASE, Tina; KRÜGER, Arno; HINDENNACH, Milo; OLDHAFFER, Karl J.; PREIM, Bernhard: Visualisierungstechniken für die Fallbasierte Chirurgieausbildung. In: *Rechnergestützte Lehr- und Lernsysteme in der Medizin*, 2004, S. 13–24
- [Bade et al. 2004b] BADE, Ragnar; MIRSCHEL, Sebastian; OLDHAFFER, Karl J.; PREIM, Bernhard: Ein fallbasiertes Lernsystem für die Behandlung von Lebertumoren. In: *Bildverarbeitung für die Medizin (BVM'04)*, Springer, 2004, S. 438–442
- [Bade et al. 2006b] BADE, Ragnar; RIEDEL, Ivonne; SCHMIDT, Lars; OLDHAFFER, Karl J.; PREIM, Bernhard: Combining Training and Computerized Planning of Oncologic Liver Surgery. In: *Bildverarbeitung für die Medizin (BVM'06)*, Springer, 2006, S. 409–413

- [Bade et al. 2005] BADE, Ragnar ; RITTER, Felix ; PREIM, Bernhard: Usability Comparison of Mouse-Based Interaction Techniques for Predictable 3D Rotation. In: *Proceedings of the 5th International Symposium on Smart Graphics*, Springer, August 2005 (Lecture Notes in Computer Science), S. 138–150
- [Bade et al. 2007b] BADE, Ragnar ; SCHUMANN, Christian ; SESHADHRI, Santhosh ; JANIGA, Gabor ; BÖLKE, Torsten ; KRISCHEK Özlem ; SKALEJ, Martin ; ROSE, Georg ; THÉVENIN, Dominique ; PREIM, Bernhard: High-quality Surface Generation for Flow Simulation in Cerebral Aneurysms. In: *Computer und Roboter Assistierte Chirurgie (CURAC'07)*, 2007, S. 125–128
- [Baer et al. 2007] BAER, Alexandra ; TIETJEN, Christian ; BADE, Ragnar ; PREIM, Bernhard: Hardware-Accelerated Stippling of Surfaces Derived from Medical Volume Data. In: *Eurographics/IEEE VGTC Symposium on Visualization (EuroVis'07)*, 2007, S. 235–242
- [Bajaj und Xu 2003] BAJAJ, Chandrajit L. ; XU, Guoliang: Anisotropic Diffusion on Surfaces and Functions on Surfaces. In: *ACM Transactions on Graphics* 22 (2003), Nr. 1, S. 4–32
- [Baumgart 1972] BAUMGART, Bruce G.: Winged-edge Polyhedron Representation / Computer Science Department, Stanford University. 1972 (CS-TR-72-320). – Forschungsbericht
- [Baumgart 1975] BAUMGART, Bruce G.: A polyhedron representation for computer vision. In: *National Computer Conference* Bd. 44, AFIPS Press, 1975, S. 589–596
- [Baumgartner und Payr 1999] BAUMGARTNER, Peter ; PAYR, Sabina: *Lernen mit Software*. 2. Auflage. Wien : Studien Verlag, 1999 (Lernen mit interaktiven Medien, Bd. 1)
- [Bell 1988] BELL, Gavin: *Bell's Virtual Trackball*. [http://members.tripod.com/professor\\_tom/index.html](http://members.tripod.com/professor_tom/index.html). Version: 1988. – [letzter Besuch 23.02.2007]
- [Belyaev und Ohtake 2003] BELYAEV, Alexander ; OHTAKE, Yutaka: A Comparison of Mesh Smoothing Methods. In: *Proc. Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics*, 2003, S. 83–87
- [Benninghoff und Drenckhahn 2002] BENNINGHOFF, Alfred ; DRENCKHAHN, Detlev: *Anatomie – Band 1*. München : Urban & Fischer, 2002
- [Bister et al. 1990] BISTER, M. ; CORNELIS, J. ; ROSENFELD, A.: A critical view of pyramid segmentation algorithms. In: *Pattern Recognition Letters* 11 (1990), Nr. 9, S. 605–617
- [Bloomenthal 1994] BLOOMENTHAL, Jules: An Implicit Surface Polygonizer. In: HECKBERT, Paul S. (Hrsg.): *Graphics Gems IV*. Boston : Morgan Kaufmann, 1994, S. 324–349
- [Bornik et al. 2005] BORNIK, Alexander ; REITINGER, Bernhard ; BEICHEL, Reinhard: Reconstruction and Representation of Tubular Structures using Simplex Meshes. In: *Proceedings of WSCG 2005 - Short Papers*, 2005, S. 61–65
- [Botsch et al. 2007] BOTSCH, Mario ; PAULY, Mark ; KOBELT, Leif ; ALLIEZ, Pierre ; LEVY, Bruno ; BISCHOFF, Stephan ; RÖSSL, Christian: Geometric Modeling Based on Polygonal Meshes. In: *SIGGRAPH 2007 Course Notes*, 2007

- [Botsch et al. 2002] BOTSCH, Mario ; STEINBERG, Stephan ; BISCHOFF, Stephan ; KOBBELT, Leif: OpenMesh – a Generic and Efficient Polygon Mesh Data Structure. In: *OpenSG Symposium*, 2002
- [Braude 2005] BRAUDE, Ilya: *Smooth 3D Surface Reconstruction from Contours of Biological Data with MPU Implicits*, Drexel University, Diplomarbeit, 2005. <http://dspace.library.drexel.edu/handle/1860/545>
- [Braude et al. 2007] BRAUDE, Ilya ; MARKER, Jeffrey ; MUSETH, Ken ; NISSANOV, Jonathan ; BREEN, David: Contour-Based Surface Reconstruction using MPU Implicit Models. In: *Graphical Models* 69 (2007), Nr. 2, S. 139–157
- [Brinkley und Rosse 1997] BRINKLEY, James F. ; ROSSE, Cornelius: The Digital Anatomist Distributed Framework and Its Applications to Knowledge-based Medical Imaging. In: *Journal of the American Medical Informatics Association* 4 (1997), Nr. 3, S. 165–183
- [Brinkley et al. 1999] BRINKLEY, James F. ; WONG, Benjamin A. ; HINSHAW, Kevin P. ; ROSSE, Cornelius: Design of an Anatomy Information System. In: *IEEE Computer Graphics and Applications* 19 (1999), Nr. 3, S. 38–48
- [Britton et al. 1978] BRITTON, Edward G. ; LIPSCOMB, James S. ; PIQUE, Michael E.: Making nested rotations convenient for the user. In: *ACM SIGGRAPH Computer Graphics* 12 (1978), Nr. 3, S. 222–227
- [Broelsch et al. 1990] BROELSCH, C.E. ; EMOND, J.C. ; WHITINGTON, P.F. ; THISTLETHWAITE, J.R. ; BAKER, A.L. ; LICHTOR, J.L.: Application of reduced-size liver transplants as split grafts, auxiliary orthotopic grafts, and living related segmental transplants. In: *Annals of Surgery* 212 (1990), Nr. 3, S. 368–377
- [Bruckner et al. 2005] BRUCKNER, Stefan ; GRIMM, Sören ; KANITSAR, Armin ; GRÖLLER, Meister E.: Illustrative Context-Preserving Volume Rendering. In: *Eurographics/IEEE VGTC Symposium on Visualization (EuroVis'05)*, 2005, S. 69–76
- [Bruckner und Gröller 2007] BRUCKNER, Stefan ; GRÖLLER, Eduard: Style Transfer Functions for Illustrative Volume Rendering. In: *Computer Graphics Forum* 26 (2007), Nr. 3, S. 715–724
- [Butz 1994] BUTZ, Andreas: Betty: Planning and Generating Animations for the Visualization of Movements and Spatial Relations. In: *Advanced Visual Interfaces (AVI'94)*, ACM Press, 1994, S. 53–58
- [Çakmak und Kühnapfel 2000] ÇAKMAK, H.K. ; KÜHNAPFEL, U.: Animation and Simulation Techniques for VR-Training Systems in Endoscopic Surgery. In: *European Workshop on Computer Animation*, 2000, S. 173–185
- [Carroll 2000] CARROLL, John M.: Five reasons for scenario-based design. In: *Interacting with Computers* 13 (2000), Nr. 1, S. 43–60
- [Casus 2007] CASUS: *Casus*. <http://www.casus.eu/>, 2007. – [letzter Besuch 24.07.2007]
- [CGAL 2007] CGAL: *Computational Geometry Algorithms Library*. <http://www.cgal.org>, 2007. – [letzter Besuch 03.08.2007]

- [Chapanis und Kinkade 1972] CHAPANIS, A. ; KINKADE, R.: Design of controls. In: VAN-COTT, H.P. (Hrsg.) ; KINKADE, R. (Hrsg.): *Human engineering guide to equipment design*. Washington DC : US Government Printing Office, 1972
- [Chen et al. 2004] CHEN, Chun-Yen ; CHENG, Kuo-Young ; LIAO, Hong-Yuan M.: Fairing of Polygon Meshes Via Bayesian Discriminant Analysis. In: *Journal of WSCG* 12 (2004), Nr. 1–3, S. 175–182
- [Chen et al. 1988] CHEN, Michael ; MOUNTFORD, S.Joy ; SELLEN, Abigail: A study in interactive 3D rotation using 2D control devices. In: *ACM SIGGRAPH Computer Graphics* 22 (1988), Nr. 4, S. 121–129
- [Chernyaev 1995] CHERNYAEV, Evgeni V.: Marching Cubes 33: Construction of Topologically Correct Isosurfaces / CERN, Geneva, Switzerland. 1995 (CN/95-17,). – Forschungsbericht. –  
<http://wwwinfo.cern.ch/asdoc/psdir/mc.ps.gz>
- [Cignoni et al. 1998] CIGNONI, P. ; ROCCHINI, C. ; SCOPIGNO, R.: Metro: Measuring Error on Simplified Surfaces. In: *Computer Graphics Forum* 17 (1998), Nr. 2, S. 167–174
- [Cignoni et al. 1997] CIGNONI, Paolo ; MARINO, Paola ; MONTANI, Claudio ; PUPPO, Enrico ; SCOPIGNO, Roberto: Speeding Up Isosurface Extraction Using Interval Trees. In: *IEEE Transactions on Visualization and Computer Graphics* 3 (1997), Nr. 2, S. 158–170
- [Cignoni et al. 2000] CIGNONI, Paulo ; GANOVELLI, Fabio ; MONTANI, Claudio ; SCOPIGNO, Roberto: Reconstruction of Topologically Correct and Adaptive Trilinear Isosurfaces. In: *Journal of Computers and Graphics* 24 (2000), Nr. 3, S. 19–31
- [Clarenz et al. 2000] CLARENZ, U. ; DIEWALD, U. ; RUMPF, M.: Anisotropic geometric diffusion in surface processing. In: *IEEE Visualization (VIS'00)*, 2000, S. 397–405
- [Conway 1997] CONWAY, M. J.: *Alice: Easy-to-Learn 3D Scripting for Novices*, Faculty of the School of Engineering and Applied Science at the University of Virginia, Diss., 1997
- [Cooper et al. 2003] COOPER, Stephen ; DANN, Wanda ; PAUSCH, Randy: Teaching objects-first in introductory computer science. In: *SIGCSE Technical Symposium on Computer Science Education*, 2003, S. 191–195
- [Cordes et al. 2006a] CORDES, Jeanette ; DORNHEIM, Jana ; PREIM, Bernhard ; HERTEL, Ilka ; STRAUSS, Gero: Preoperative Segmentation of Neck CT Datasets for the Planning of Neck Dissections. In: *SPIE Medical Imaging 2006: Image Processing* Bd. 6144, 2006, S. 1447–1456
- [Cordes et al. 2007a] CORDES, Jeanette ; MÜHLER, Konrad ; OLDHAFFER, Karl J. ; STAVROU, Gregor ; HILLERT, Christian ; PREIM, Bernhard: Evaluation of a training system of the computer-based planning of liver surgery. In: *Computer und Roboter Assistierte Chirurgie (CURAC'07)*, 2007, S. 151–154
- [Cordes et al. 2007b] CORDES, Jeanette ; MÜHLER, Konrad ; OLDHAFFER, Karl J. ; STAVROU, Gregor ; HILLERT, Christian ; PREIM, Bernhard: Szenariobasierte Entwicklung eines chirurgischen Trainingssystems. In: *Workshop der GMDS AG Computergestützte Lehr- und Lernsysteme in der Medizin (CBT'07)*, Shaker-Verlag, 2007, S. 17–30



- [Cordes et al. 2006b] CORDES, Jeanette ; MÜHLER, Konrad ; PREIM, Bernhard: Die Konzeption des LiverSurgery-Trainers. In: *GI-Workshop „Softwareassistenten - Computerunterstützung für die medizinische Diagnose und Therapieplanung“ im Rahmen der GI-Jahrestagung 2006*, Springer, 2006, S. 514–521
- [Delingette 1999] DELINGETTE, Hervé: General Object Reconstruction based on Simplex Meshes. In: *International Journal of Computer Vision* 32 (1999), Nr. 2, S. 111–146
- [Delingette und Ayache 2005] DELINGETTE, Hervé ; AYACHE, Nicholas: Hepatic Surgery Simulation. In: *Communications of the ACM* 48 (2005), Nr. 2, S. 31–36
- [Desbrun et al. 2006] DESBRUN, Mathieu ; KANSO, Eva ; TONG, Yiyang: Discrete differential forms for computational modeling. In: *ACM SIGGRAPH Courses*, 2006, S. 39–54
- [Desbrun et al. 1999] DESBRUN, Mathieu ; MEYER, Mark ; SCHRÖDER, Peter ; BARR, Alan H.: Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow. In: *SIGGRAPH*, 1999, S. 317–324
- [Diagram Group 1983] DIAGRAM GROUP: *The Human Body on File*. New York : Facts on File, 1983
- [Docs 'n Drugs 2007] DOCS 'N DRUGS: *Docs 'n Drugs: Die virtuelle Poliklinik*. <http://www.docs-n-drugs.de/system/>, 2007. – [letzter Besuch 25.07.2007]
- [Dornheim et al. 2006] DORNHEIM, Jana ; SEIM, Heiko ; PREIM, Bernhard ; HERTEL, Ilka ; STRAUSS, Gero: Segmentation of Neck Lymph Nodes in CT Datasets with Stable 3D Mass-Spring Models. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI'06)* Bd. 9, 2006, S. 904–911
- [Dürst 1988] DÜRST, Martin: Letters: Additional Reference to Marching Cubes. In: *ACM SIGGRAPH Computer Graphics* 22 (1988), Nr. 2, S. 72–73
- [Dyn et al. 2001] DYN, Nira ; HORMANN, Kai ; KIM, Sun-Jeong ; LEVIN, David: Optimizing 3D Triangulations Using Discrete Curvature Analysis. In: *Mathematical Methods for Curves and Surfaces: Oslo 2000*. Vanderbilt University Press, 2001, S. 135–146
- [Eckstein et al. 2007] ECKSTEIN, I. ; TONG, Y. ; KUO, C.-C. J. ; DESBRUN, M.: Volume-controlled surface fairing. In: *SIGGRAPH '07: ACM SIGGRAPH 2007 sketches*, ACM Press, August 2007, S. 8
- [Elisabeth André 1996] ELISABETH ANDRÉ, Thomas R. Jochen Müller M. Jochen Müller: WIP/PPP: Automatic Generation of Personalized Multimedia Presentations. In: *ACM International Multimedia Conference (Multimedia'96)*, 1996, S. 407–408
- [Engel et al. 2006] ENGEL, Klaus ; HADWIGER, Markus ; KNISS, Joe M. ; REZK-SALAMA, Christof ; WEISKOPF, Daniel: *Real-Time Volume Graphics*. AK Peters, 2006
- [ETH Zürich 2007] ETH ZÜRICH: *Hysteroscopy Simulator*. <http://www.hystsim.ethz.ch>, 2007. – [letzter Besuch 31.07.2007]

- [Fall et al. 2005] FALL, L.H. ; BERMAN, N.B. ; SMITH, S. ; WHITE, C.B. ; WOODHEAD, J.C. ; OLSON, A.L.: Multi-institutional Development and Utilization of a Computer-Assisted Learning Program for the Pediatrics Clerkship: The CLIPP Project. In: *Journal of the Association of American Medical Colleges: Academic Medicine (Acad Med)* 80 (2005), Nr. 9, S. 847–855
- [Fischer et al. 1996] FISCHER, M. ; SCHAUER, S. ; GRÄSEL, G. ; BÄHRING, T. ; MANDL, H. et al.: Modellversuch CASUS-Entwicklung eines Autorensystems für die problemorientierte Lehre in der Medizin. In: *Zeitschrift für Ärztliche Fortbildung* 90 (1996), Nr. 5, S. 385–389
- [Fleishman et al. 2003] FLEISHMAN, S. ; DRORI, I. ; COHEN-OR, D.: Bilateral mesh denoising. In: *ACM Transactions on Graphics* 22 (2003), Nr. 3, S. 950–953
- [Foley et al. 1996] FOLEY, James D. ; VAN DAM, Andries ; FEINER, Steven K. ; HUGHES, John F.: *Computer graphics : principles and practice*. 2. Auflage in C. Massachusetts : Addison-Wesley, 1996
- [Garde et al. 2005] GARDE, Sebastian ; BAUCH, Matthias ; HAAG, Martin ; HEID, Jörn ; HUWENDIEK, Sören et al.: CAMPUS - computer-based training in medicine as part of a problem-oriented educational strategy. In: *Studies in Learning, Evaluation, Innovation and Development* 2 (2005), Nr. 1, S. 10–19
- [Gibson 1998] GIBSON, Sarah F. F.: Constrained Elastic Surface Nets: Generating Smooth Surfaces from Binary Segmented Data. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI'98)*, Springer, 1998, S. 888–898
- [Gladilin et al. 2004] GLADILIN, E. ; ZACHOW, Stefan ; DEUFLHARD, Peter ; HEGE, Hans-Christian: Anatomy- and physics-based facial animation for craniofacial surgery simulations. In: *Medical & Biological Engineering & Computing* 42 (2004), Nr. 2, S. 167–170
- [Glüer et al. 2006] GLÜER, C.-C. ; BARKMANN, R. ; HAHN, H. K. ; MAJUMDAR, S. ; ECKSTEIN, F. ; NICKELSEN, T. N. ; BOLTE, H. ; DICKEN, V. ; HELLER, M.: Parametrische biomedizinische Bildgebung - was macht die Qualität quantitativer radiologischer Verfahren aus? In: *RöFo - Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren* 178 (2006), Nr. 12, S. 1187–1201
- [Golland et al. 1999] GOLLAND, P. ; KIKINIS, R. ; HALLE, M. ; UMANS, C. ; GRIMSON, W.E.L. ; SHENTON, M.E. ; RICHOLT, J.A.: AnatomyBrowser: A Novel Approach to Visualization and Integration of Medical Information. In: *Computer Aided Surgery* 4 (1999), S. 129–143
- [Grimm und Hughes 1995] GRIMM, Cindy M. ; HUGHES, John F.: Smooth Isosurface Approximation. In: *Eurographics Workshop Implicit Surface*, 1995, S. 57–67
- [Haase 2005] HAASE, Jens: *Glättung von Polygonnetzen in medizinischen Visualisierungen*, Fakultät für Informatik, O.-v.-Guericke Universität Magdeburg, Diplomarbeit, 2005
- [Hahn et al. 2001] HAHN, Horst K. ; PREIM, Bernhard ; SELLE, Dirk ; PEITGEN, Heinz-Otto: Visualization and Interaction Techniques for the Exploration of Vascular Structures. In: *IEEE Visualization (VIS'01)*, 2001, S. 395–402

- [Hamann et al. 1997] HAMANN, Bernd ; TROTTS, Issac J. ; FARIN, Gerald E.: On Approximating Contours of the Piecewise Trilinear Interpolant Using Triangular Rational-Quadratic Bézier Patches. In: *IEEE Transactions on Visualization and Computer Graphics* 3 (1997), Nr. 3, S. 215–227
- [Hameiri und Shimshoni 2002] HAMEIRI, Eyal ; SHIMSHONI, Ilan: Estimating the principal curvatures and the darboux frame from real 3d range data. In: *3D Data Processing Visualization and Transmission (3DPVT'02)*, 2002, S. 258–266
- [Hauser et al. 2002] HAUSER, Helwig ; LEDERMANN, Florian ; DOLEISCH, Helmut: Angular Brushing of Extended Parallel Coordinates. In: *IEEE Information Visualization (InfoVis'02)*, IEEE Computer Society Press, 2002, S. 127–133
- [Hauser und Wohlfart 2008] HAUSER, Helwig ; WOHLFART, Michael: *Story Telling Informed Volume Visualization*. 2008. – Eingereicht zu IEEE Transactions on Visualization and Computer Graphics
- [Hege et al. 1997] HEGE, Hans-Christian ; SEEBASS, Martin ; STALLING, Detlev ; ZÖCKLER, Malte: A Generalized Marching Cubes Algorithm Based on Non-Binary Classifications / Konrad-Zuse-Zentrum. 1997 (SC-97-05). – Forschungsbericht
- [Henriksen et al. 2004] HENRIKSEN, Knud ; SPORRING, Jon ; HORNBAEK, Kasper: Virtual trackballs revisited. In: *IEEE Transactions on Visualization and Computer Graphics* 10 (2004), Nr. 2, S. 206–216
- [Herczeg 2006] HERCZEG, Michael: *Interaktionsdesign*. München : Oldenbourg Wissenschaftsverlag, 2006
- [Höhne et al. 2000] HÖHNE, Karl-Heinz ; PFLESSER, Bernhard ; POMMERT, Andreas ; RIEMER, Martin ; SCHUBERT, Rainer ; SCHIEMANN, Thomas ; TIEDE, Ulf ; SCHUMACHER, Udo: A Realistic Model of the Inner Organs from the Visible Human Data. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI'00)*, 2000, S. 776–785
- [Hildebrandt und Polthier 2004] HILDEBRANDT, Klaus ; POLTHIER, Konrad: Anisotropic Filtering of Non-Linear Surface Features. In: *Computer Graphics Forum* 23 (2004), Nr. 3, S. 391–400
- [Hinckley et al. 1998] HINCKLEY, Ken ; PAUSCH, Randy ; PROFFITT, Dennis ; KASSELL, Neal F.: Two-handed virtual manipulation. In: *ACM Transactions on Computer-Human Interaction* 5 (1998), Nr. 3, S. 260–302
- [Hinckley et al. 1997] HINCKLEY, Ken ; TULLIO, Joe ; PAUSCH, Randy ; PROFFITT, Dennis ; KASSELL, Neal: Usability analysis of 3D rotation techniques. In: *Proc. of ACM Symposium on User Interface Software and Technology*, ACM Press, 1997, S. 1–10
- [Hultquist 1990] HULTQUIST, Jeff: A Virtual Trackball. In: GLASSNER, Andrew S. (Hrsg.): *Graphics Gems*. San Diego : Morgan Kaufmann, 1990, S. 462–463
- [Huwendiek et al. 2006] HUWENDIEK, Sören ; KÖPF, Sabine ; HÖCKER, Britta ; HEID, Jörn ; BAUCH, Matthias et al.: Fünf Jahre Erfahrung mit dem curricularen Einsatz des fall- und webbasierten Lernsystems CAMPUS-Pädiatrie an der Medizinischen Fakultät Heidelberg. In: *GMS Zeitschrift für Medizinische Ausbildung* 23 (2006), Nr. 1, S. 71–72

- [iInTIME 2007] iInTIME: *CLIPP: Computer-assisted Learning in Pediatrics Program*. <http://www.clippcases.org/>, 2007. – [letzter Besuch 24.07.2007]
- [Inselberg 1985] INSELBERG, Alfred: The plane with parallel coordinates. In: *The Visual Computer* 1 (1985), S. 69–92
- [Iserhardt-Bauer et al. 2001] ISERHARDT-BAUER, Sabine ; HASTREITER, Peter ; ERTL, Thomas ; EBERHARDT, K. ; TOMANDL, B.: Case Study: Medical Web Service for the Automatic 3D Documentation for Neuroradiological Diagnosis. In: *IEEE Visualization (VIS'01)*, 2001, S. 425–428
- [Iserhardt-Bauer et al. 2002] ISERHARDT-BAUER, Sabine ; HASTREITER, Peter ; TOMANDL, B. ; KÖSTNER, N. ; SCHEMPERSHOFE, M. ; NISSEN, U. ; ERTL, Thomas: Standardized Analysis of Intracranial Aneurysms Using Digital Video Sequences. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI'02)*, 2002, S. 411–418
- [Issing und Klimsa 1997] ISSING, Ludwig J. ; KLIMSA, Paul: *Information und Lernen mit Multimedia*. 2. Auflage. Weinheim : Psychologie Verlags Union, 1997
- [Itoh und Koyamada 1995] ITOH, Takayuki ; KOYAMADA, Koji: Automatic Isosurface Propagation Using an Extrema Graph and Sorted Boundary Cell Lists. In: *IEEE Transactions on Visualization and Computer Graphics* 1 (1995), Nr. 4, S. 319–327
- [Jacob und Oliver 1995] JACOB, Inés ; OLIVER, Javier: Evaluation of Techniques for Specifying 3D Rotations with 2D Input Device. In: *Proc. of (HCI'95)*, Cambridge University Press, 1995, S. 63–76
- [Jalba und Roerdink 2006] JALBA, Andrei ; ROERDINK, Jos B. T. M.: Efficient Surface Reconstruction from Noisy Data using Regularized Membrane Potentials. In: *Eurographics/IEEE VGTC Symposium on Visualization (EuroVis'06)*, Eurographics Association, 2006, S. 83–90
- [Janke et al. 2006] JANKE, Claudia ; TIETJEN, Christian ; BAER, Alexandra ; ZWICK, Carola ; PREIM, Bernhard ; HERTEL, Ilka ; STRAUSS, Gero: Design und Realisierung eines Softwareassistenten zur Planung von Halsoperationen. In: *Mensch & Computer 2006: Mensch und Computer im StrukturWandel*, 2006, S. 373–378
- [Jones et al. 2003] JONES, T. R. ; DURAND, F. ; DESBRUN, Mathieu: Non-iterative, feature-preserving mesh smoothing. In: *ACM Transactions on Graphics* 22 (2003), Nr. 3, S. 943–949
- [Kalogerakis et al. 2007] KALOGERAKIS, Evangelos ; SIMARI, Patricio ; NOWROUZEZHAI, Derek ; SINGH, Karan: Robust statistical estimation of curvature on discretized surfaces. In: *Eurographics Symposium on Geometry Processing*, 2007, S. 13–22
- [Karp und Feiner 1993] KARP, Peter ; FEINER, Steven: Automated presentation planning of animation using task decomposition with heuristic reasoning. In: *Graphics Interface*, 1993, S. 118–127
- [Köckerling und Waclawiczek 1999] KÖCKERLING, Ferdinand ; WACLAWICZEK, Hans: *Leberchirurgie, Anatomie – Operationstechniken – Komplikationsvermeidung*. Heidelberg : Johann Ambrosius Barth Verlag, 1999

- [Keim 2005] KEIM, Daniel: *Scaling Visual Analytics to Very Large Data Sets*. Talk at Workshop on Visual Analytics, Darmstadt, Germany, Juni 2005
- [Khalil et al. 2005a] KHALIL, Mohammed K. ; PAAS, Fred ; JOHNSON, Tristan E. ; PAYER, Andrew F.: Interactive and dynamic visualizations in teaching and learning of anatomy: A cognitive load perspective. In: *The Anatomical Record Part B: The New Anatomist* 286B (2005), Nr. 1, S. 8–14
- [Khalil et al. 2005b] KHALIL, Mohammed K. ; PAAS, Fred ; JOHNSON, Tristan E. ; PAYER, Andrew F.: Design of interactive and dynamic anatomical visualizations: The implication of cognitive load theory. In: *The Anatomical Record Part B: The New Anatomist* 286B (2005), Nr. 1, S. 15–20
- [Kikinis et al. 1996] KIKINIS, Ron ; SHENTON, Martha E. ; IOSIFESCU, Dan V. ; MCCARLEY, Robert W. ; SAIVIROONPORN, Pairash ; HOKAMA, Hiroto H. et al.: A Digital Brain Atlas for Surgical Planning, Model-Driven Segmentation, and Teaching. In: *IEEE Transactions on Visualization and Computer Graphics* 2 (1996), Nr. 3, S. 232–241
- [Kitware Inc. 2007] KITWARE INC.: *VTK 5.1.0 Documentation*.  
<http://www.vtk.org/doc/nightly/html>, 2007. – [stand 25.03.2007]
- [Kitware Inc 2007] KITWARE INC: *VTK Home Page*. <http://www.vtk.org>. Version: 2007. – [letzter Besuch 27.02.2007]
- [Kleimann 2007] KLEIMANN, Bernd: E-Learning 2.0 an deutschen Universitäten. In: *Tagung der Gesellschaft für Medien in der Wissenschaft (GMW'07)*, 2007
- [Kobbelt 1997] KOBBELT, Leif: Discrete Fairing. In: *Mathematics of Surfaces*, 1997, S. 101–130
- [Kojekine et al. 2003] KOJEKINE, Nikita ; HAGIWARA, Ichiro ; SAVCHENKO, Vladimir: Software tools using CSRBFs for processing scattered data. In: *Journal of Computers and Graphics* 27 (2003), Nr. 2, S. 311–319
- [Krüger et al. 2005a] KRÜGER, Arno ; TIETJEN, Christian ; HINTZE, Jana ; PREIM, Bernhard ; HERTEL, Ilka ; STRAUSS, Gero: Analysis and Exploration of 3D-Visualizations for Neck Dissection Planning. In: *International Congress on Computer Assisted Radiology and Surgery (CARS'05)*, 2005, S. 497–503
- [Krüger et al. 2005b] KRÜGER, Arno ; TIETJEN, Christian ; HINTZE, Jana ; PREIM, Bernhard ; HERTEL, Ilka ; STRAUSS, Gero: Interactive Visualization for Neck Dissection Planning. In: *Eurographics/IEEE VGTC Symposium on Visualization (EuroVis'05)*, 2005, S. 295–302
- [Kuntz und Weymuller 1999] KUNTZ, A.L. ; WEYMULLER, E.A.: Impact of neck dissection on quality of life. In: *Laryngoscope* 109 (1999), Nr. 8, S. 1334–1338
- [Lakare und Kaufman 2003] LAKARE, Sarang ; KAUFMAN, Arie: Anti-aliased volume extraction. In: *Eurographics/IEEE VGTC Symposium on Visualization (VisSym'03)*, 2003, S. 113–122

- [Lang et al. 2005a] LANG, Hauke ; RADTKE, Arnold ; HINDENNACH, Milo ; SCHROEDER, Tobias ; FRÜHAUF, Nils R. ; MALAGÓ, Massimo et al.: Impact of Virtual Tumor Resection and Computer-Assisted Risk Analysis on Operation Planning and Intraoperative Strategy in Major Hepatic Resection. In: *Archives of Surgery* 140 (2005), Nr. 7, S. 629–638
- [Lang et al. 2005b] LANG, Hauke ; RADTKE, Arnold ; LIU, C ; SOTIROPOULOS, G. C. ; HINDENNACH, Milo ; SCHROEDER, Tobias et al.: Improved assessment of functional resectability in repeated hepatectomy by computer-assisted operation planning. In: *Hepato-Gastroenterology* 52 (2005), Nr. 66, S. 1645–1648
- [Lee et al. 2005] LEE, Chang H. ; VARSHNEY, Amitabh ; JACOBS, David W.: Mesh Saliency. In: *ACM Transactions on Graphics* 24 (2005), Nr. 3, S. 659–666
- [Lee und Owens 2000] LEE, W.W. ; OWENS, D.L.: *Multimedia-based instructional design: Computer-based training, web-based training, and distance learning*. San Francisco : Jossey-Bass/Pfeiffer, 2000
- [Leven et al. 2006] LEVEN, Franz-Josef ; BAUCH, Matthias ; HAAG, Martin: E-Learning in der Mediziner Ausbildung in Deutschland: Status und Perspektiven. In: *GMS Medizinische Informatik, Biometrie und Epidemiologie* 2 (2006), Nr. 3, S. Doc28
- [Lopes und Brodlie 2003] LOPES, Adriano ; BRODLIE, Ken: Improving the Robustness and Accuracy of the Marching Cubes Algorithm for Isosurfacing. In: *IEEE Transactions on Visualization and Computer Graphics* 9 (2003), Nr. 1, S. 16–29
- [Lorensen und Cline 1987] LORENSEN, William E. ; CLINE, Harvey E.: Marching Cubes: A high resolution 3D surface construction algorithm. In: *ACM SIGGRAPH Computer Graphics* 21 (1987), Nr. 4, S. 163–169
- [LRSMed 2007] LRSMED: *Learning Resource Server Medizin*. Universitätsklinikum Essen : <http://mmedia.medizin.uni-essen.de/portal/>, 2007. – [stand 10.04.2007, letzter Besuch 17.08.2007]
- [Mader und Stöckl 1999] MADER, Günter ; STÖCKL, Walter: *Virtuelles Lernen: Begriffsbestimmung und aktuelle empirische Befunde*. Wien : Studien Verlag, 1999 (Lernen mit interaktiven Medien, Bd. 5)
- [Marfil et al. 2006] MARFIL, R. ; MOLINA-TANCO, L. ; BANDERA, A. ; RODRÍGUEZ, J. A. ; SANDOVAL, F.: Pyramid segmentation algorithms revisited. In: *Pattern Recognition* 39 (2006), Nr. 8, S. 1430–1451
- [Marten et al. 2005] MARTEN, K. ; RUMMENY, E. J. ; ENGELKE, C.: Computerassistierter Nachweis und automatisierte Volumetrie pulmonaler Rundherde in der Multislice-CT: Aktueller Stand und Perspektiven. In: *RöFo - Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren* 177 (2005), Nr. 2, S. 188–196
- [Martens et al. 1999] MARTENS, Alke ; BERNAUER, J. ; ILLMANN, T. ; SCHEUERER, C. ; SEITZ, A. ; WEBER, M.: Docs 'n Drugs - ein webbasiertes, multimediales Lehrsystem für die Medizin. In: *Workshop der GMDS AG Computergestützte Lehr- und Lernsysteme in der Medizin (CBT'99)*, 1999, S. 65–72

- [Martens et al. 2001] MARTENS, Alke ; BERNAUER, J. ; ILLMANN, T. ; SEITZ, A.: Docs 'n Drugs - The Virtual Polyclinic: An Intelligent Tutoring System for Web-Based and Case-Oriented Training in Medicine. In: *American Medical Informatics Conference (AMIA '01)*, 2001, S. 433–437
- [Masutani et al. 1996] MASUTANI, Yoshitaka ; MASAMUNE, Ken ; DOHI, Takeyoshi: Region-Growing Based Feature Extraction Algorithm for Tree-Like Objects. In: *Visualization in Biomedical Computing (VBC'96)* Bd. 1131, Springer, LNCS, 1996, S. 161–171
- [McCue 2005] MCCUE, Patrick: The crucial role of animated children's educational games. In: *SIGGRAPH '05: ACM SIGGRAPH 2005 Educators program*. New York, NY, USA : ACM Press, 2005
- [MediCase 2007] MEDICASE: *CAMPUS Lehr- und Lernsystem*. <http://www.medicase.de>, 2007. – [letzter Besuch 26.07.2007]
- [Mehrabi et al. 2000] MEHRABI, A. ; GLÜCKSTEIN, Ch. ; BENNER, A. ; HASHEMI, B. ; HERFARTH, Ch. ; KALINOWSKI, F.: A new way for surgical education-development and evaluation of a computer-based training module. In: *Computers in Biology and Medicine* 30 (2000), S. 97–109
- [Mercury Computer Systems 2007] MERCURY COMPUTER SYSTEMS: *Amira - Advanced 3D Visualization and Volume Modeling*. <http://www.amiravis.com/>, 2007. – [letzter Besuch 28.03.2007]
- [Merriënboer et al. 2002] MERRIËNBOER, Jeroen J. G. v. ; CLARK, Richard E. ; CROOCK, Marcel B. M. d.: Blueprints for Complex Learning: The 4C/ID-Model. In: *Educational Technology Research and Development* 50 (2002), Nr. 2, S. 39–61
- [MeVis 2007] MEVIS: *MeVis*. <http://www.mevis.de>, 2007. – [letzter Besuch 09.07.2007]
- [MeVis BreastCare 2007] MEVIS BREASTCARE: *MeVis BreastCare*. <http://www.mevisbreastcare.de>, 2007. – [letzter Besuch 13.06.2007]
- [MeVis Research 2007] MEVIS RESEARCH: *MeVisLab*. <http://www.mevislab.de>, 2007. – [letzter Besuch 28.03.2007]
- [Mewes 2007] MEWES, Maik: *Implementierung und Evaluierung von Visualisierungs- und Interaktionstechniken in Patientendatenbanken*, Fakultät für Informatik, O.-v.-Guericke Universität Magdeburg, Diplomarbeit, 2007
- [Meyer et al. 2002] MEYER, Mark ; DESBRUN, Mathieu ; SCHRÖDER, Peter ; BARR, Alan H.: Discrete differential-geometry operators for triangulated 2-manifolds. In: *International Workshop on Visualization and Mathematics (VisMath'02)*, Springer-Verlag, 2002, S. 35–57
- [Mühler et al. 2007] MÜHLER, Konrad ; NEUGEBAUER, Mathias ; TIETJEN, Christian ; PREIM, Bernhard: Viewpoint Selection for Intervention Planning. In: *Eurographics/IEEE VGTC Symposium on Visualization (EuroVis'07)*, 2007, S. 267–274
- [Mirschel 2004] MIRSCHEL, Sebastian: *Erstellung eines Prototypen für ein fallbasiertes Lernsystem in der Leberchirurgie*, Fakultät für Informatik, O.-v.-Guericke Universität Magdeburg, Diplomarbeit, 2004

- [Montani et al. 1994] MONTANI, Claudio ; SCATENI, Riccardo ; SCOPIGNO, Roberto: A modified look-up table for implicit disambiguation of Marching Cubes. In: *The Visual Computer* 10 (1994), Nr. 6, S. 353–355
- [Muehler 2005] MUEHLER, Konrad: *Skriptbasierte Generierung von Animationen für die medizinische Ausbildung und Therapieplanung*, Fakultät für Informatik, O.-v.-Guericke Universität Magdeburg, Diplomarbeit, 2005
- [Muehler et al. 2006] MUEHLER, Konrad ; BADE, Ragnar ; PREIM, Bernhard: Skriptbasierte Animationen für die Operationsplanung und Ausbildung. In: *Bildverarbeitung für die Medizin (BVM'06)*, 2006, S. 296–300
- [Neubauer 2005] NEUBAUER, André: *Virtual Endoscopy for Preoperative Planning and Training of Endonasal Transsphenoidal Pituitary Surgery*, VRVis und ICGA TU Wien, Diss., 2005
- [Neubauer et al. 2004] NEUBAUER, André ; FORSTER, Marie-Theres ; WEGENKITTL, Rainer ; MROZ, Lukas ; BÜHLER, Katja: Efficient Display of Background Objects for Virtual Endoscopy Using Flexible First-Hit Ray Casting. In: *Eurographics/IEEE VGTC Symposium on Visualization (VisSym'04)*, 2004, S. 301–310
- [Nielson 2004] NIELSON, Gregory M.: Dual Marching Cubes. In: *IEEE Visualization (VIS'04)*, 2004, S. 489–496
- [Nielson und Hamann 1991] NIELSON, Gregory M. ; HAMANN, Bernd: The asymptotic decider: resolving the ambiguity in marching cubes. In: *IEEE Visualization (VIS'91)*, 1991, S. 83–91
- [Oeltze und Preim 2004] OELTZE, Steffen ; PREIM, Bernhard: Visualization of Anatomic Tree Structures with Convolution Surfaces. In: *Eurographics/IEEE VGTC Symposium on Visualization (EuroVis'04)*, 2004, S. 311–320
- [Oeltze und Preim 2005] OELTZE, Steffen ; PREIM, Bernhard: Visualization of Vascular Structures: Method, Validation and Evaluation. In: *IEEE Transactions on Medical Imaging* 24 (2005), Nr. 4, S. 540–548
- [Ohtake et al. 2003] OHTAKE, Yutaka ; BELYAEV, Alexander ; ALEXA, Marc ; TURK, Greg ; SEIDEL, Hans-Peter: Multi-level Partition of Unity Implicits. In: *ACM Transactions on Graphics* 22 (2003), Nr. 3, S. 463–470
- [OpenMesh 2007] OPENMESH: *OpenMesh*. <http://www.openmesh.org>, 2007. – [letzter Besuch 03.08.2007]
- [Page et al. 2002] PAGE, D. L. ; SUN, Y. ; KOSCHAN, A. F. ; PAIK, J. ; ABIDI, M. A.: Surface parametrization and curvature measurement of arbitrary 3-d objects: Five practical methods. In: *Graphical Models* 64 (2002), Nr. 3/4, S. 199–229
- [Pakdel und Samavati 2007] PAKDEL, Hamid-Reza ; SAMAVATI, Faramarz F.: Incremental subdivision for triangle meshes. In: *International Journal of Computational Science and Engineering* 3 (2007), Nr. 1, S. 80–92



- [Partala 1999] PARTALA, Timo: Controlling a Single 3D Object: Viewpoint Metaphors, Speed and Subjective Satisfaction. In: *Proc. of Conference on Human-Computer Interaction (INTERACT'99)*, IOS Press, 1999, S. 486–493
- [Pommert 2004] POMMERT, Andreas: *Simulationsstudien zur Untersuchung der Bildqualität für die 3D-Visualisierung tomographischer Volumendaten*, Universität Hamburg, Fachbereich Informatik, Diss., 2004
- [Pommert et al. 2006] POMMERT, Andreas; HÖHNE, Karl H.; BURMESTER, Eike; GEHRMANN, Sebastian; LEUWER, Rudolf et al.: Computer-based anatomy: A prerequisite for computer-assisted radiology and surgery. In: *Academic Radiology* 13 (2006), Nr. 1, S. 104–112
- [Pommert et al. 2001] POMMERT, Andreas; HÖHNE, Karl H.; PFLESSER, Bernhard; RICHTER, Ernst; RIEMER, Martin et al.: Creating a High-resolution Spatial/Symbolic Model of the Inner Organs Based on the Visible Human. In: *Medical Image Analysis* 5 (2001), Nr. 3, S. 221–228
- [Preim 1998] PREIM, Bernhard: *Interaktive Illustrationen und Animationen zur Erklärung komplexer räumlicher Zusammenhänge*, Fakultät für Informatik, O.-v.-Guericke Universität Magdeburg, Diss., 1998
- [Preim und Bartz 2007] PREIM, Bernhard; BARTZ, Dirk: *Visualization in Medicine: Theory, Algorithms, and Applications*. London : Morgan Kaufmann, 2007
- [Preim et al. 2003] PREIM, Bernhard; HINDENNACH, Milo; SPINDLER, Wolf; SCHENK, Andrea; LITTMANN, Arne; PEITGEN, Heinz-Otto: Visualisierungs- und Interaktionstechniken für die Planung lokaler Therapien. In: *Simulation und Visualisierung (SimVis'03)*, SCS, 2003, S. 237–248
- [Preim und Oeltze 2007] PREIM, Bernhard; OELTZE, Steffen: 3D Visualization of Vasculature: An Overview. In: LINSEN, Lars (Hrsg.); HAGEN, Hans (Hrsg.); HAMANN, Bernd (Hrsg.): *Visualization in Medicine and Life Sciences*. Springer, 2007, S. 19–39
- [Preim et al. 1997] PREIM, Bernhard; RAAB, Andreas; STROTHOTTE, Thomas: Coherent Zooming of Illustrations with 3D-Graphics and Text. In: *Graphics Interface (GI'97)*, 1997, S. 105–113
- [Preim et al. 1996a] PREIM, Bernhard; RITTER, Alf; STEINICKE, Gunnar: Gestaltung von Animationen zur Erklärung komplexer 3D-Modelle. In: *Simulation und Animation für Planung, Bildung und Präsentation*, 1996, S. 255–266
- [Preim et al. 1996b] PREIM, Bernhard; RITTER, Alf; STROTHOTTE, Thomas: Illustrating Anatomic Models: A Semi-Interactive Approach. In: *Visualization in Biomedical Computing (VBC'96)*, Springer-Verlag, 1996, S. 23–32
- [Preim und Strothotte 1996] PREIM, Bernhard; STROTHOTTE, Thomas: Illustrating and Annotating 3D Geometric Models for Explanations: System Architecture. In: *European Conference on Artificial Intelligence (AI'96)*, 1996, S. 1–8
- [Pressey 1926] PRESSEY, Sidney L.: A simple apparatus which gives tests and scores- and teaches. In: *School and Society* 23 (1926), Nr. 586, S. 373–376

- [Prewer und Kitchen 2001] PREWER, David ; KITCHEN, Les: Soft Image Segmentation by Weighted Linked Pyramid. In: *Pattern Recognition Letters* 22 (2001), Nr. 2, S. 123–132
- [Priesching 1986] PRIESCHING, Alfred: *Leberresektionen. Chirurgische Anatomie, Indikationen, Technik*. München : Urban & Schwarzenberg, 1986
- [Raia et al. 1989] RAIA, S. ; NERY, J.R. ; MIES, S.: Liver transplantation from live donors. In: *Lancet* 2 (1989), Nr. 8661, S. 497–498
- [Rao und Card 1994] RAO, R. ; CARD, S. K.: The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In: *SIGCHI conference on Human factors in computing systems (CHI'94)*. New York, NY, USA : ACM Press, 1994. – ISBN 0-89791-650-6, S. 318–322
- [Reichert et al. 2007] REICHERT, F. ; BRASCH, C. ; HEID, Jörn ; BOSSE, Hans-Martin ; HAAG, Martin et al.: Design virtueller Patienten für die medizinische Ausbildung: Evaluation zweier Abspielvarianten des CAMPUS-Systems durch Medizinstudierende. In: *Workshop der GMDS AG Computergestützte Lehr- und Lernsysteme in der Medizin (CBT'07, 2007, S. 222–234*
- [Reitinger et al. 2005] REITINGER, Bernhard ; BORNIK, Alexander ; BEICHEL, Reinhard: Consistent Mesh Generation for Non-Binary Medical Datasets. In: *Bildverarbeitung für die Medizin (BVM'05)*, Springer, 2005, S. 183–187
- [Rhijn und Mulder 2006] RHIJN, Arjen v. ; MULDER, Jurriaan D.: Spatial input device structure and bimanual object manipulation in virtual environments. In: *ACM Symposium on Virtual Reality Software and Technology (VRST'06)*, 2006, S. 51–60
- [Riedel 2005] RIEDEL, Ivonne: *Erstellung einer Ausbildungs- und Trainingskomponente für die Tumorablationssplanung in der Leberchirurgie*, Fakultät für Informatik, O.-v.-Guericke Universität Magdeburg, Diplomarbeit, 2005
- [Riedel et al. 2000] RIEDEL, Jens ; SINGER, Reiner ; HEID, Jörn ; LEVEN, Franz J.: CAMPUS: Ein simulatives und flexibles fallbasiertes Webbased Training System für die medizinische Ausbildung. In: *Workshop der GMDS AG Computergestützte Lehr- und Lernsysteme in der Medizin (CBT'00, 2000, S. 65–72*
- [Right Hemisphere 2007] RIGHT HEMISPHERE: *Deep Exploration*. <http://www.righthemisphere.com/products/dexp>. Version: 2007. – [letzter Besuch 27.02.2007]
- [Ritter 2005] RITTER, Felix: *Interaktives Illustrieren von Informationsräumen: Räumliche und funktionale Zusammenhänge spielerisch begreifen*, Fakultät für Informatik, O.-v.-Guericke Universität Magdeburg, Diss., 2005
- [Ritter et al. 2001] RITTER, Felix ; DEUSSEN, Oliver ; PREIM, Bernhard ; STROTHOTTE, Thomas: Virtual 3D Puzzles: A New Method for Exploring Geometric Models in VR. In: *IEEE Computer Graphics and Applications* 21 (2001), Nr. 5, S. 11–13
- [Ritter et al. 2000] RITTER, Felix ; PREIM, Bernhard ; DEUSSEN, Oliver ; STROTHOTTE, Thomas: Using a 3D Puzzle as a Metaphor for Learning Spatial Relations. In: *Graphics Interface (GI'00)*, Morgan Kaufmann, 2000, S. 171–178

- [Rößler et al. 2007] RÖSSLER, F.; NENOV, M.; ISERHARDT-BAUER, Sabine; HASTREITER, Peter; ERTL, Thomas: Investigating 3D Object Movies for Web-Based Medical Visualization. In: *Computer und Roboter Assistierte Chirurgie (CURAC'07)*, 2007, S. 209–212
- [Rosson und Carroll 2001] ROSSON, Mary B.; CARROLL, John M.: *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Morgan Kaufmann, 2001
- [Roth et al. 2007] ROTH, Alexander; SPROTTE, René; BÜSE, Daniel; HAMPEL, Thorsten: koaLA - Integrierte Lern- und Arbeitswelten für die Universität 2.0. In: *Deutsche e-Learning Fachtagung Informatik (DeLFI'07)*, 2007, S. 221–232
- [Schmidt 2005] SCHMIDT, Lars: *Erstellung einer Ausbildungs- und Trainingskomponente für die Resektionsplanung in der Leberchirurgie*, Fakultät für Informatik, O.-v.-Guericke Universität Magdeburg, Diplomarbeit, 2005
- [Schulz 2005] SCHULZ, Christian: *Approximation von Krümmungsinformationen zur Umsetzung von Techniken zur illustrativen medizinischen Visualisierung*, Fakultät für Informatik, O.-v.-Guericke Universität Magdeburg, Diplomarbeit, 2005
- [Schumann 2006] SCHUMANN, Christian: *Visualisierung baumartiger anatomischer Strukturen mit MPU Implicits*, Fakultät für Informatik, O.-v.-Guericke Universität Magdeburg, Diplomarbeit, 2006
- [Schumann et al. 2007] SCHUMANN, Christian; OELTZE, Steffen; BADE, Ragnar; PREIM, Bernhard: Model-free Surface Visualization of Vascular Trees. In: *Eurographics/IEEE VGTC Symposium on Visualization (EuroVis'07)*, 2007, S. 283–290
- [Selle et al. 2002] SELLE, Dirk; PREIM, Bernhard; SCHENK, Andrea; PEITGEN, Heinz-Otto: Analysis of Vasculature for Liver Surgical Planning. In: *IEEE Transactions on Medical Imaging* 21 (2002), Nr. 11, S. 1344–1357
- [SGI 2007] SGI: *SGI - Developer Central Open Source | Open Inventor*. <http://oss.sgi.com/projects/inventor>. Version: 2007. – [letzter Besuch 27.02.2007]
- [Shneiderman 1996] SHNEIDERMAN, Ben: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: *IEEE Symposium on Visual Languages*, IEEE Computer Society Press, 1996, S. 336–343
- [Shoemake 1992] SHOEMAKE, Ken: ARCBALL: A user interface for specifying three-dimensional orientation using a mouse. In: *Graphics Interface (GI'92)*. Vancouver: Morgan Kaufmann, 1992, S. 151–156
- [Sierra et al. 2004] SIERRA, Raimundo; BAJKA, Michael; KARADOĞAN, Celalettin; SZÉKELY, Gábor; HARDERS, Matthias: Coherent Scene Generation for Surgical Simulators. In: *Medical Simulation Symposium*, 2004, S. 221–229
- [Sierra et al. 2003a] SIERRA, Raimundo; BAJKA, Michael; SZÉKELY, Gábor: Evaluation of Different Pathology Generation Strategies for Surgical Training Simulators. In: *International Congress on Computer Assisted Radiology and Surgery (CARS'03)*, 2003, S. 376–381

- [Sierra et al. 2003b] SIERRA, Raimundo ; BAJKA, Michael ; SZÉKELY, Gábor: Pathology Growth Model Based on Particles. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI'03)* Bd. 1, Springer, 2003, S. 25–32
- [Skinner 1968] SKINNER, Burrhus F.: *The Technology of Teaching*. New York : John Wiley, 1968
- [Spenke 2001] SPENKE, M.: Visualization and interactive analysis of blood parameters with InfoZoom. In: *Artificial Intelligence in Medicine 22* (2001), Nr. 2, 159-172. [citeseer.ist.psu.edu/spenke01visualization.html](http://citeseer.ist.psu.edu/spenke01visualization.html)
- [Spenke und Beilken 1999] SPENKE, M.; BEILKEN, C.: *Visual, Interactive Data Mining with InfoZoom - the Financial Data Set*. [citeseer.ist.psu.edu/474057.html](http://citeseer.ist.psu.edu/474057.html). Version: September 1999. – Contribution to the 'Discovery Challenge' at the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases
- [Stalling et al. 1999] STALLING, Detlev ; SEEBASS, M. ; ZACHOW, Stefan: Mehrschichtige Oberflächenmodelle zur computergestützten Planung in der Chirurgie. In: *Bildverarbeitung für die Medizin (BVM'99)* Bd. 9, 1999, S. 203–207
- [Stalling et al. 1998] STALLING, Detlev ; ZÖCKLER, Malte ; HEGE, Hans-Christian: Segmentation of 3D Medical Images with Subvoxel Accuracy. In: *International Congress on Computer Assisted Radiology and Surgery (CARS'98)*, 1998, S. 137–142
- [Stokely und Wu 1992] STOKELY, Ernest M. ; WU, Shang Y.: Surface parametrization and curvature measurement of arbitrary 3-d objects: Five practical methods. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 14 (1992), Nr. 8, S. 833–840
- [Strauß et al. 2006] STRAUSS, Gero ; HERTEL, Ilka ; DORNHEIM, Jana ; CORDES, Jeanette ; BURGERT, Oliver et al.: Dreidimensionale Darstellung von CT-Datensätzen des Halses für die chirurgische Planung: Eine Machbarkeitsstudie. In: *Laryngo-, Rhino-, Otologie* 85 (2006), Nr. 10, S. 746–754
- [Strauss und Carey 1992] STRAUSS, Paul S. ; CAREY, Rikk: An object-oriented 3D graphics toolkit. In: *ACM SIGGRAPH Computer Graphics* 26 (1992), Nr. 2, S. 341–349
- [Structural Informatics Group 2007] STRUCTURAL INFORMATICS GROUP: *Digital Anatomist Project: Interactive Atlases*. University of Washington : <http://www9.biostr.washington.edu/da.html>, 2007. – [stand 18.06.2007]
- [SumTotal 2007] SUMTOTAL: *ToolBook*. <http://www.toolbook.com/>, 2007. – [letzter Besuch 12.12.2007]
- [Surgical Planning Lab 2007] SURGICAL PLANNING LAB: *SPL/NSL Anatomy Browser*. [http://groups.csail.mit.edu/vision/medical-vision/anatomy\\_browser/models/abdomen.html](http://groups.csail.mit.edu/vision/medical-vision/anatomy_browser/models/abdomen.html), 2007. – [letzter Besuch 19.07.2007]
- [Suzuki et al. 2007] SUZUKI, S. ; ETO k. ; HATTORI, A. ; YANAGA, K. ; SUZUKI, N.: Surgery simulation using patient-specific models for laparoscopic colectomy. In: *Studies in Health Technology and Informatics* 125 (2007), S. 464–466

- [Tang und Byrne 2007] TANG, Michael; BYRNE, Roxanne: Regular versus Online versus Blended: A Qualitative Description of the Advantages of the Electronic Modes and a Quantitative Evaluation. In: *International Journal on E-Learning* 6 (2007), Nr. 2, S. 257–266
- [Taubin 1995a] TAUBIN, Gabriel: Estimating The Tensor Of Curvature Of A Surface From A Polyhedral Approximation. In: *International Conference on Computer Vision (ICCV'95)*, 1995, S. 902–908
- [Taubin 1995b] TAUBIN, Gabriel: A Signal Processing Approach to Fair Surface Design. In: *SIGGRAPH*, 1995, S. 351–358
- [Taubin 2000] TAUBIN, Gabriel: Geometric Signal Processing on Polygonal Meshes. In: *STAR Proceedings of Eurographics*, 2000, S. 81–96
- [Taubin et al. 1996] TAUBIN, Gabriel; ZHANG, Tong; GOLUB, Gene H.: Optimal surface smoothing as filter design. In: *European Conference on Computer Vision (ECCV'96)*, Springer-Verlag, 1996, S. 283–292
- [Theisel 2002] THEISEL, Holger: Exact Isosurfaces for Marching Cubes. In: *Computer Graphics Forum* 21 (2002), Nr. 1, S. 19–31
- [Tiede 1999] TIEDE, Ulf: *Realistische 3D-Visualisierung multiattribuierter und multiparametrischer Volumendaten*, Universität Hamburg, Fachbereich Informatik, Diss., 1999
- [Tiede et al. 2002] TIEDE, Ulf; STERNBERG-GOSPOS, Norman von; STEINER, Paul; HÖHNE, Karl H.: Virtual Endoscopy Using Cubic QuickTime-VR Panorama Views. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI'02)*, Springer, 2002, S. 186–192
- [Tomandl et al. 2003] TOMANDL, Bernd F.; HASTREITER, Peter; ISERHARDT-BAUER, Sabine; KÖSTNER, Niels C.; SCHEMPERSHOFE, Miriam; HUK, Walter J.; ERTL, Thomas; STRAUSS, Christian; ROMSTOCK, Johann: Standardized evaluation of CT angiography with remote generation of 3D video sequences for the detection of intracranial aneurysms. In: *Radiographics* 23 (2003), Nr. 2, S. e12
- [Tomasi und Manduchi 1998] TOMASI, C.; MANDUCHI, R.: Bilateral Filtering for Gray and Color Images. In: *International Conference on Computer Vision (ICCV'98)*, 1998, S. 839–846
- [Turk und O'Brien 2002] TURK, Greg; O'BRIEN, James F.: Modelling with implicit surfaces that interpolate. In: *ACM Transactions on Graphics* 21 (2002), Nr. 4, S. 855–873
- [Tyrrell et al. 2007] TYRRELL, James A.; TOMASO, Emmanuelle di; FUJA, Daniel; TONG, Ricky; KOZAK, Kevin; JAIN, Rakesh K.; ROYSAM, Badrinath: Robust 3-D Modeling of Vasculature Imagery Using Superellipsoids. In: *IEEE Transactions on Medical Imaging* 26 (2007), Nr. 2, S. 223–237
- [Šramek und Kaufman 1998] ŠRAMEK, Miloš; KAUFMAN, Arie: Object Voxelization by Filtering. In: *IEEE Symposium on Volume Visualization (VolVis'98)*, ACM Press, 1998, S. 111–118

- [Viola et al. 2006] VIOLA, Ivan; FEIXAS, Miquel; SBERT, Mateu ; GRÖLLER, Eduard: Importance-Driven Focus of Attention. In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), Nr. 5, S. 933–940
- [Vis-Research Group 2007] VIS-RESEARCH GROUP: *Comparison of Mesh Smoothing Algorithms and Strategies for Medical Surface Models*. <http://www.isg.cs.uni-magdeburg.de/cv/projects/LST/smoothing>, 2007. – Fakultät für Informatik, O.-v.-Guericke Universität Magdeburg – [stand 01.11.2007]
- [Vollmer et al. 1999] VOLLMER, J. ; MENCEL, R. ; MÜLLER, H.: Improved Laplacian Smoothing of Noisy Surface Meshes. In: *Computer Graphics Forum* 18 (1999), Nr. 3, S. 131–138
- [Ware und Rose 1999] WARE, Colin ; ROSE, Jeff: Rotating virtual objects with real handles. In: *ACM Transactions on Computer-Human Interaction* 6 (1999), Nr. 2, S. 162–180
- [Whitaker 2000] WHITAKER, Ross T.: Reducing aliasing artifacts in iso-surfaces of binary volumes. In: *IEEE Symposium on Volume Visualization (VolVis'00)*, ACM Press, 2000, S. 23–32
- [Wikimedia Foundation Inc. 2005] WIKIMEDIA FOUNDATION INC.: *Wikipedia, the free encyclopedia*. <http://en.wikipedia.org/wiki/Humerus>, 2005. – [letzter Besuch 19.09.2005]
- [Wilhelms und Gelder 1992] WILHELMS, Jane ; GELDER, Allen V.: Octrees for faster isosurface generation. In: *ACM Transactions on Graphics* 11 (1992), Nr. 3, S. 201–227
- [Xia et al. 2007] XIA, Xu ; IRANI, Pourang ; WANG, Jing: Evaluation of Guiard's Theory of Bimanual Control for Navigation and Selection. In: *Human-Computer Interaction (HCI'07)*, Springer, 2007, S. 368–377
- [Yagou et al. 2002a] YAGOU, Hirokazu ; BELYAEV, Alexander ; WEI, Daming: Mesh Median Filtering for Smoothing 3-D Polygonal Surfaces. In: *Cyber Worlds (CW'02)*, 2002, S. 488–495
- [Yagou et al. 2002b] YAGOU, Hirokazu ; OHTAKE, Yutaka ; BELYAEV, Alexander: Mesh Smoothing via Mean and Median Filtering Applied to Face Normals. In: *Geometric Modeling and Processing (GMP'02)*, IEEE Computer Society, 2002, S. 124–131
- [Yagou et al. 2003] YAGOU, Hirokazu ; OHTAKE, Yutaka ; BELYAEV, Alexander: Mesh Denoising via Iterative Alpha-Trimming and Nonlinear Diffusion of Normals with Automatic Thresholding. In: *Computer Graphics International*, 2003, S. 28–33
- [Yoshizawa et al. 2006] YOSHIKAWA, Shin ; BELYAEV, Alexander G. ; SEIDEL, Hans-Peter: Smoothing by Example: Mesh Denoising by Averaging with Similarity-based Weights. In: *International Conference on Shape Modeling and Applications (SMI'06)*, 2006, S. 38–44
- [Zachow 2005] ZACHOW, Stefan: *Computergestützte 3D Osteotomieplanung in der Mund-Kiefer-Gesichtschirurgie unter Berücksichtigung der räumlichen Weichgewebeanordnung.*, Berlin, Technische Universität, Diss., 2005

- [Zachow et al. 2006] ZACHOW, Stefan ; HEGE, Hans-Christian ; DEUFLHARD, Peter: Computer Assisted Planning in Cranio-Maxillofacial Surgery. In: *Journal of Computing and Information Technology - Special Issue on Computer-Based Craniofacial Modelling and Reconstruction* 14 (2006), Nr. 1, S. 53–64
- [Zhai 1998] ZHAI, Shumin: User performance in relation to 3D input device design. In: *ACM SIGGRAPH Computer Graphics* 30 (1998), Nr. 4, S. 50–54
- [Zhao und Osher 2002] ZHAO, Hongkai ; OSHER, Stanley: Visualization, Analysis and Shape Reconstruction of Unorganized Data Sets. In: OSHER, Stanley (Hrsg.) ; PARAGIOS, Nikos (Hrsg.): *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer, 2002
- [Zhihong et al. 2006] ZHIHONG, Mao ; LIZHUANG, Ma ; MINGXI, Zhao ; ZHONG, Li: Feature-preserving mesh denoising based on contextual discontinuities. In: *Journal of Zhejiang University - Science A* 7 (2006), Nr. 9, S. 1603–1608
- [Zucker und Hummel 1981] ZUCKER, Steven W. ; HUMMEL, Robert A.: A Three-Dimensional Edge Operator. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 3 (1981), Nr. 3, S. 324–331





# Ergebnisdiagramme der Evaluation der Glättungsverfahren

---

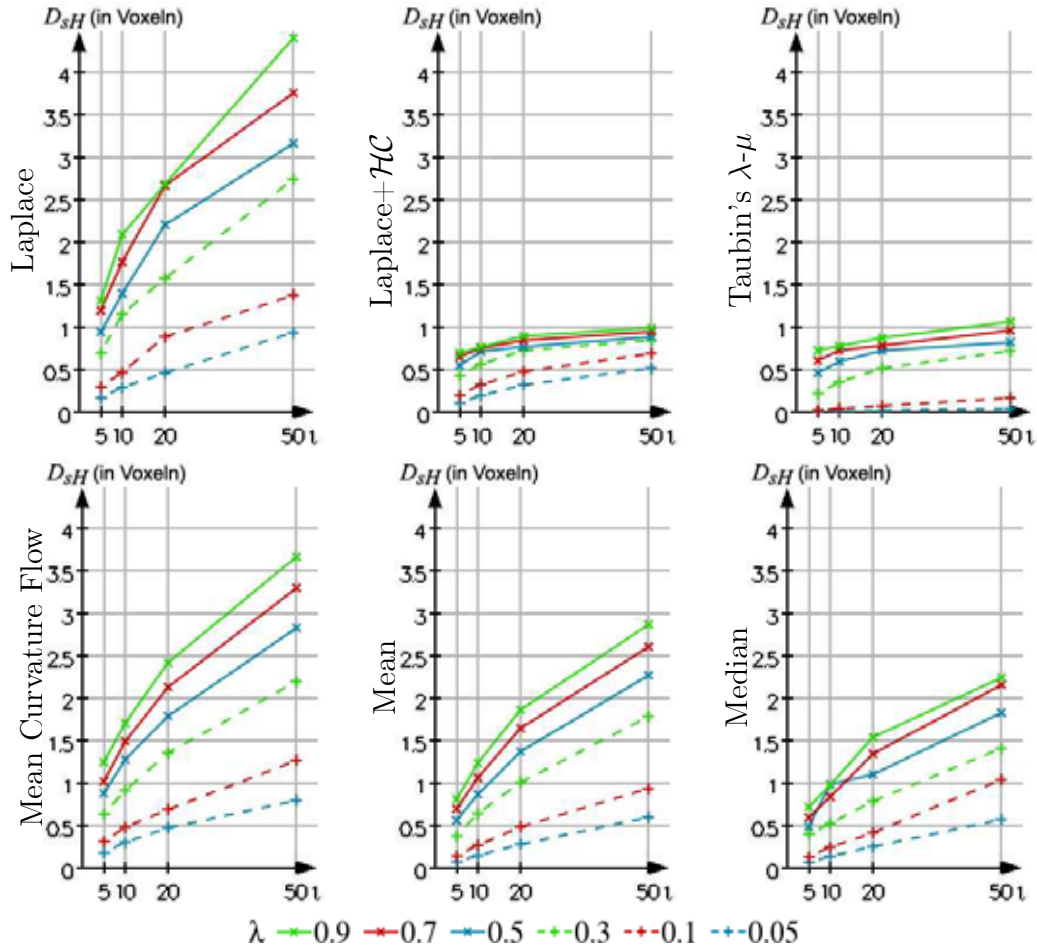
Im Folgenden werden alle Ergebnisdiagramme aus der durchgeführten Evaluation der Glättungsverfahren als Ergänzung zu Kapitel 5 aufgeführt<sup>1</sup>.

---

<sup>1</sup>Dieser Anhang kann farbig in elektronischer Form über:  
<http://www.isg.cs.uni-magdeburg.de/~rbade/dissertation> bezogen werden.

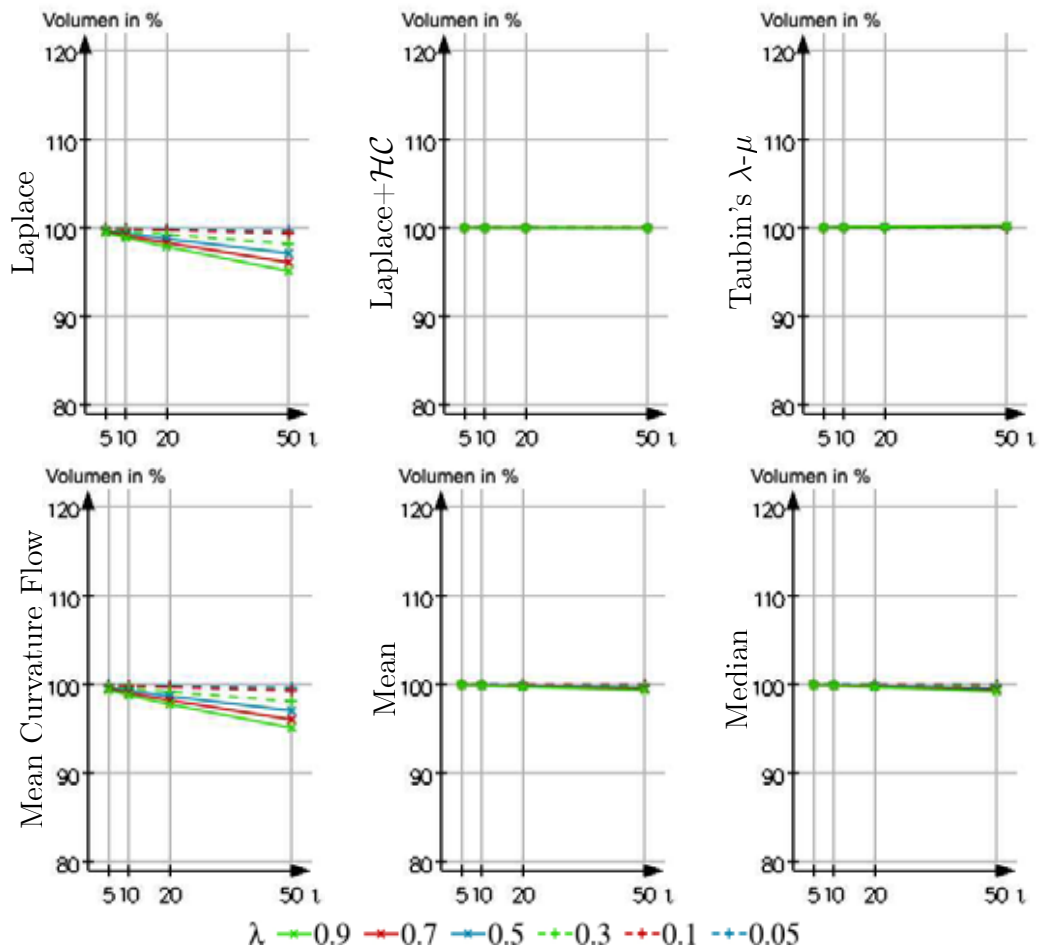
## A.1 Oberflächenmodell der Leber

### A.1.1 Symmetrische Hausdorff-Distanz – Lebermodell



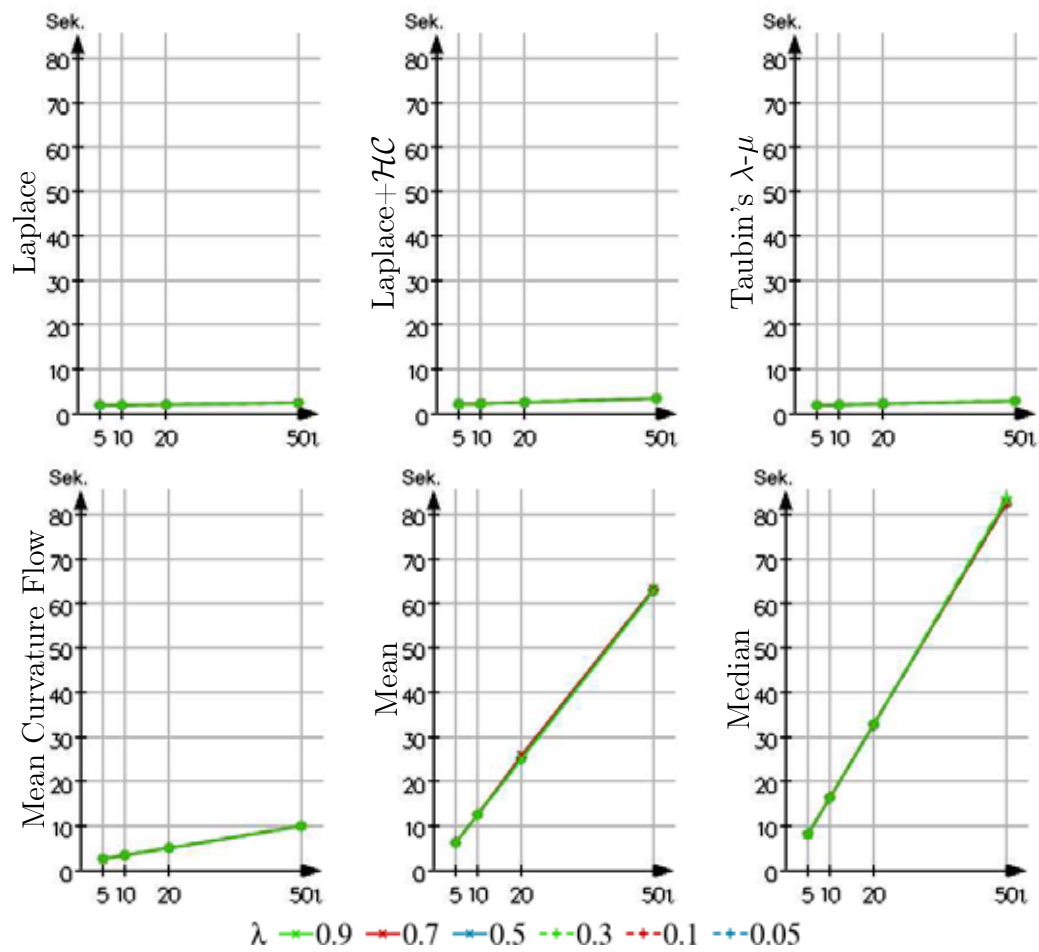
**Abbildung A.1:** Symmetrische Hausdorff-Distanz ( $D_{sH}$ ) der Glättungsergebnisse der verschiedenen Filter zum initialen Oberflächenmodell der Leber (in Voxelkoordinaten) bei unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .

## A.1.2 Volumenerhalt – Lebermodell



**Abbildung A.2:** Volumenerhalt ( $V_{\%}$  in % des Originals) der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells der Leber mit unterschiedlicher Iterationsanzahl  $t$  und Wichtung  $\lambda$ .

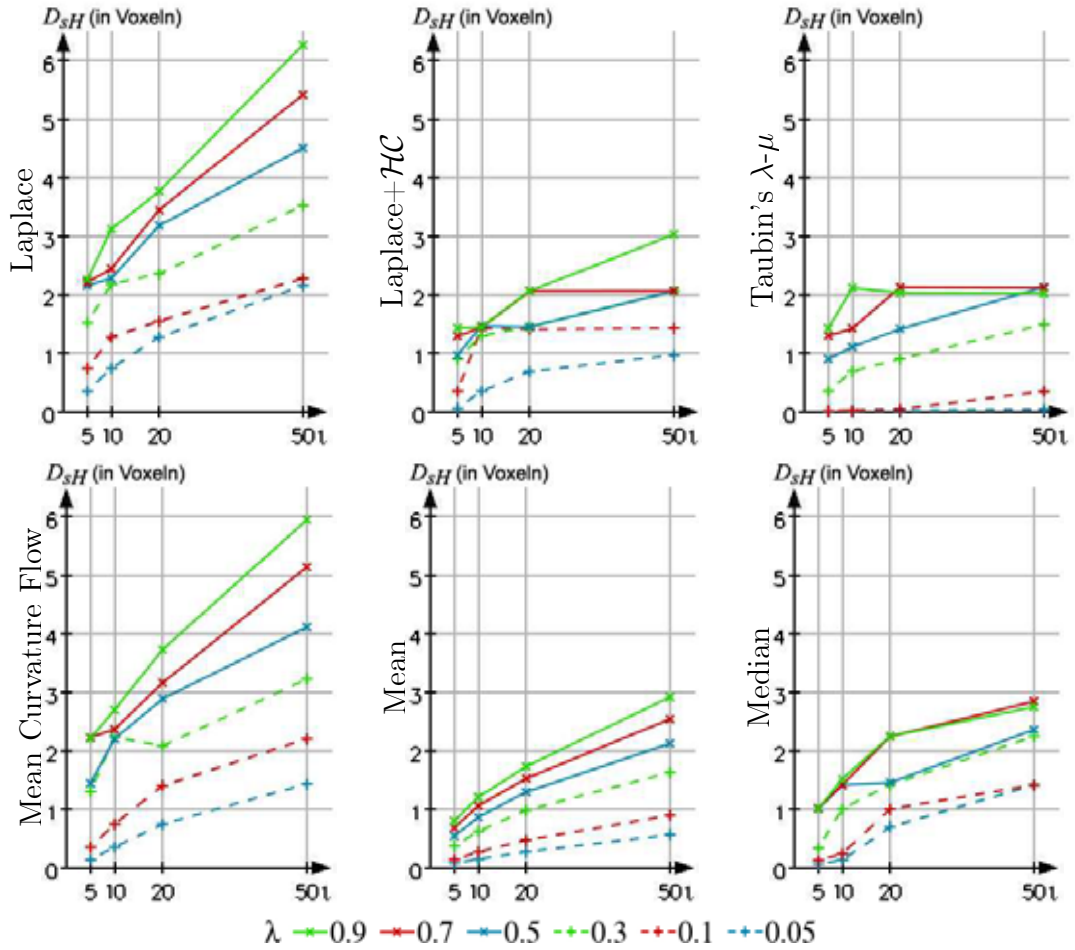
### A.1.3 Rechenzeit – Lebermodell



**Abbildung A.3:** Rechenzeit der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells der Leber mit unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .

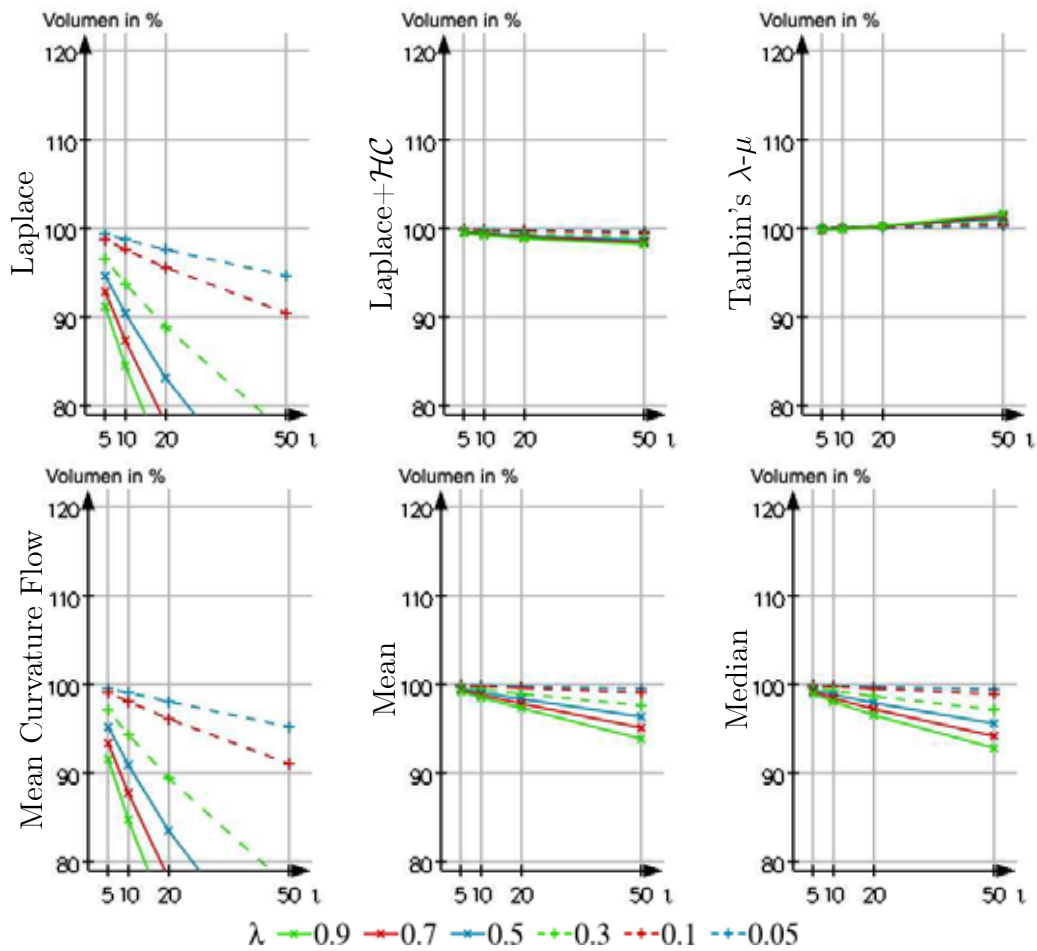
## A.2 Oberflächenmodell des Lymphknotens

### A.2.1 Symmetrische Hausdorff-Distanz – Lymphknotenmodell



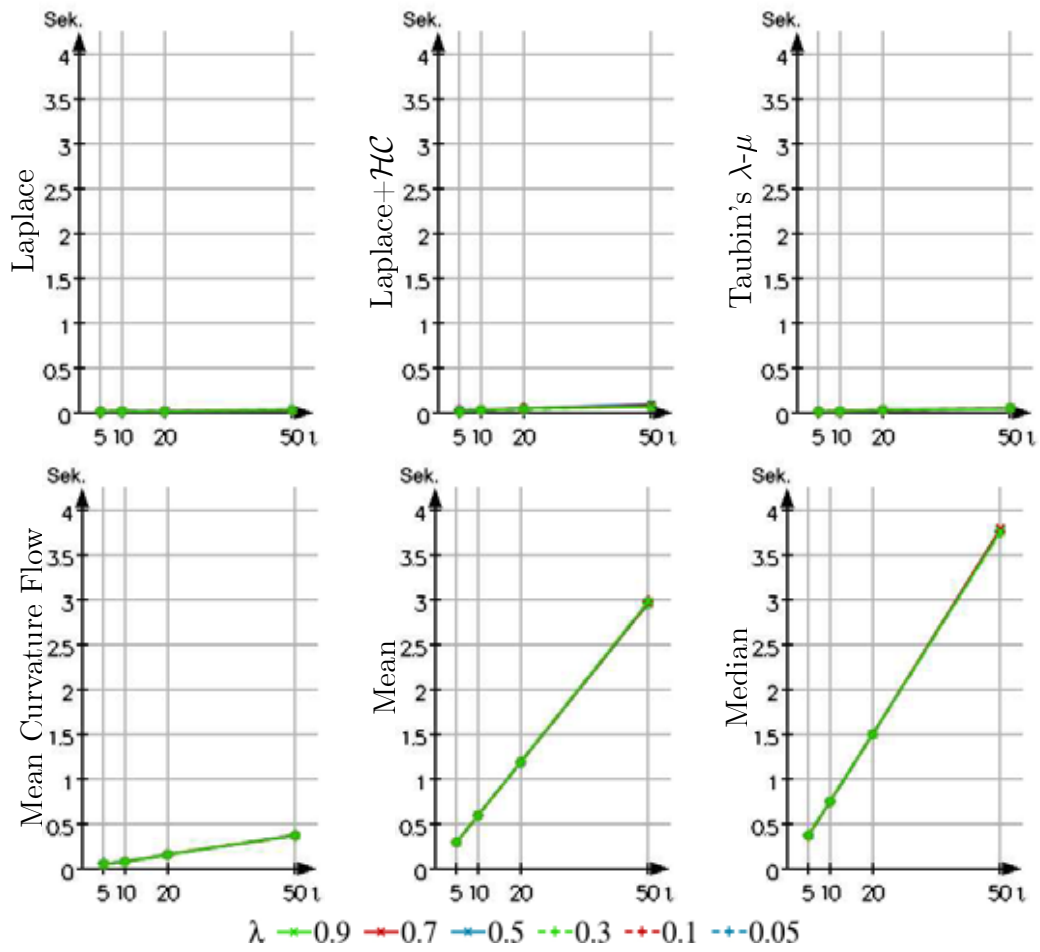
**Abbildung A.4:** Symmetrische Hausdorff-Distanz ( $D_{sH}$ ) der Glättungsergebnisse der verschiedenen Filter zum initialen Oberflächenmodell des Lymphknotens (in Voxelkoordinaten) bei unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .

### A.2.2 Volumenerhalt – Lymphknotenmodell



**Abbildung A.5:** Volumenerhalt ( $V_{\%}$  in % des Originals) der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells des Lymphknotens mit unterschiedlicher Iterationsanzahl  $t$  und Wichtung  $\lambda$ .

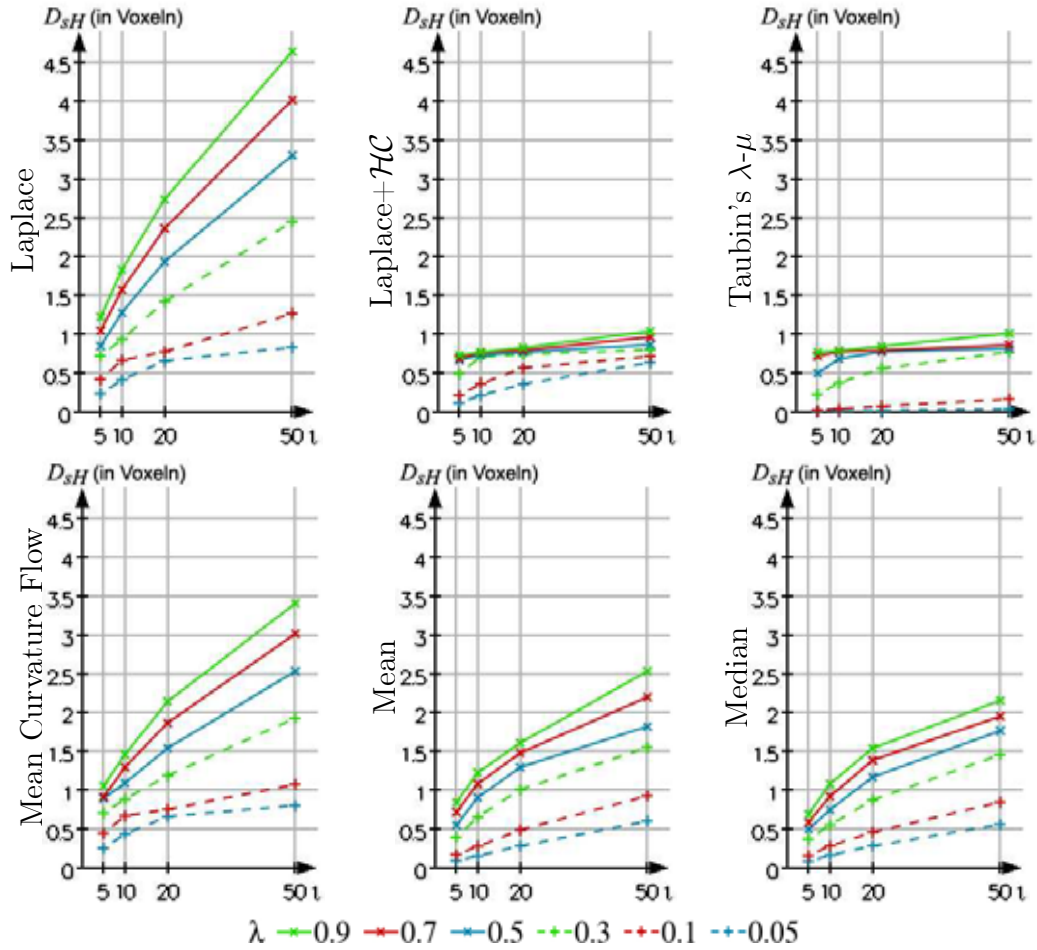
## A.2.3 Rechenzeit – Lymphknotenmodell



**Abbildung A.6:** Rechenzeit der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells des Lymphknotens mit unterschiedlicher Iterationsanzahl  $l$  und Wichtung  $\lambda$ .

## A.3 Oberflächenmodell des Beckens

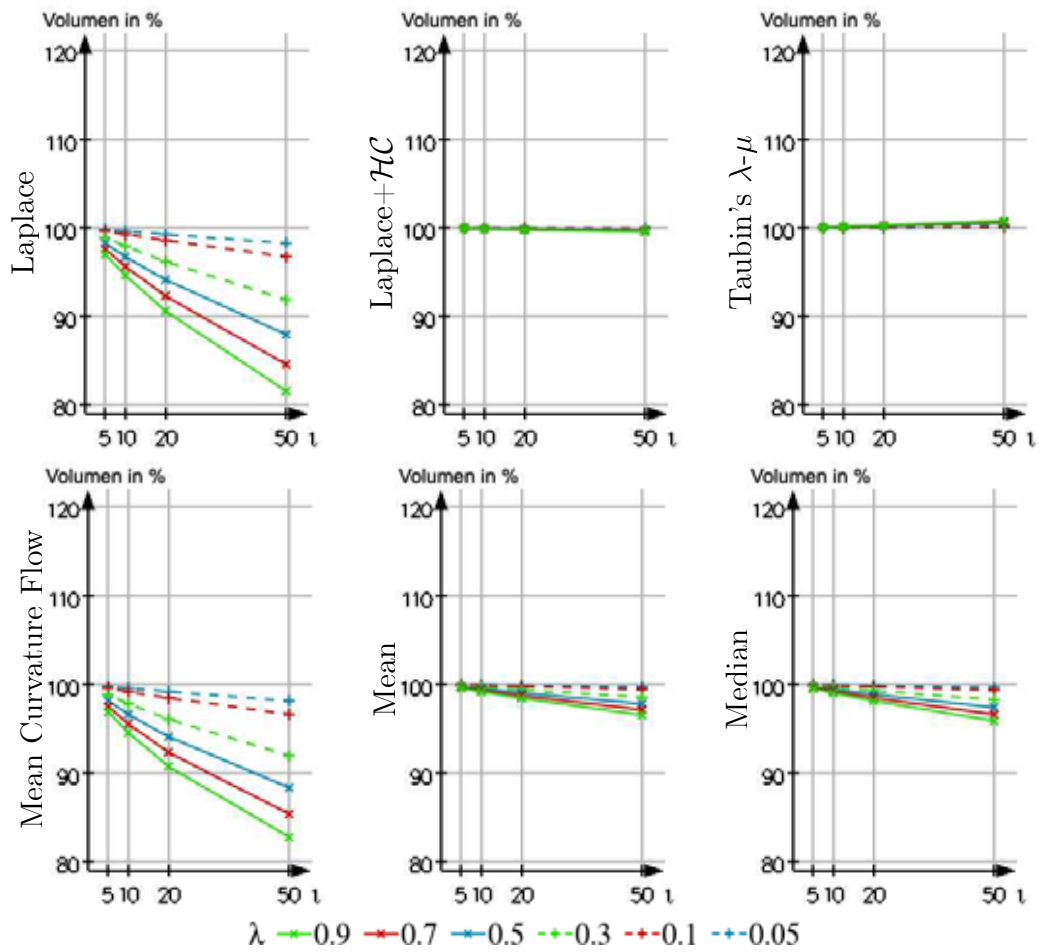
### A.3.1 Symmetrische Hausdorff-Distanz – Beckenmodell



**Abbildung A.7:** Symmetrische Hausdorff-Distanz ( $D_{sH}$ ) der Glättungsergebnisse der verschiedenen Filter zum initialen Oberflächenmodell des Beckens (in Voxelkoordinaten) bei unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .

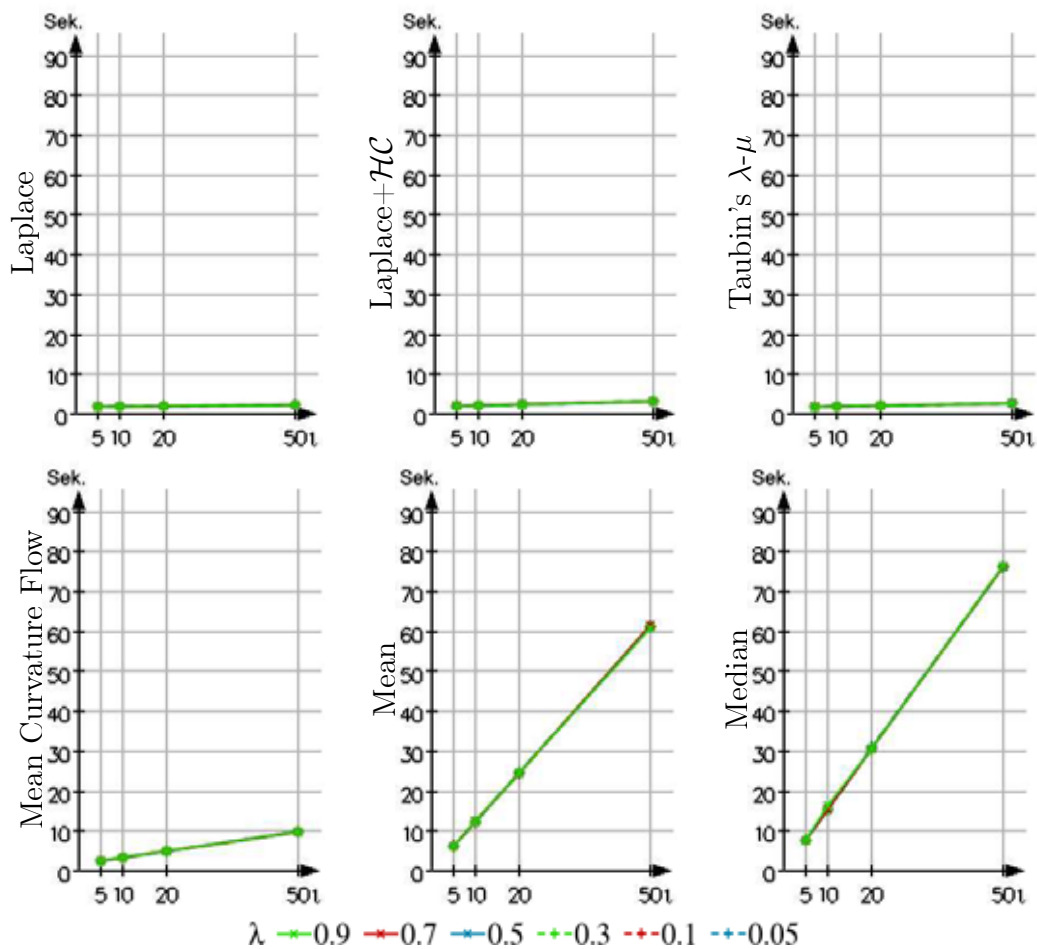


## A.3.2 Volumenerhalt – Beckenmodell



**Abbildung A.8:** Volumenerhalt ( $V_{\%}$  in % des Originals) der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells des Beckens mit unterschiedlicher Iterationsanzahl  $t$  und Wichtung  $\lambda$ .

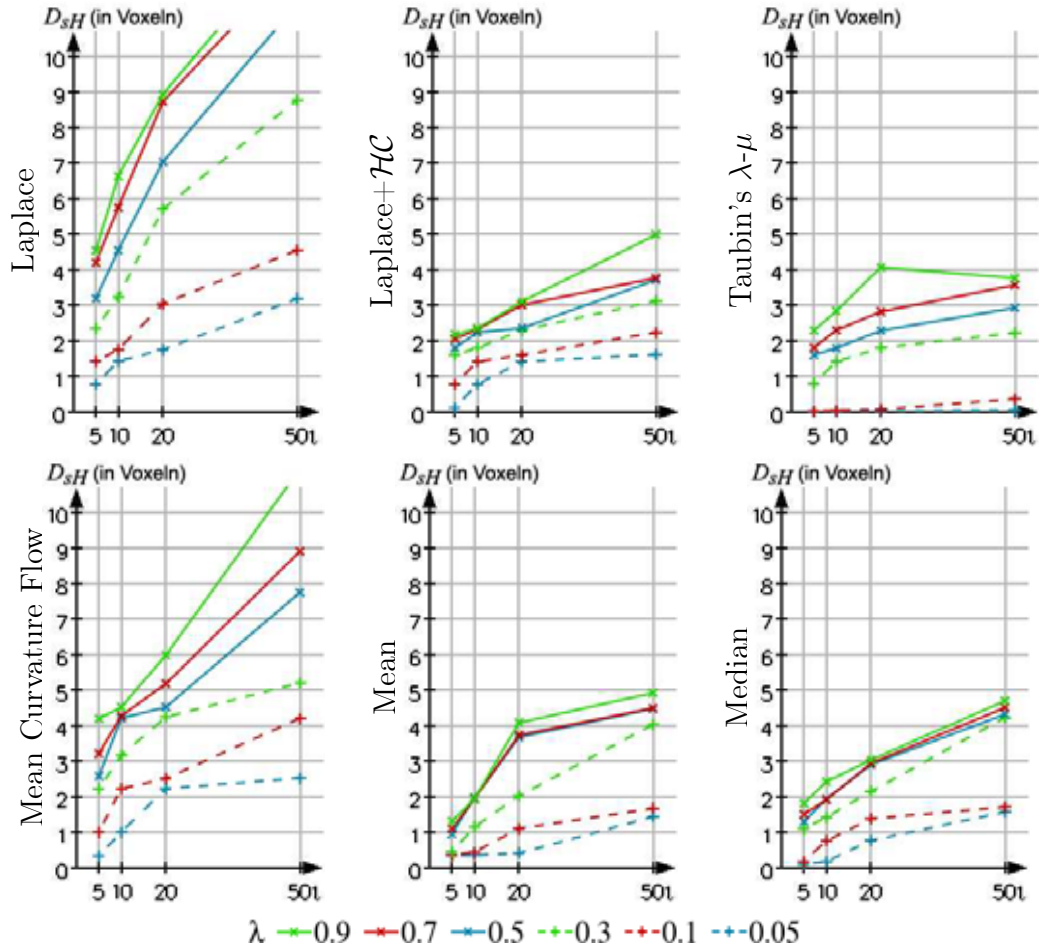
### A.3.3 Rechenzeit – Beckenmodell



**Abbildung A.9:** Rechenzeit der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells des Beckens mit unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .

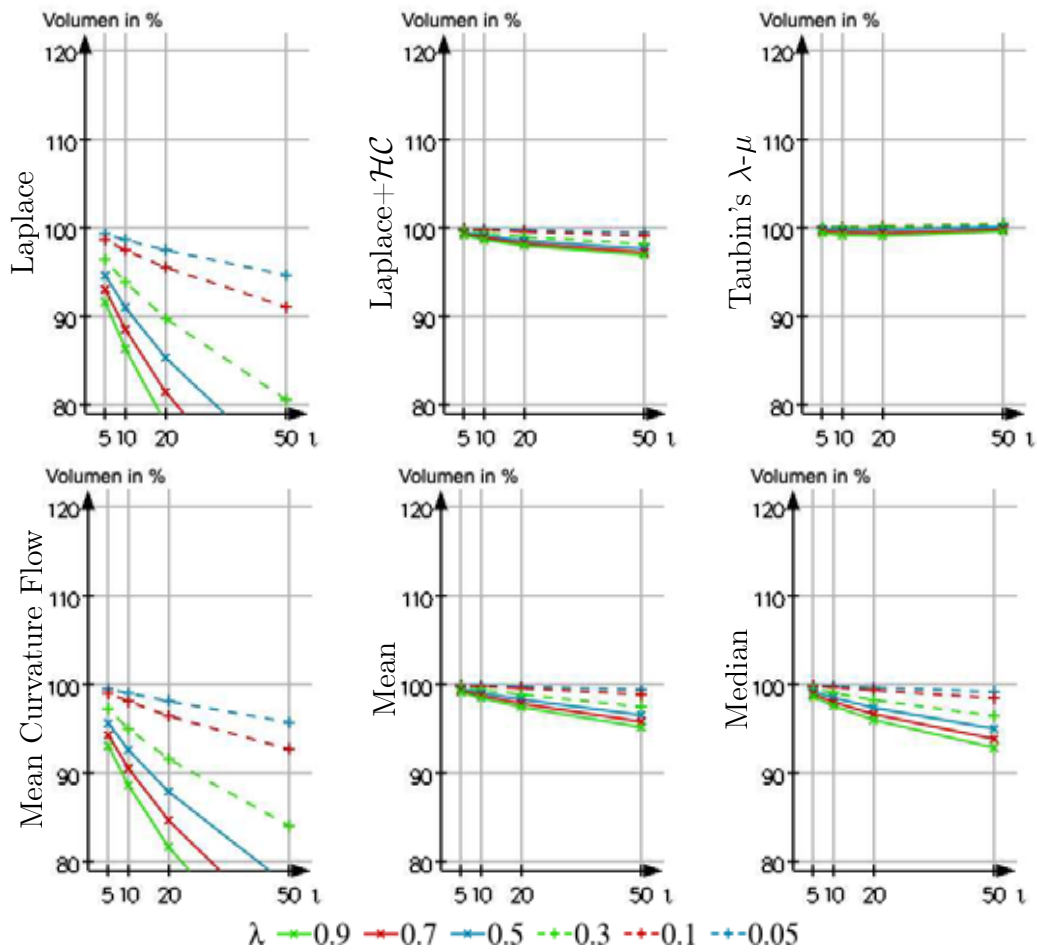
## A.4 Oberflächenmodell des Halsmuskels

### A.4.1 Symmetrische Hausdorff-Distanz – Halsmuskelmodell



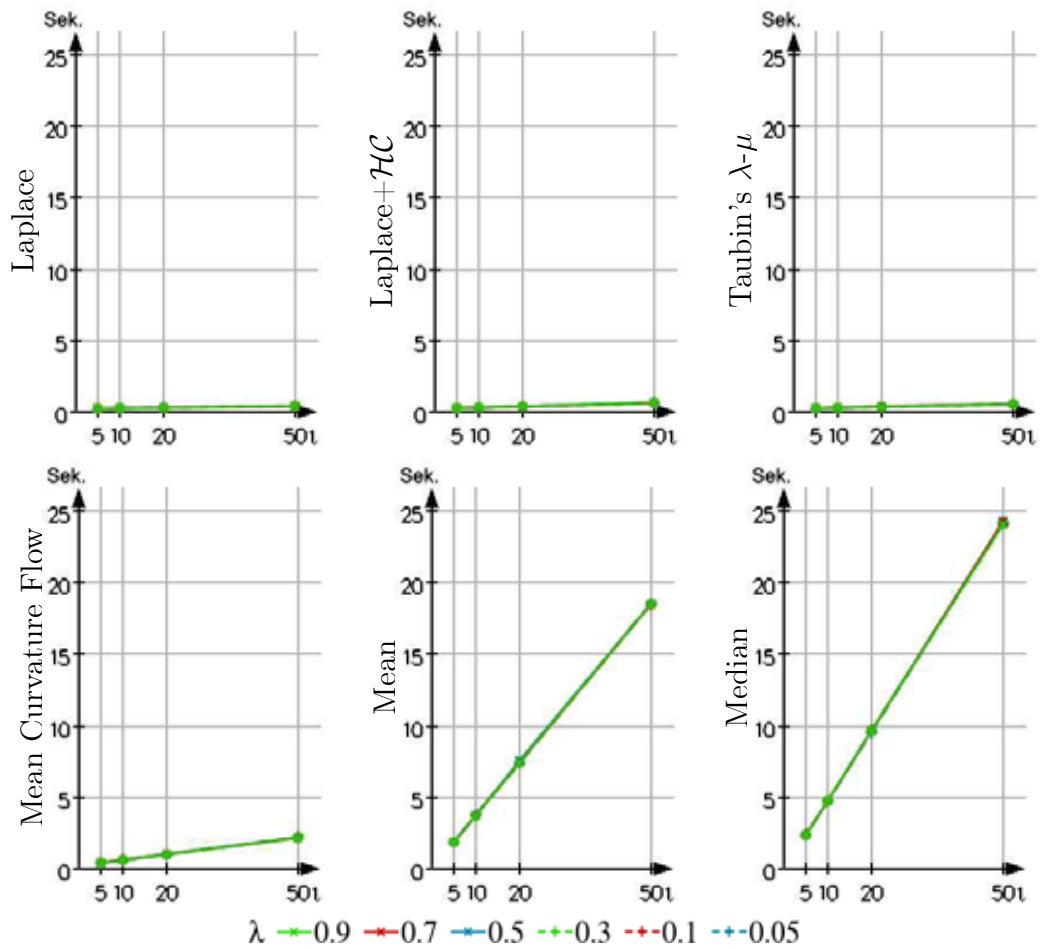
**Abbildung A.10:** Symmetrische Hausdorff-Distanz ( $D_{sH}$ ) der Glättungsergebnisse der verschiedenen Filter zum initialen Oberflächenmodell des Halsmuskels (in Voxelkoordinaten) bei unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .

### A.4.2 Volumenerhalt – Halsmuskelsmodell



**Abbildung A.11:** Volumenerhalt ( $V_{\%}$  in % des Originals) der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells des Halsmuskels mit unterschiedlicher Iterationsanzahl  $t$  und Wichtung  $\lambda$ .

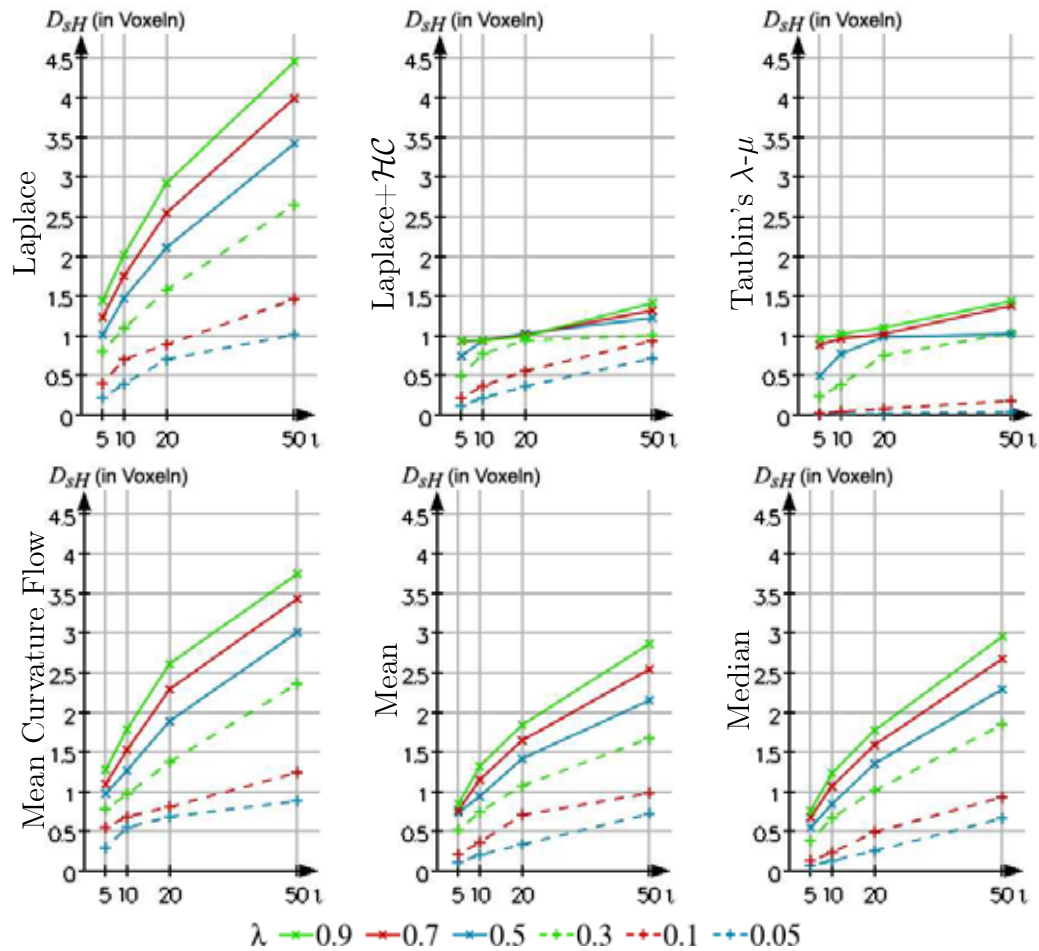
## A.4.3 Rechenzeit – Halsmuskellmodell



**Abbildung A.12:** Rechenzeit der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells des Halsmuskels mit unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .

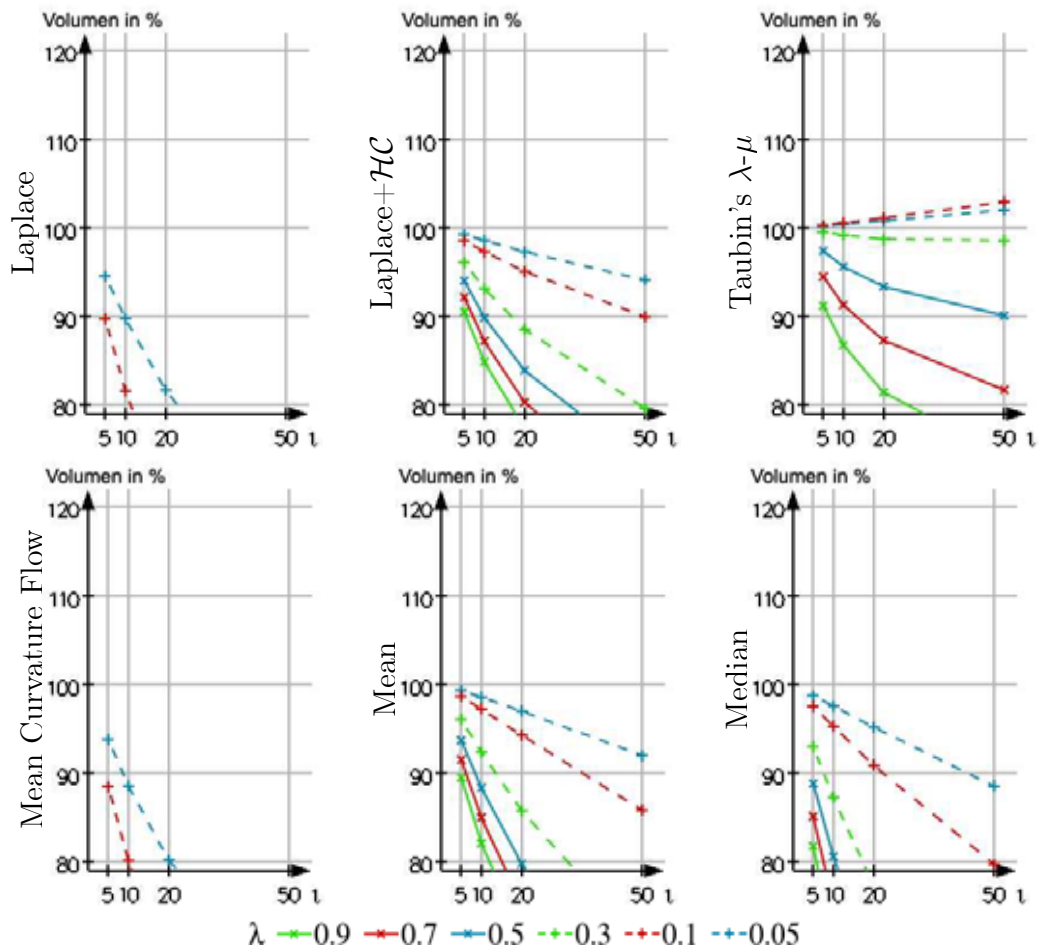
## A.5 Oberflächenmodell des Gefäßbaumes

### A.5.1 Symmetrische Hausdorff-Distanz – Gefäßbaummodell



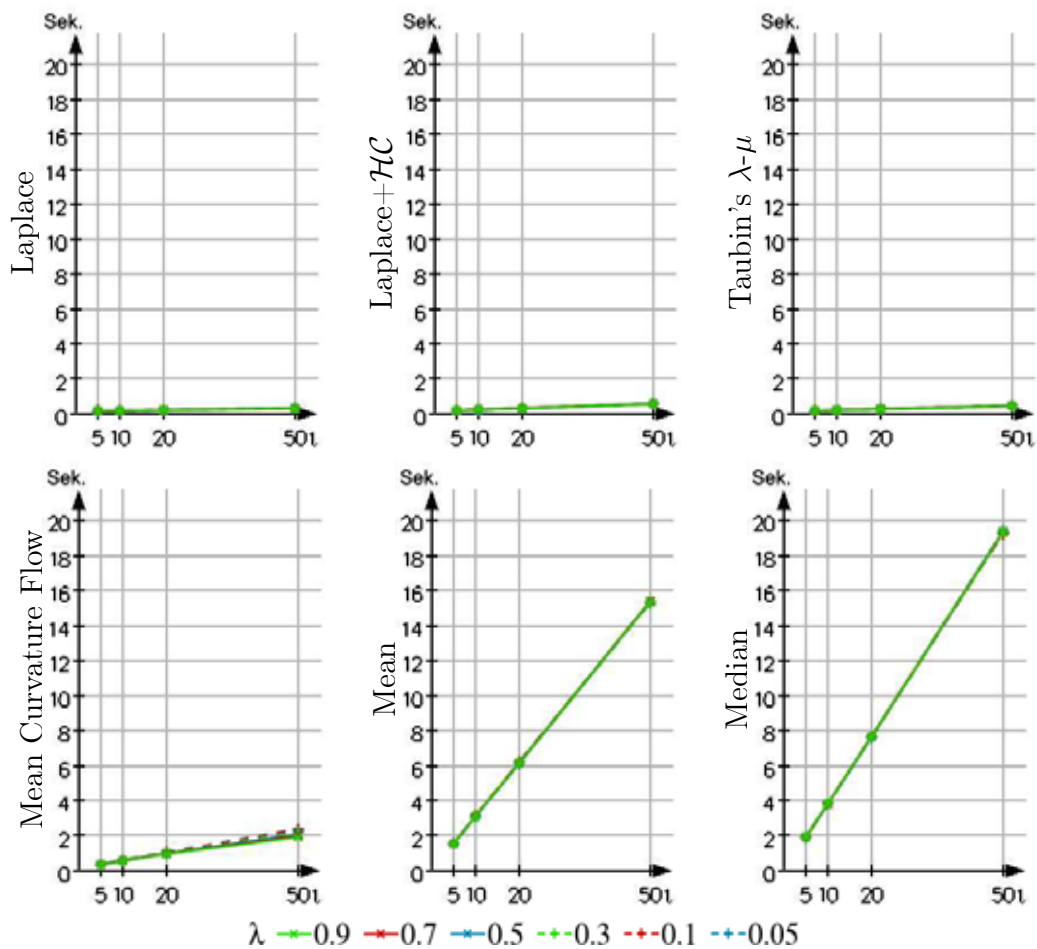
**Abbildung A.13:** Symmetrische Hausdorff-Distanz ( $D_{sH}$ ) der Glättungsergebnisse der verschiedenen Filter zum initialen Oberflächenmodell des Gefäßbaumes (in Voxelkoordinaten) bei unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .

## A.5.2 Volumenerhalt – Gefäßbaummodell



**Abbildung A.14:** Volumenerhalt ( $V_{\%}$  in % des Originals) der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells des Gefäßbaumes mit unterschiedlicher Iterationsanzahl  $t$  und Wichtung  $\lambda$ .

## A.5.3 Rechenzeit – Gefäßbaummodell

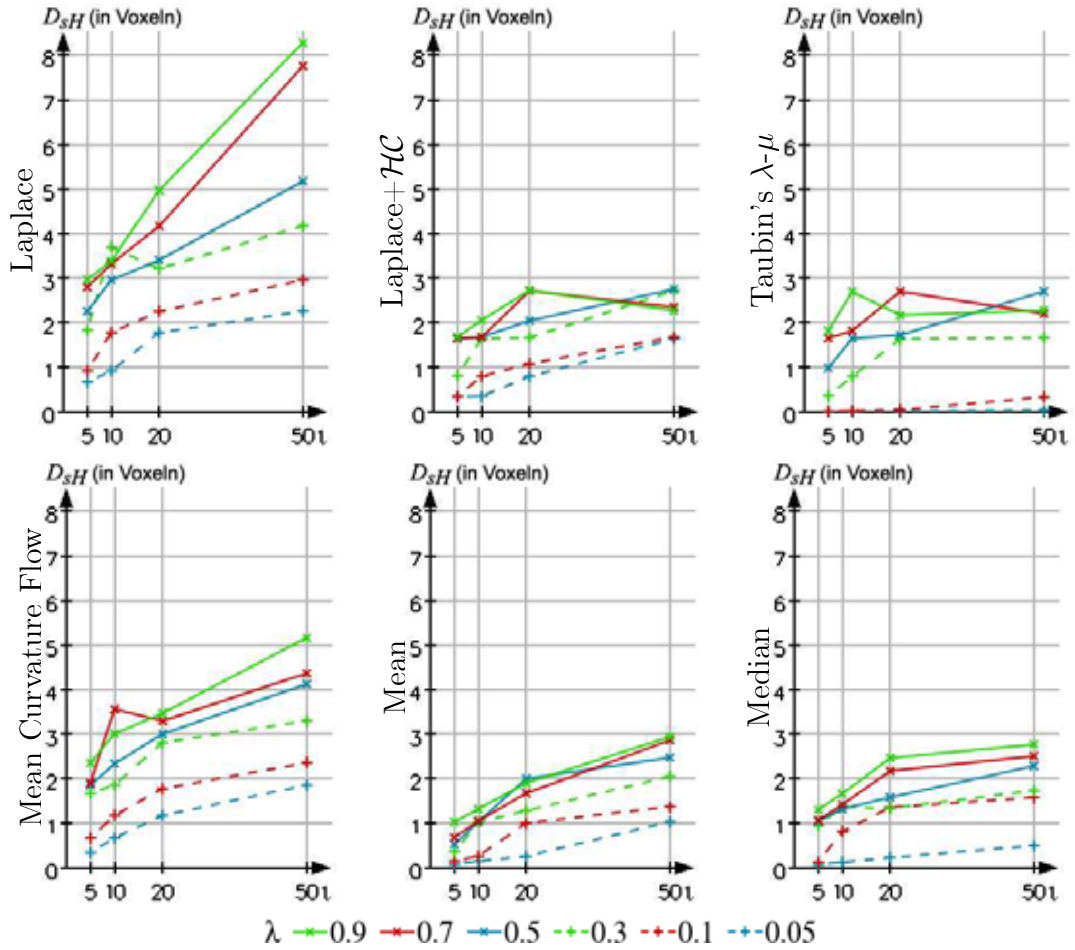


**Abbildung A.15:** Rechenzeit der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells des Gefäßbaumes mit unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .



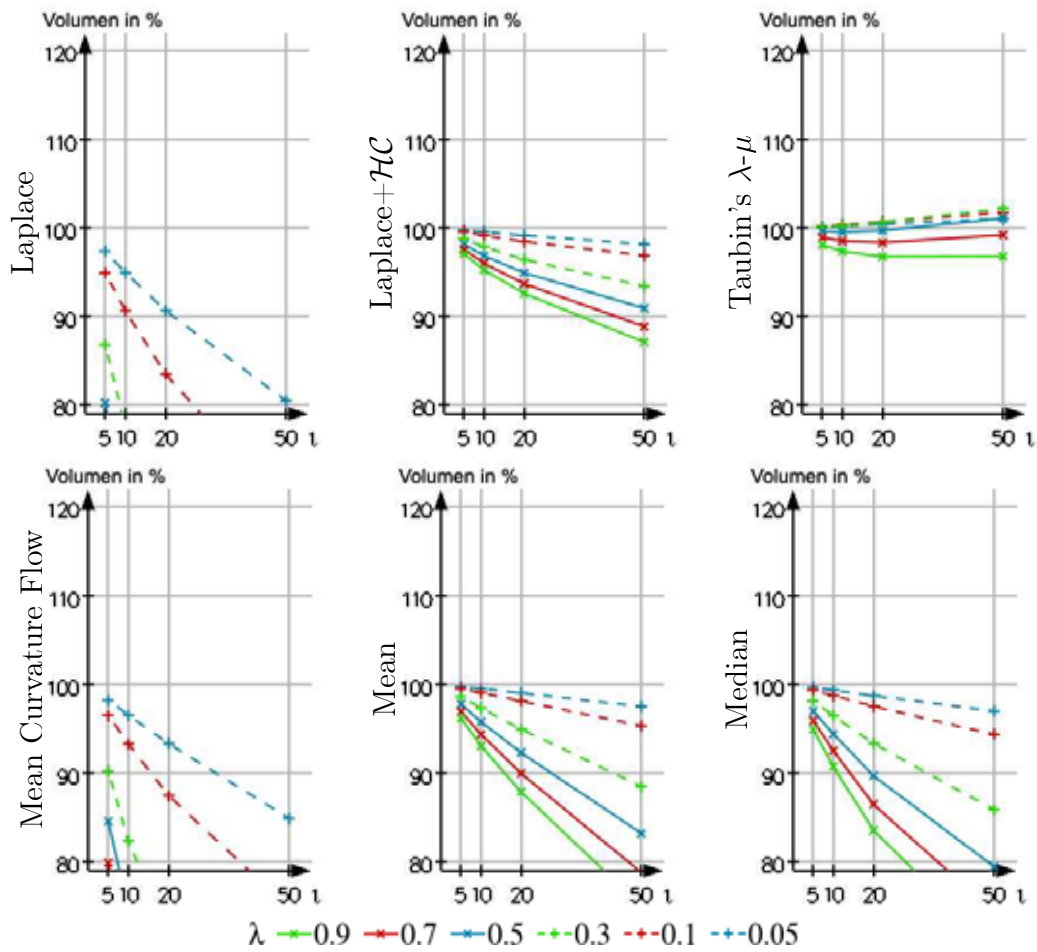
## A.6 Oberflächenmodell der A. carotis

### A.6.1 Symmetrische Hausdorff-Distanz – A. carotis-Modell



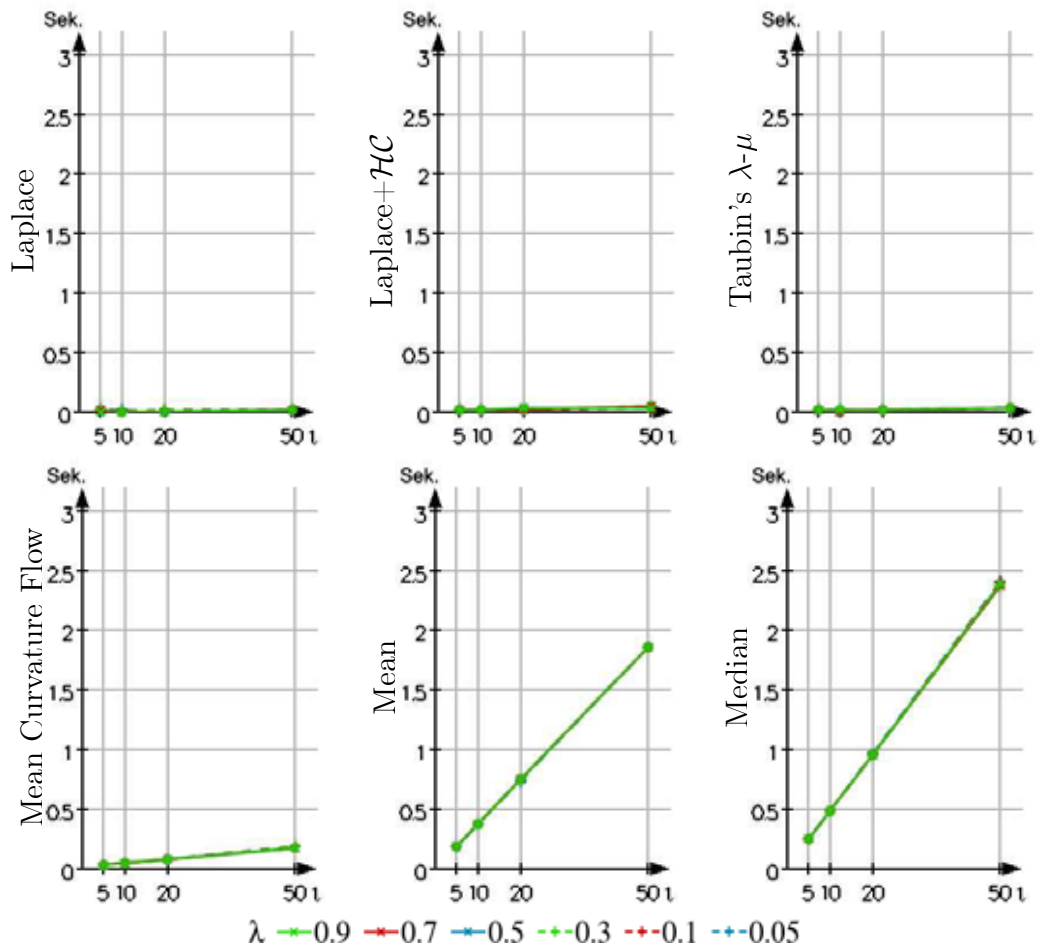
**Abbildung A.16:** Symmetrische Hausdorff-Distanz ( $D_{sH}$ ) der Glättungsergebnisse der verschiedenen Filter zum initialen Oberflächenmodell der A. carotis (in Voxelkoordinaten) bei unterschiedlicher Iterationsanzahl  $\iota$  und Wichtung  $\lambda$ .

## A.6.2 Volumenerhalt – A. carotis-Modell



**Abbildung A.17:** Volumenerhalt ( $V_{\%}$  in % des Originals) der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells der A. carotis mit unterschiedlicher Iterationsanzahl  $t$  und Wichtung  $\lambda$ .

## A.6.3 Rechenzeit – A. carotis-Modell



**Abbildung A.18:** Rechenzeit der verschiedenen Glättungsverfahren beim Glätten des Oberflächenmodells der A. carotis mit unterschiedlicher Iterationsanzahl  $l$  und Wichtung  $\lambda$ .

