

**Otto-von-Guericke-Universität Magdeburg**

Fakultät für Informatik

Institut für Simulation und Graphik



# Diplomarbeit

---

## **Krümmungs- und modellbasierte Schraffierung auf patientenspezifischen, anatomischen Oberflächen**

---

von: Rocco Gasteiger  
geb. am: 07. Oktober 1978  
in: Grimma  
Matrikelnummer: 162790

1. Gutachter: Prof. Dr.-Ing. Bernhard Preim  
2. Gutachter: Prof. Dr.-Ing. Stefan Schlechtweg

Betreuer: Prof. Dr.-Ing. Bernhard Preim  
Dipl.-Ing. Alexandra Baer  
Dipl.-Ing. Christian Tietjen

Zeit der Diplomarbeit: 16. Februar 2007 – 16. August 2007

**Rocco Gasteiger :**

Matrikelnummer: 162790

*Krümmungs- und modellbasierte Schraffierung  
auf patientenspezifischen, anatomischen Oberflächen*

Diplomarbeit, Otto-von-Guericke-Universität Magdeburg, 2007.

©Rocco Gasteiger

## **Selbständigkeitserklärung**

Hiermit erkläre ich, Rocco Gasteiger (Matrikel-Nr. 162790), dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen angefertigt habe.

Rocco Gasteiger  
Magdeburg, den 16. August 2007



## Danksagung

*Wenn der HERR nicht das Haus baut, so arbeiten umsonst, die daran bauen. Wenn der HERR nicht die Stadt behütet, so wacht der Wächter umsonst. (Die Bibel, Psalm 127,1)*

Wem sage ich zuerst Dank, wenn nicht Dir mein Herr Jesus Christus. Ohne Dich hätte ich diese Arbeit niemals schreiben können und der obige Spruch aus Deinem Wort begleitete mich während dieser Zeit. Ich danke Dir von ganzem Herzen für das Erreichte, verbunden mit der dafür geschenkten Motivation, Weisheit, Kraft, Gesundheit sowie Zeit. Du hast mich in hellen und dunklen Tagen durch mein Studium begleitet, geformt und auf wunderbare Weise bis hierher geführt. Zu Deiner Ehre soll Dir daher diese Arbeit reichen und gewidmet sein. In tiefer Verbundenheit und Dankbarkeit, Dein Rocco. Amen.

Weiterhin möchte ich mich aber auch sehr bei meiner Familie bedanken. Durch ihre jahrelange Unterstützung und Ermutigung haben sie mir geholfen dieses Studium zu meistern. Dies erachte ich nicht für selbstverständlich und bin dankbar sowie froh, dass wir diesen Weg gemeinsam gegangen sind. In Gedenken an Oma Hanna, die diesen Weg leider nicht bis zum Ende mitgehen konnte, gilt auch ihr dieser Dank und diese hohe Wertschätzung.

Was wäre diese Arbeit ohne meine Betreuer Prof. Dr. Bernhard Preim, Alexandra Baer sowie Christian Tietjen. Ihnen gilt mein besonderer Dank, da sie mich nicht nur durch ihre starke fachliche Kompetenz unterstützt, sondern auch durch aufmunternde Worte stets motiviert haben. Die konstruktiven Diskussionen um das Thema, halfen mir dabei sehr bei der Orientierung der Arbeit. Ich achte die von ihnen dafür aufgebraachte Zeit, gerade während der Endphase, hoch an. In diesem Zusammenhang möchte ich mich auch bei Mathias Neugebauer bedanken, mit dem ich ebenfalls konstruktive Gespräche über Problemstellungen des Themas führen konnte. Die dabei entstandenen Ideen und Anregungen halfen mir sehr bei der Bearbeitung dieser Arbeit.

Zudem gilt mein Dank auch Olaf Konrad von MEVIS RESEARCH und Steffen Oeltze, die mir beim Umgang mit der MLGRAPH-Bibliothek geholfen haben. Auch die mit Olaf geführten Diskussionen über bestimmte Problemstellungen waren gewinnbringend für die Arbeit. Bei Christian Hansen und meiner Glaubensschwester Kirsten Sieber möchte ich mich ganz herzlich bedanken, dass sie meine Arbeit Korrektur gelesen haben.

Abschließend bedanke ich mich sehr bei meinen Freunden, die für mich gebetet haben und die trotz der wenigen Zeit, die ich für sie hatte, weiterhin mit mir befreundet sind. Auch danke ich besonders meinem „Banknachbarn“ Christian Panzer, mit dem ich die meiste Zeit das Diplomandenlabor teilte. Neben den fachlichen Diskussionen konnten wir uns stets bei einer regelmäßigen Kaffeepause gegenseitig ermutigen sowie mal „abschalten“. Auch danke ich den anderen Diplomanden Stefan Hiller, Roland Pfister, dem Peter sowie Dirk Lehmann für die lockere Atmosphäre im Labor und gegenseitige Ermunterung.

Rocco Gasteiger, 16. August 2007



---

---

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Zielsetzung . . . . .	2
1.2	Gliederung der Arbeit . . . . .	2
<b>2</b>	<b>Schraffuren in medizinischen Illustrationen</b>	<b>5</b>
2.1	Medizinische Illustrationen . . . . .	5
2.2	Schraffur in illustrativen Darstellungen . . . . .	9
2.2.1	Gestaltungsparameter bei Schraffuren . . . . .	9
2.2.2	Oberflächenwahrnehmung mit Schraffuren . . . . .	11
2.2.3	Analyse von Schraffuren in medizinischen Illustrationen . . . . .	15
2.2.4	Zusammenfassung und Schlussfolgerung . . . . .	20
2.3	Computergenerierte Schraffierungsansätze . . . . .	21
2.3.1	Anforderungsanalyse . . . . .	21
2.3.2	Bildbasierter Ansatz . . . . .	23
2.3.3	Objektbasierter Ansatz . . . . .	26
2.3.4	Texturbasierter Ansatz . . . . .	29
2.3.5	Zusammenfassung . . . . .	32
2.4	Texturierung auf 3D-Modellen . . . . .	33
2.4.1	2D-Texture Mapping . . . . .	35
2.4.2	Parametrisierung . . . . .	37
2.5	Zusammenfassung und Schlussfolgerungen . . . . .	41
<b>3</b>	<b>Krümmungsapproximation auf 3D-Modellen</b>	<b>43</b>
3.1	Mathematische Grundlagen . . . . .	43
3.1.1	Krümmung auf einer Kurve . . . . .	43
3.1.2	Krümmung auf einer Fläche . . . . .	44
3.2	Verfahren zur Krümmungsapproximation . . . . .	47

3.2.1	Curve Fitting . . . . .	48
3.2.2	Surface Fitting . . . . .	50
3.3	Zusammenfassung und Schlussfolgerungen . . . . .	54
<b>4</b>	<b>Entwurf für Schraffierung medizinischer Datensätze</b>	<b>57</b>
4.1	Vorbetrachtungen . . . . .	57
4.1.1	Analyse der patientenspezifischen Datensätze . . . . .	57
4.1.2	Beschreibung der Schraffurichtung anatomischer Strukturen . . . . .	59
4.2	Anforderungsanalyse und Übersicht des Entwurfes . . . . .	63
4.2.1	Anforderungsanalyse . . . . .	63
4.2.2	Entwurfsübersicht . . . . .	65
4.3	Generierung des krümmungs- und modellbasierten Vektorfeldes . . . . .	66
4.3.1	Glättung der Oberfläche für die Krümmungsapproximation . . . . .	66
4.3.2	Krümmungsapproximationen der anatomischen Strukturen . . . . .	67
4.3.3	Auswahl der Approximationsverfahren . . . . .	69
4.3.4	Ermittlung der modellbezogenen Vorzugsrichtung . . . . .	73
4.3.5	Kombination von Krümmungsinformationen und Vorzugsrichtung . . . . .	82
4.4	Parametrisierung der Oberfläche . . . . .	85
4.4.1	Erzeugung des Startpatch und Überführung in den Parameterraum . . . . .	86
4.4.2	Abbruchkriterien für ein Patchwachstum . . . . .	91
4.4.3	Hinzufügen weiterer Patches . . . . .	95
4.4.4	Behandlung der Patchübergänge . . . . .	100
4.5	Texturierung . . . . .	107
4.5.1	Integration der Beleuchtung . . . . .	108
4.6	Zusammenfassung . . . . .	108
<b>5</b>	<b>Implementierung</b>	<b>111</b>
5.1	Entwicklungswerkzeuge und verwendete Bibliotheken . . . . .	111
5.1.1	MeVisLab . . . . .	111
5.1.2	Verwendete Bibliotheken . . . . .	112
5.2	Erzeugung des Vektorfeldes . . . . .	114
5.2.1	Vorzugsrichtung bei Gefäßstrukturen . . . . .	115
5.2.2	Vorzugsrichtung bei Muskelstrukturen . . . . .	115
5.2.3	Approximation fehlender Krümmungsinformation . . . . .	116

---

---

5.2.4	Datenstrukturen . . . . .	116
5.3	Oberflächenparametrisierung . . . . .	118
5.3.1	Verwaltung Parameterkoordinaten . . . . .	118
5.3.2	Zuweisung der Parameterkoordinaten . . . . .	119
5.3.3	Datenstrukturen . . . . .	120
5.4	Texturierung . . . . .	121
5.4.1	Generierung der Open Inventor-Szene . . . . .	122
5.4.2	Texturierung in MeVisLab . . . . .	122
5.5	Entwickelte MeVisLab-Module . . . . .	123
5.5.1	Entwickelte Module . . . . .	124
<b>6</b>	<b>Ergebnisse</b>	<b>127</b>
6.1	Vektorfelder . . . . .	127
6.1.1	Gefäßstrukturen . . . . .	128
6.1.2	Muskelstrukturen . . . . .	131
6.1.3	Organe und Knochenstrukturen . . . . .	134
6.1.4	Schlussfolgerungen . . . . .	138
6.2	Schraffierungen . . . . .	139
6.2.1	Muskelstrukturen . . . . .	139
6.2.2	Gefäße . . . . .	141
6.2.3	Organe und Knochenstrukturen . . . . .	142
6.2.4	Kombinierte Oberflächendarstellung . . . . .	143
6.2.5	Schlussfolgerung . . . . .	145
6.3	Laufzeitanalyse . . . . .	145
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>147</b>
7.1	Ausblick und Verbesserungsvorschläge . . . . .	148
	<b>Literaturverzeichnis</b>	<b>151</b>



## 1 Einleitung

Seit etwa 20 Jahren werden in der Computergraphik mit Hilfe des *Non-Photorealistic Rendering* (NPR) illustrative Darstellungsstile nachgeahmt (STROTHOTTE UND SCHLECHT- WEG [2002]). Motiviert durch deren traditionelle Anwendung in anatomischen Illustrationen, erschloss sich dem NPR-Bereich ein interessantes Anwendungsfeld. Die Darstellungsstile, mit denen diese Illustrationen bisher erstellt wurden, haben sich über Jahrhunderte entwickelt und durchgesetzt. Besonders *Pen-and-Ink*-Techniken, zu denen Silhouetten, Merkmalslinien (*Feature Lines*), Punktierungen (*Stippling*) sowie Schraffierungen (*Hatching*) gehören, dienen zur Erzeugung von informativen und expressiven Abbildungen der menschlichen Anatomie. Die dabei im NPR-Bereich entstandenen Verfahren werden als *Illustrative Rendering*-Techniken bezeichnet.

Traditionelle Illustrationen werden jedoch nicht patientenspezifisch erzeugt. Auf Grund ihres langwierigen und aufwendigen Gestaltungsprozesses sowie ihrer Anwendung in der medizinischen Lehre werden sie vielmehr in der Vermittlung von standardisierten, anatomischen Strukturen eingesetzt. Wünschenswert ist es daher zum einen, computerunterstützte Verfahren zu entwickeln, die das Potential illustrativer Techniken auch auf individuellen Patientendatensätzen anwenden. Zum anderen diese Techniken mit den Möglichkeiten der interaktiven Exploration bei 3D-Visualisierungen zu kombinieren. Dieses Ziel verfolgt die vorliegende Arbeit und stellt einen neuen Beitrag bezüglich dieser Entwicklung im Bereich der medizinischen Illustration dar.

Als eine besonders effektive *Pen-and-Ink*-Technik hat sich die Schraffierung in der medizinischen Illustration bewährt. Schraffuren vermitteln Informationen über Oberflächenverlauf, Material, Beleuchtung und räumliche Lage zu anderen Strukturen. Der Fokus dieser Arbeit liegt daher auf der Untersuchung von Schraffuren anatomischer Strukturen und wie diese automatisch auf entsprechend segmentierten 3D-Modellen aus Patientendatensätzen erzeugt werden können. In bisherigen Arbeiten zu oberflächenbasierten *Pen-and-Ink* Schraffuren werden die Linien u. a. anhand von Krümmungsinformationen ausgerichtet. Dieser Ansatz soll in der vorliegenden Arbeit weiterverfolgt und zusätzlich um Modellwissen der anatomischen Strukturen erweitert werden. Für die Schraffierung wird ein texturbasierter Ansatz verwendet sowie die Arbeit von SCHULZ [2005] zur Krümmungsapproximation fortgesetzt.

## 1.1 Zielsetzung

Das Ziel dieser Arbeit ist die Erstellung eines Verfahrens, das segmentierte, polygonale 3D-Modelle aus Patientendatensätzen für eine bessere Formwahrnehmung und Strukturunterscheidung schraffiert. Das Verfahren soll eine *frame*-kohärente und echtzeitfähige Interaktion in der Betrachtung erlauben. Dazu sind zunächst verschiedene Untersuchungen notwendig. Einerseits die Betrachtung von Schraffuren anatomischer Strukturen in entsprechenden Atlanten oder anderen medizinischen Illustrationen. Nicht jede Struktur wird schraffiert und die Orientierungen der Schraffurlinien unterscheiden sich in Abhängigkeit der jeweiligen Struktur. Andererseits können die polygonalen 3D-Modelle spezielle Problembereiche besitzen, die es zu berücksichtigen gilt.

Neben der Interaktions- und Echtzeitfähigkeit sind an das Schraffieren von polygonalen, medizinischen 3D-Modellen weitere Anforderungen gestellt. Anhand von bisherigen Forschungsarbeiten sollen existierende Ansätze betrachtet und bezüglich der Erfordernisse an das hier zu entwickelnde Verfahren diskutiert werden. Die dabei erfassten Vor- und Nachteile werden erläutert sowie Rückschlüsse für die eigene Entwicklung gezogen. Dabei fließen neben wahrnehmungspsychologischen und illustrativen Aspekten auch Betrachtungen von Krümmungsapproximationen und Texturierung polygonaler 3D-Modelle mit ein.

## 1.2 Gliederung der Arbeit

Zum Bearbeiten der Zielstellung ist die vorliegende Arbeit in folgende Kapitel gegliedert:

In **Kapitel 2** wird zunächst auf die traditionelle Schraffierungstechnik in illustrativen Darstellungen eingegangen. Dazu werden medizinische Atlanten und Illustrationen bezüglich der Schraffierung anatomischer Strukturen untersucht. Im Anschluss erfolgt eine Betrachtung und Diskussion bestehender Schraffierungsansätze. Für den texturbasierten Ansatz dieser Arbeit werden Grundlagen für die Texturierung polygonaler 3D-Modelle erarbeitet.

Für die Orientierung der Schraffurlinien werden u. a. Krümmungsinformationen der zu schraffierenden Objekte verwendet. **Kapitel 3** erläutert die dafür notwendigen mathematischen Grundlagen und stellt bestehende Approximationsverfahren sowie Schlussfolgerungen für deren Anwendung vor.

---

---

In **Kapitel 4** wird der Entwurf für das hier zu entwickelnde Verfahren erarbeitet. Auf Grundlage der gesammelten Untersuchungsergebnisse und den verwendeten Datensätzen werden zunächst wichtige Vorbetrachtungen und Anforderungen für das Verfahren getroffen sowie zusammengefasst. Daraus ergeben sich zwei komplexe Vorverarbeitungsschritte. Zum einen die Erzeugung von krümmungs- und modellbasierten Vektorfeldern und zum anderen die Parametrisierung der Oberflächenmodelle. Der Entwurf dieser beiden Aufgaben stellt den Schwerpunkt des Kapitels dar.

Basierend auf den eigenen Entwurf wird in **Kapitel 5** eine echtzeitfähige Visualisierung mit Hilfe der Bildverarbeitungs- und Visualisierungsplattform MEVISLAB und der 3D-Graphikbibliothek OPEN INVENTOR erzeugt. Die einzelnen Vorverarbeitungsschritte aus Kapitel 4 sind dazu in entsprechenden Klassen implementiert. Speziell verwendete Datenstrukturen und Bibliotheken werden aufgezeigt sowie erläutert.

In **Kapitel 6** erfolgt eine Präsentation der erzielten Ergebnisse. Diese werden mit den definierten Anforderungen verglichen und diskutiert. Die Ergebnisse beziehen sich dabei auf die erzeugten krümmungs- und modellbasierten Vektorfelder sowie auf die Oberflächenparametrisierung.

Eine Zusammenfassung der gesamten Arbeit findet in **Kapitel 7** statt. Alle wichtigen gewonnenen Resultate und Erkenntnisse werden dargelegt und bestehende Probleme genannt sowie mögliche Verbesserungsvorschläge und weiterführende Arbeiten aufgezeigt.



## 2 Schraffuren in medizinischen Illustrationen

Die vorliegende Arbeit ist im Bereich der klassischen *Pen-and-Ink*-Techniken für medizinische Illustrationen eingeordnet. Der Fokus liegt dabei auf einer krümmungs- und modellbasierten Echtzeitschraffierung von polygonalen Oberflächen aus individuellen Patientendatensätzen. Zunächst beginnt dieses Kapitel mit einer Einführung und der Motivation für den Einsatz von illustrativen Darstellungen in der Medizin. Anschließend werden Schraffuren vorgestellt und dabei wichtige Oberflächenwahrnehmungs- und Gestaltprinzipien bei dieser Art von Illustrationstechnik aufgezeigt. Darauf erfolgt eine Analyse medizinischer Illustrationen hinsichtlich der Verwendung von Schraffuren bei anatomischen Strukturen.

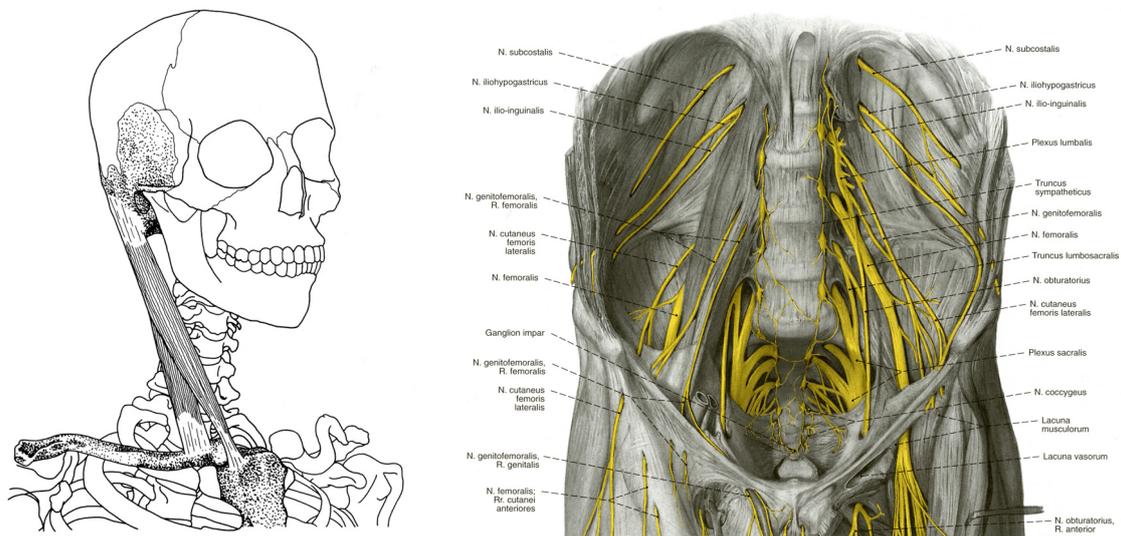
Es gibt eine Vielzahl bereits vorhandener Verfahren, die Schraffuren umsetzen und in diesem Kapitel klassifiziert sowie diskutiert werden. Vor- und Nachteile bezüglich der zu erreichenden Ziele dieser Arbeit werden herausgearbeitet und Rückschlüsse für den eigenen Ansatz gezogen. Basierend auf der Analyse bestehender Schraffurverfahren wird in dieser Arbeit ein texturbasierter Schraffuransatz gewählt. Für das damit verbundene *Texture Mapping* werden wichtige Grundlagen und Probleme diskutiert. Der Einsatz von *Hardware-Shadern* im Bereich der Echtzeit-Visualisierung hat in den letzten Jahren stark zugenommen. Daher wird zusätzlich auf deren Möglichkeiten für die Verwendung in der Texturierung eingegangen.

Da die Ausrichtung der Texturen u. a. durch krümmungsbasierte Vektorfelder geschieht, erfolgt abschließend eine Erläuterung der mathematischen Grundlagen und Zusammenhänge von Oberflächenkrümmungen (vgl. Abschnitt 3.1). Es findet eine Vorstellung existierender Verfahren zur Krümmungsapproximation polygonaler Oberflächen statt, basierend auf der Arbeit von SCHULZ [2005]. Der Fokus liegt hier auf einer Untersuchung, welche Verfahren für den Einsatz bei den später verwendeten Polygonmodellen sinnvoll sind.

### 2.1 Medizinische Illustrationen

Für die vorliegende Arbeit werden Schraffuren im Kontext medizinischer Illustrationen verwendet. Bevor deren konkreter Einsatz in diesem Anwendungsgebiet untersucht wird, sollen in diesem Abschnitt allgemeine Aspekte medizinischer Illustrationen betrachtet werden, um aufzuzeigen, dass das innenwohnende Abstraktionspotenzial dieser Art von Illustrationen und von Schraffuren einen direkten Bezug zueinander haben.

Illustrationen in der Medizin haben eine lange Geschichte, deren Anfänge im späten, ausgehenden Mittelalter zu finden sind. Die älteste gedruckte, medizinische Illustration wurde in dem Werk *Fasciculus Medicinae* von Johannes de Ketham aus dem Jahre 1491 in Venedig gefunden. Detailreichere Illustrationen, die erstaunlich hohe Kenntnisse der makroskopischen Anatomie vermitteln, sind in dem 1543 erschienenen ersten Anatomiebuch *De Humani Corporis Fabrica* von Andreas Vesalius zu finden. Mit Hilfe der unter Abschnitt 2.2 erwähnten *Pen-and-Ink*-Techniken wurde versucht, den komplexen Körperbau des Menschen darzustellen. Die verwendeten Techniken sind bis heute als Stil- und Gestaltungsmittel für medizinische Illustrationen bewährt und erhalten geblieben. Dies zeigt die Abb. 2.1 (a) aus einem heutigen Atlanten, in der neben Schraffierung der Muskeln, Silhouetten, Merkmalslinien und Punktierungen für die Anatomiebeschreibung eingesetzt sind.



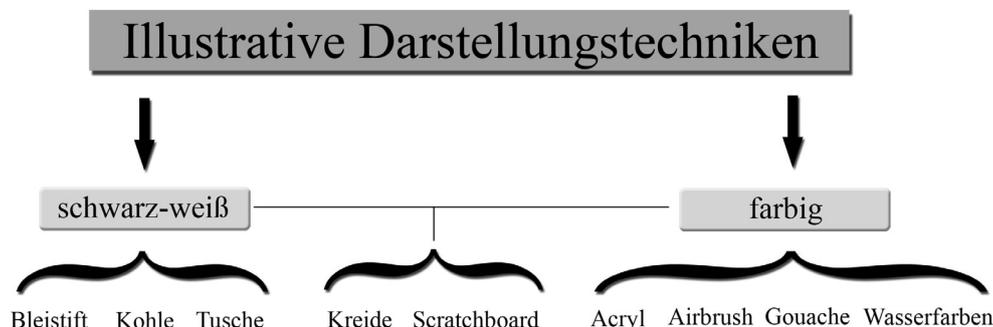
(a) Quelle: STONE UND STONE [2000]

(b) Quelle: PUTZ UND PABST [1993a]

**Abb. 2.1:** Beispiele heutiger medizinischer Illustrationen mit (a) klassischen *Pen-and-Ink*-Techniken und (b) Ergänzungen von Farbe und *Shading* zur Hervorhebung von Fokus- und Kontextstrukturen.

Ergänzt und unterstützt werden diese Techniken durch den Einsatz von Farben, neuen Zeichentechniken sowie modernen Bildbearbeitungsprogrammen, die dem Illustrator weitere Gestaltungsstile wie *Shading* zur Verfügung stellen. Damit lassen sich bessere Differenzierungen und Fokus-Kontext-Beziehungen einzelner Strukturen erzielen, wie dies in Abb. 2.1 (b) zu sehen ist. Darin ist die Lendenmuskulatur in schwarz-weiß Schraffur als Kontextstruktur und die Spinalnerven des Lenden- und Kreuzabschnittes farblich-glatt als Fokusstruktur dargestellt. Eine allgemeine Übersicht der gängigen Darstellungstechniken zeigt Abb. 2.2, wobei eine nähere Betrachtung dieser Techniken in dieser Arbeit nicht erfolgen kann. Der interessierte Leser sei auf die Arbeiten von EBERT UND SVAKHINE

[2006], SOUSA U. A. [2005], SALAH [2006] sowie VIOLA U. A. [2006] verwiesen.



**Abb. 2.2:** Klassifikation von illustrativen Darstellungstechniken.

Das Ziel medizinischer Illustrationen ist somit die Vermittlung von Informationen komplexer, anatomischer Zusammenhänge oder Sachverhalte. Dabei gilt das Prinzip: „So detailliert wie nötig und so einfach wie möglich“. Unwichtige oder störende Informationen werden ausgeblendet oder unterdrückt und relevante Informationen hervorgehoben. Hierin zeigen sie klare Vorteile gegenüber den gängigen Abbildungsverfahren (Fotographie, CT, MRT und Ultraschall), die teilweise das Verständnis der Aufnahmen für den (ungeübten) Betrachter erschweren oder behindern. Die Vorteile lassen sich wie folgt zusammenfassen:

- **Entfernung von Aufnahmeartefakten**

Zu diesen gehören beispielsweise Rauschen, Partialvolumeneffekte, Überblendungen durch Metallimplantate bei CT-Aufnahmen oder auch Bewegungsunschärfe. Diese verfälschen die Bilder und erschweren deren Interpretation.

- **Abstraktion**

Durch die Abstraktion der Realität werden „störende“ und irrelevante Informationen, wie Blut und Fettgewebe, weggelassen. Die Form und Struktur, Betonung und Abschwächung von Fokus- und Kontextinformationen können somit schnell und effektiv vermittelt werden (VIOLA U. A. [2005]). Dieses Prinzip veranschaulicht Abb. 2.3. Anatomisch wichtige Informationen sind in der apparativen OP-Aufnahme, für einen ungeübten Betrachter schwer zu extrahieren. Störende Strukturen wie Blut, Fett und OP-Materialien behindern den Betrachter beim Verstehen der Aufnahme. Mit einer simplifizierten Illustration beschränkt sich die Situation auf die wesentlichsten Informationen und ermöglicht dem Betrachter klarere Einblicke in die Anatomie.

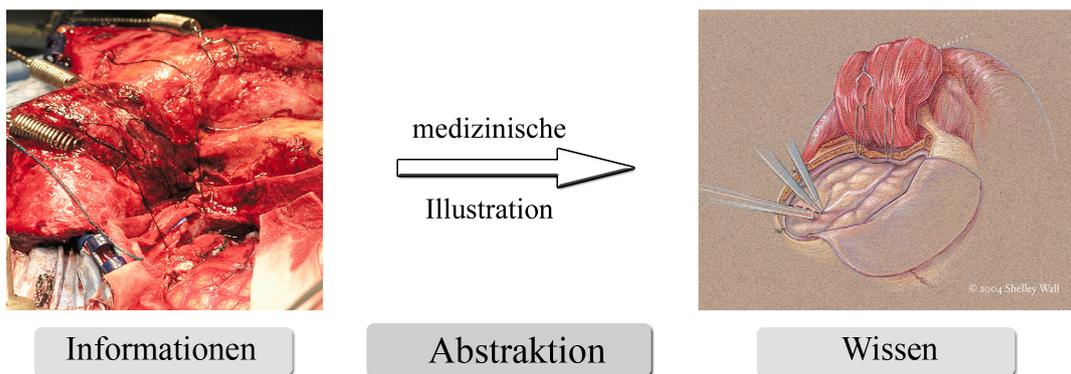
- **Wahl der Sichtbarkeit**

Dazu gehört zum einen das Entfernen von Verdeckungen auf relevante Strukturen, z. B. durch Haut, Muskeln, Organe oder Knochen. Zum anderen kann ein optimaler

Blickwinkel gewählt werden, so dass der Betrachter eine freie Sicht auf den relevanten Bereich hat.

- **Hervorhebung und Abschwächung**

Gezielt können Fokus- und Kontextstrukturen dargestellt und die Aufmerksamkeit auf das Wesentliche gerichtet werden. Probleme mit mangelnder Gewebedifferenzierung durch zu geringe Intensitätsschwankungen in den Schnittbildverfahren werden beispielsweise damit umgangen.



**Abb. 2.3:** Veranschaulichung des Prinzips von medizinischen Illustrationen. Es findet eine Informationsreduktion auf das Wesentliche statt. Bildquellen: SOUSA [2006] (links), WOOLRIDGE UND JENKINSON [2004] (rechts)

Angewendet werden die medizinischen Illustrationen heutzutage hauptsächlich in anatomischen Atlanten zu Lehr- und Ausbildungszwecken. Beispielsweise besitzen Medizinstudenten noch nicht das nötige Fachwissen und die mentale Vorstellung über reale anatomische Strukturen. Durch derartige Illustrationen kann ihnen schrittweise dieses Wissen vermittelt werden. Auf Grund ihrer innewohnenden Abstraktion sind Illustrationen jedoch nicht anwendbar, wenn es um diagnostische Untersuchungen bzw. Therapieentscheidungen geht. Bei diesen kann der Informationsgehalt nicht hoch genug sein und sie werden nur anhand der oben genannten Aufnahmeverfahren mit ihren diesbezüglich klaren Vorteilen getroffen.

In diesem Zusammenhang sind die Arbeiten von KRÜGER U. A. [2005] und TIETJEN U. A. [2005] zu erwähnen, die jeweils für patientenindividuelle Datensätze bei der Chirurgieausbildung und Therapieplanung verschiedene illustrative Darstellungs- und Interaktionstechniken mit Volumenvisualisierungen kombinieren. Die Ergebnisse und Evaluierungen von Medizinern zeigen dabei, dass sich die eingesetzten Techniken zur Übermittlung von räumlichen Informationen eignen und die Therapieplanung unterstützen.

## 2.2 Schraffur in illustrativen Darstellungen

Die Schraffierung gehört zu der Gruppe der *Pen-and-Ink*-Techniken und ist ein bewährter traditioneller Zeichenstil für die Erstellung von Illustrationen. Die Bildszene wird mit Hilfe von geschickt platzierten Strichen, die auch *Strokes* genannt werden, und Linienzügen abstrahiert dargestellt. Der Illustrator erzeugt dadurch Fokus- und Kontextbereiche, Schattierungen (*Shading*), Texturen, Oberflächeneigenschaften sowie Tiefeninformationen (*Depth Cues*) der Objektszene (ANDREWS [2006]). In Abb. 2.4 (a) ist dazu ein Beispiel einer zeitgenössischen Illustration gegeben. Die Beleuchtung, verschiedene Materialeigenschaften und Objektformen werden durch variationsreichen Einsatz von einfachen Strichen beschrieben.

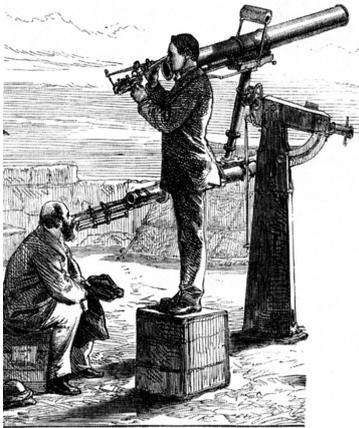
Bei einer näheren Untersuchung von Schraffuren in illustrativen Darstellungen werden zwei wichtige Aspekte deutlich, die sich gegenseitig bedingen. Zum einen verwendet der Illustrator verschiedene Gestaltungsparameter bei der Strichführung und -platzierung für die expressive Gestaltung der Szene. Zum anderen werden über diese Parameter Informationen über Oberflächenverlauf und -orientierung vermittelt. Diese beiden Konzepte werden in den folgenden Abschnitten 2.2.1 und 2.2.2 näher betrachtet.

Die Schraffierungstechnik findet durch ihre expressive Aussagekraft in vielen Bereichen der illustrativen Darstellung (z. B. Botanik, Zoologie, Archäologie, Technik und Medizin) ihre Anwendung. Da für diese Arbeit die medizinische Illustration von Interesse ist, wird in Abschnitt 2.2.3 der Einsatz von Schraffuren in anatomischen Illustrationen untersucht. Eine Motivation ist die mögliche Verwendung bei eingebetteten Oberflächen, wo gleichzeitig ein äußeres Objekt, z. B. ein Organ, und innere Objekte, z. B. Gefäße und Tumore, dargestellt werden sollen. Hier könnten Schraffuren eine Alternative oder Ergänzung zur bisher semitransparenten Dargestellung des äußeren Objektes sein.

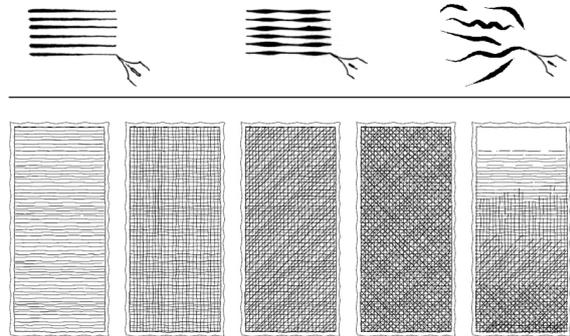
Bezüglich *Pen-and-Ink*-Techniken sei noch erwähnt, dass zu dieser Gruppe neben Schraffuren auch Silhouetten, Merkmalslinien und Punktierung gehören. Diese sollen aber nicht näher betrachtet werden, da sie im Rahmen dieser Arbeit nicht weiter relevant sind. Der interessierte Leser sei diesbezüglich auf die Arbeiten von ISENBERG U. A. [2003], SOUSA UND PRUSINKIEWICZ [2003] sowie PASTOR U. A. [2003] verwiesen. Eine interessante Studie von ISENBERG U. A. [2006] zeigt in diesem Zusammenhang, dass aktuelle NPR-Verfahren, die *Pen-and-Ink* Illustrationen erzeugen, zum einen nicht in jedem Fall die Qualität von Handzeichnungen erreichen. Zum anderen verdeutlicht sie aber auch, welches Potential und Unterstützungsmöglichkeit für den Illustrator in dem Verfahren steckt.

### 2.2.1 Gestaltungsparameter bei Schraffuren

Striche und Linienzüge besitzen verschiedene Gestaltungsparameter oder auch Freiheitsgrade, die ein Illustrator einsetzt. Diese werden beispielsweise durch Stiftdruck, natürli-



(a) Quelle: STROTHOTTE UND SCHLECHTWEG [2002]



(b) Quelle: SALISBURY U. A. [1994], SOUSA U. A. [2004]

**Abb. 2.4:** (a) Einsatz von Schraffuren in illustrativen Darstellungen und (b) Beispiele verschiedener Freiheitsgrade bezüglich Linienstile und Tönungsstufen.

che Handführungen und ästhetisches Empfinden des Zeichners bestimmt und beeinflussen neben seiner Begabung maßgeblich das Ergebnis der Illustration (STROTHOTTE UND SCHLECHTWEG [2002]). Insgesamt lassen sich vier Parameter aus den Illustrationen ableiten, die im Folgenden kurz beschrieben werden.

### Linienabstand

Der Illustrator steuert maßgeblich über den Linienabstand die Tönung eines Objektes. Dunkle Bereiche werden durch eng aneinander liegende Linien erzeugt und hellere entsprechend mit weiterem Abstand. Zur Verstärkung von dunklen Gebieten werden häufig auch Kreuzschraffuren (*Cross-Hatching*) verwendet. Durch Kombinationen können dann verschiedene Tönungsstufen erzeugt werden (vgl. Abb. 2.4 (b), untere Reihe). Auch Materialeigenschaften lassen sich dadurch vermitteln. Dabei wirken beispielsweise Oberflächen hart, wenn diese mit relativ gleichmäßig parallelen Linien gezeichnet werden und weich durch unregelmäßigere Abstände der Schraffurlinien (WINKENBACH UND SALESIN [1994]). Dies ist in Abb. 2.4 (a) bei den Kistenflächen und dem Erdboden zu erkennen.

### Linienlänge

Über die unterschiedliche Länge der Linien wird zum einen die Beleuchtungsrichtung definiert. An Stellen, wo Glanzpunkte auf dem Objekt entstehen oder das Objekt sehr stark beleuchtet ist, werden die Linien abgeschnitten bzw. laufen aus (vgl. Abb. 2.5 und 2.6).

Zum anderen beeinflusst die Linielänge den Gesamteindruck der Illustration. So bewirken kurze, immer wieder neu angesetzte Linien einen natürlichen, skizzenhaften Eindruck. Durchgezogene Linien erzeugen hingegen mehr einen technischen, maschinenhaften Stil. Verkürzt sich die Länge auf Punktgröße, entspricht die Schraffierung der Punktierungstechnik, was beispielsweise für auslaufende Linien in hellen Objektbereichen angewendet wird.

### **Linienstil**

Zum Linienstil gehören die Breite und Linienführung (vgl. Abb. 2.4 (b), obere Reihe), womit bestimmte Akzente und Oberflächenhinweise für ein Objekt gesetzt werden. Eine unterschiedliche Breite vermittelt dabei Informationen über den lokalen Krümmungsverlauf, indem Linienbereiche an starken Krümmungen breiter gezeichnet werden als in flacheren Bereichen. Dickere Linienzüge geben auch Hinweise von Zusammenführungen sowie Schattenregionen und unterstreichen damit die Beleuchtung der Szene. Zur Hervorhebung der Objekttiefe werden dünne Linienzüge verwendet, wenn Objekte weiter vom Betrachter oder einem Referenzobjekt liegen. Wellige Linienzüge entsprechen mehr einer handgezeichneten, skizzenhaften Darstellung oder auch weicherem Material. Hingegen unterstützen gerade Linien einen technischen Zeichenstil oder auch harte Materialien.

### **Linienrichtung**

Die Orientierung der Linien beinhaltet sehr stark wahrnehmungspsychologische Aspekte. Zum einen vermitteln sie Informationen über den Oberflächenverlauf, d. h. die Krümmung eines Objektes. Sie kennzeichnen dabei elliptische<sup>1</sup>, zylindrische und hyperbolische Bereiche. Zum anderen verdeutlichen sie auch Metainformationen, wie beispielsweise der Kontraktions- und Dehnungsrichtung von Muskeln (vgl. Abschnitt 2.2.3). Weiterhin beeinflussen sie zusätzlich die Objektwahrnehmung wie der folgende Abschnitt zeigt.

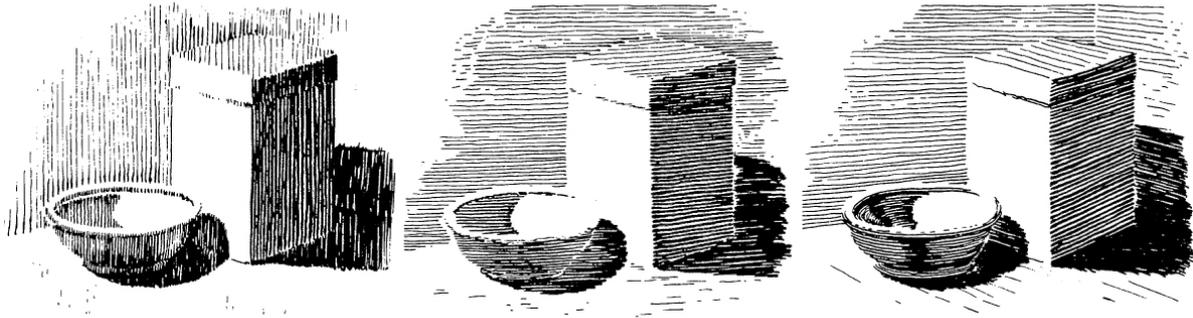
#### **2.2.2 Oberflächenwahrnehmung mit Schraffuren**

Es wurde bereits erwähnt, dass verschiedene Linienstile und Orientierungen der Schraffur Einfluss auf Tiefen- und Oberflächenwahrnehmung haben. Wahrnehmungspsychologische Untersuchungen haben dazu festgestellt, dass die Ausrichtung der Linien einen sehr starken Einfluss hat und findet sich bestätigt in den Aussagen und Abbildungen verschiedener Illustratoren (PITZ [1984], ANDREWS [2006]). Dies soll zunächst anhand von Abb. 2.5 verdeutlicht werden. Dieselbe Szene wird in drei unterschiedlichen Schraffurrichtungen repräsentiert, vermittelt aber jeweils eine unterschiedliche Wahrnehmung

---

<sup>1</sup>Darin sind auch die Oberflächenorientierungen konvex und konkav eingeschlossen.

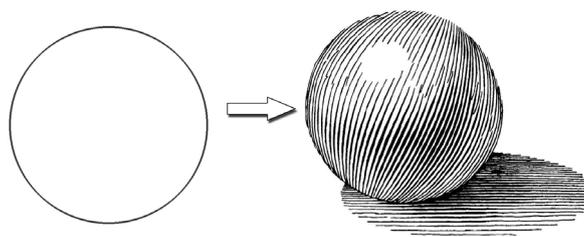
der Objekte. Dabei wirken die Objekte durch eine vertikale Schraffur im linken Bild länger und im mittleren Bild durch eine horizontale Ausrichtung breiter. Auch die Figur-Grund-Trennung und Tiefenwahrnehmung werden durch diese Art der Linienführung nicht unterstützt und vermitteln falsche Aussagen beispielsweise im Bereich des Deckels der beiden erwähnten Bilder. Erst im rechten Bild wird durch unterschiedlich orientierte Linien eine aussagekräftige und korrekte Objektwahrnehmung erzeugt.



**Abb. 2.5:** Einfluss der Linienrichtung auf die Objekt- und Tiefenwahrnehmung sowie Figur-Grund-Trennung. Die linke und mittlere Szene vermitteln jeweils verschiedene Ausdehnungen der gleichen Objekte. Zudem erschwert die Orientierung der Schraffurlinien in den beiden Abbildungen die Figur-Grund-Trennung. Die rechte Szene, mit korrekt ausgerichteter Schraffur, vermittelt hingegen eine gute Objektwahrnehmung. Quelle: PITZ [1984]

Des Weiteren ist die Richtung ein wichtiger Indikator für die schon erwähnte Vermittlung von Krümmungsinformationen der Oberfläche. Am Beispiel von Abb. 2.6 soll dies veranschaulicht werden. Mit Hilfe von bewusst gesetzten Schraffurlinien und Linienstilen erscheint der Kreis als eine in Szene gesetzte Kugel. Die Orientierung der Linien entlang der Oberflächenkrümmung, das parallele Ausrichten der Linien auf dem imaginären „Untergrund“ sowie zusätzliche Variationen in Linienstil und -länge verleihen dem Kreis eine räumliche Dimension. Es zeigt sich daher, dass allein durch Silhouetten oder auch eventuell darin enthaltenen Merkmalslinien nicht genügend Informationen über den Oberflächenverlauf vermittelt werden können. Ein Problem ist zudem bei Silhouetten als auch beim *Shading*, dass diese im Gegensatz zu Krümmungslinien nicht geometrisch-invariant, sondern sicht- und beleuchtungsabhängig sind. Obwohl sie ebenfalls als wichtige Indikatoren (*Cues*) für die Oberflächen- und Tiefenwahrnehmung von Objekten gelten, eignen sie sich daher nicht für genaue Aussagen über den Krümmungsverlauf von Objekten (GIRSHICK U. A. [2000]). An dieser Stelle sei noch mal erwähnt, dass Schraffuren auch beleuchtungsabhängig gesetzt sowie modifiziert werden und damit ebenfalls *Shading*-Informationen vermitteln. Dies ist beispielsweise in den Schattenbereichen der Abb. 2.5 und 2.6 zu erkennen, worauf auch schon der Abschnitt 2.2.1 über die Gestaltungsparameter eingegangen ist.

Die Tatsache, dass die Linienorientierung den Oberflächenverlauf eines Objektes beschreibt, wurde in verschiedenen Studien untersucht und bestätigt. Die Autoren der



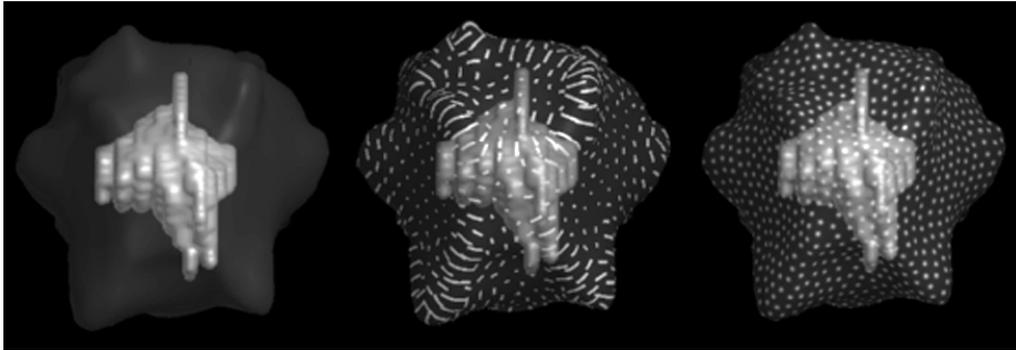
**Abb. 2.6:** Hervorhebung von Oberflächenkrümmung, -beleuchtung und -tiefe durch den Einsatz von Schraffurlinien. Quelle: nach HODGES [1988]

verschiedenen Arbeiten setzten dabei an der psychophysikalischen Feststellung an, dass die menschliche Wahrnehmung sensibel und präattentiv auf Texturen zur Oberflächenbeschreibung reagiert. Sie gingen daher der Frage nach, wie Texturen gestaltet und orientiert sein müssen, um den 3D-Oberflächenverlauf am besten zu vermitteln.

In den ersten Studien von der Arbeitsgruppe um Victoria Interrante (INTERRANTE U. A. [1997], INTERRANTE [1997]) wurde untersucht, wie eingebettete Oberflächen bei der Bestrahlungsplanung so visualisiert werden können, dass die räumliche Beziehung zueinander genau erkennbar sind (vgl. Abb. 2.7). Untersuchungsgegenstände waren dabei die gleichzeitige Darstellung eines Isozentrums um das zu bestrahlende Gebiet (grau, opak) herum. Es hat sich gezeigt, dass eine texturierte Darstellung des Isozentrums, wo die einzelnen Linien an der ersten Hauptkrümmungsrichtung<sup>2</sup> ausgerichtet sind, am besten geeignet sind. Sie gewährleisteten einerseits eine gute Vermittlung des Oberflächenverlaufes vom Isozentrum und andererseits einen klaren Blick auf das innen liegende zu bestrahlende Objekt. Die Untersuchungspersonen konnten durch diese Visualisierung einen genaueren Abstand zwischen beiden Objekten ermitteln, als mit anderen Texturmustern, Linienausrichtungen oder einer transparenten Isozentrendarstellung (vgl. Abb. 2.7, links und rechts). Außerdem war bei diesen Visualisierungen der Aufwand zur Abstandsbestimmung mit einem höheren Interaktionsaufwand verbunden. Die Autoren erwähnen noch, dass zusätzliche Indikatoren für die Oberflächenbeschreibung wie *Shading* der Textur sowie Glanzlichter ergänzend wirken und das Ergebnis verbessern.

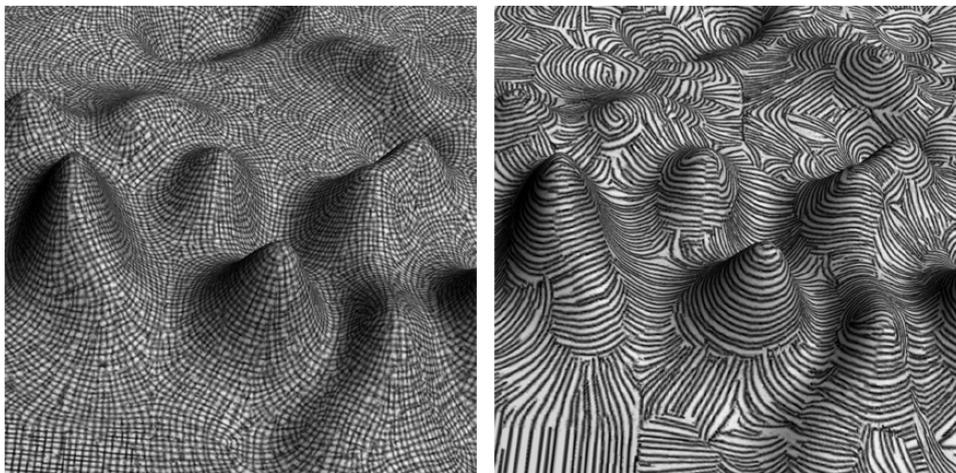
Eine genauere Untersuchung der Texturrichtung für die Beschreibung des Oberflächenverlaufes fand in INTERRANTE U. A. [2002] und KIM U. A. [2003] statt. Dabei wurden u. a. Linientexturen auf künstlich generierte Oberflächen projiziert und anhand der ersten oder zweiten Hauptkrümmungsrichtung sowie uniform ausgerichtet (vgl. Abb. 2.8). Probanden sollten daraufhin die Normalenrichtung an bestimmten Oberflächenpunkten angeben. Die genauesten Ergebnisse wurden mit Linientexturen erreicht, die an der ersten Hauptkrümmungsrichtung ausgerichtet sind. KIM U. A. [2003] stellte zudem fest, dass Kreuzschraffuren, die an beiden Hauptkrümmungsrichtungen ausgerichtet sind, et-

<sup>2</sup>Die erste Hauptkrümmungsrichtung eines Oberflächenpunktes gibt die Richtung der stärksten Krümmung an. Orthogonal dazu verläuft die zweite Hauptkrümmungsrichtung mit der geringsten Krümmung (vgl. Abschnitt 3).



**Abb. 2.7:** Schraffurlinien, die entlang der Hauptkrümmungsrichtung verlaufen (Mitte), verbessern die Wahrnehmung der räumlichen Oberflächenstruktur und erzielen eine genauere Abstandsbestimmung zum Bestrahlungsgebiet. Transparente Darstellungen (links) oder andere Texturmuster und -orientierungen (rechts) vermitteln ungenügende Informationen über den Oberflächenverlauf. Quelle: INTERRANTE U. A. [1997]

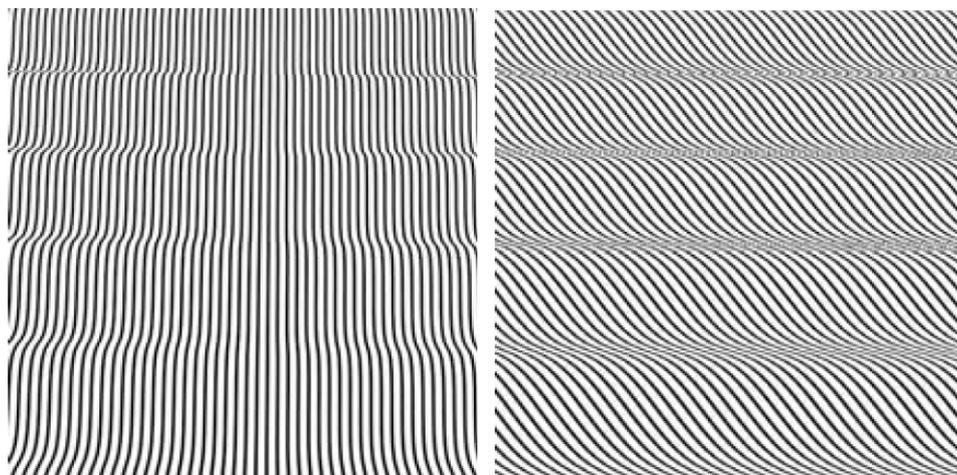
was bessere Ergebnisse erzielten, als bei einfach gerichteten Texturmustern an nur einer Hauptkrümmungsrichtung. Dies wurde in der Studie von KIM U. A. [2004] bestätigt, in welcher die Testpersonen verschiedene Oberflächenbereiche (elliptisch, zylindrisch oder hyperbolisch) klassifizieren sowie die dort vorhandene Oberflächenorientierung (flach, konkav, konvex oder hyperbolisch) bestimmen sollten. Ergänzend stellten die Autoren fest, dass die Blickrichtung dabei ebenfalls eine Rolle spielt. Ein schräger Blickwinkel auf die Textur und Oberfläche lieferte exaktere Ergebnisse als eine Draufsicht.



**Abb. 2.8:** Ausrichtung der Schraffurlinien entlang beider (links) oder einer der Hauptkrümmungsrichtungen (rechts). Quelle: KIM U. A. [2003]

Den Zusammenhang von Linienrichtung der Textur und Blickwinkel auf die Wahrnehmung der Oberflächenorientierung untersuchte die Arbeit von SWEET UND WARE [2004].

Die Probanden sollten auch hier die Normalenrichtung an verschiedenen Oberflächenpunkten bestimmen. Die Autoren zeigten, dass die Ausrichtung der Schraffurlinien an eine der beiden Hauptkrümmungsrichtungen teilweise nicht genügend Informationen über den Oberflächenverlauf vermittelt, wenn die Blickrichtung parallel dazu verläuft. In Abb. 2.9 ist dazu ein Beispiel gegeben. Es ist jeweils die gleiche Sinusoberfläche dargestellt, bei der im linken Bild die Schraffurlinien entlang der ersten Hauptkrümmungsrichtung und parallel zur Sichtrichtung verlaufen und im rechten schräg zur Sichtrichtung orientiert sind. Erkennbar ist, dass die Oberflächenform in der rechten Variante besser beschrieben wird, als in der linken.



**Abb. 2.9:** Eine parallele Blickrichtung zur Schraffur (links) bewirkt eine geringere Oberflächenwahrnehmung als ein windschiefes Verhältnis (rechts). Quelle: SWEET UND WARE [2004]

Diese Untersuchungen zeigten also, dass für eine optimale Beschreibung des Oberflächenverlaufes zunächst anisotrope Texturmuster geeignet sind. Diese werden durch Schraffurlinien, die entlang der Hauptkrümmungsrichtung ausgerichtet sind, erzeugt. Eine Darstellung der Linien in beide Hauptkrümmungsrichtungen liefert dabei mehr Informationen über den Oberflächenverlauf als nur in einer. Zu beachten ist allerdings eine Vermeidung von paralleler Sicht- und Schraffurrichtung, vor allem bei interaktiven Szenen. Schraffuren entlang der ersten Hauptkrümmungsrichtung umschließender Oberflächen eignen sich zudem für eine bessere Differenzierung und räumliche Lagebeziehung bei eingebetteten Oberflächenvisualisierungen. Ergänzend lassen sich zudem Schraffuren helligkeitsabhängig platzieren und vermitteln damit Beleuchtungs- und Tiefeninformationen durch das resultierende *Shading*.

### 2.2.3 Analyse von Schraffuren in medizinischen Illustrationen

Für den modellbasierten Entwurf eines Verfahrens zur Schraffur medizinischer Daten ist es notwendig zu untersuchen, wie üblicherweise anatomische Strukturen schraffiert wer-

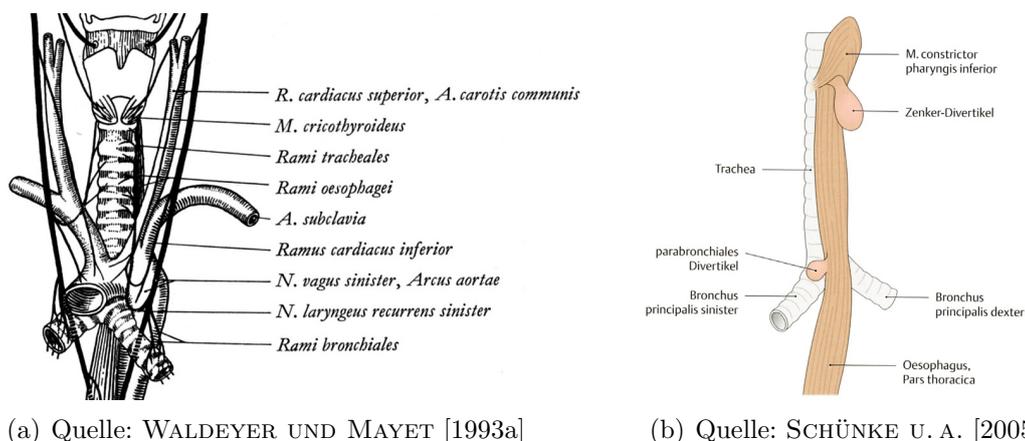
den. Dazu betrachtet dieser Abschnitt Abbildungen von verschiedenen Strukturen aus anatomischen Illustrationen. Die Betrachtung konzentriert sich dabei auf vier verschiedene Strukturtypen, bei denen u. a. Schraffur angewendet wird. Dazu gehören:

- **Röhrenförmige Strukturen,**
- **Muskel-, Sehnen- und Bindegewebe,**
- **Organe und Organbereiche** sowie
- **Knochenstrukturen.**

Eine differenzierte Analyse ist notwendig, da sich teilweise für die Strukturen verschiedene Schraffurverläufe und -stile gezeigt haben. Ein allgemeines Vorgehen ist daher nicht möglich. Die jeweils gewonnenen Informationen über den Verlauf fließen in das später zu entwickelnde Schraffurmodell der entsprechenden Strukturen ein.

### **Röhrenförmige Strukturen**

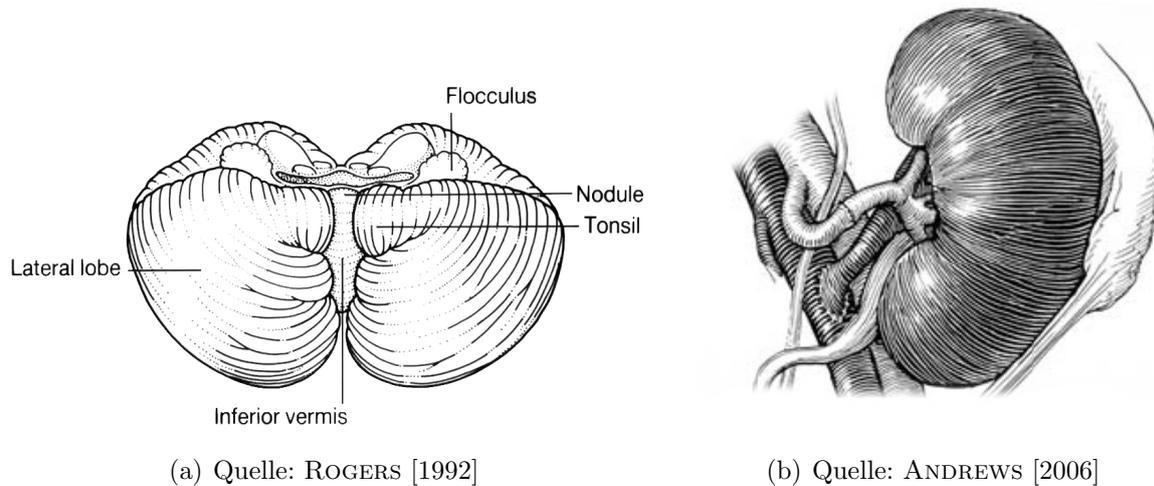
Diese Art von Strukturen kennzeichnet ein länglicher, zylindrischer Verlauf. Dazu gehören beispielsweise alle Arten von Blutgefäßen, Luft- und Speiseröhre, Darmverläufe sowie Nervenbahnen. In den Atlanten werden diese Strukturen jedoch meist und für die Nervenbahnen ausschließlich, nicht schraffiert. Letzteres begründet sich in einem zu geringen Durchmesserlauf. In Fällen wo doch schraffiert wird, hauptsächlich bei Blutgefäßen, verlaufen die Linien, außer bei der Speiseröhre, durchweg in radiale Richtung auf der Gefäßoberfläche (vgl. Abb. 2.10 (a)). Besonders bei Nahaufnahmen wird dadurch die zylindrische Form hervorgehoben. In Abb. 2.10 (b) ist eine Speiseröhre dargestellt, wobei sich die Schraffur entlang der Kontraktionsrichtung des Schluckmuskels orientiert. Die Beleuchtungssituation und zusätzliche Tiefenhinweise werden durch Variationen in Linienbreite, -länge und -dichte erreicht. Kreuzschraffuren für dunklere Bereiche kommen dafür nicht zum Einsatz, vermutlich weil diese die Wahrnehmung des zylindrischen Oberflächenverlaufes stören würden.



**Abb. 2.10:** Schraffur von röhrenförmigen Strukturen: In (a) werden besonders die Blutgefäße durch einen radialen Verlauf der Schraffurlinien beschrieben. Ansatzweise findet sich auch eine Schraffurung der Speiseröhre (*Oesophagus*). Diese ist jedoch in (b) deutlicher hervorgehoben.

## Organe und Organbereiche

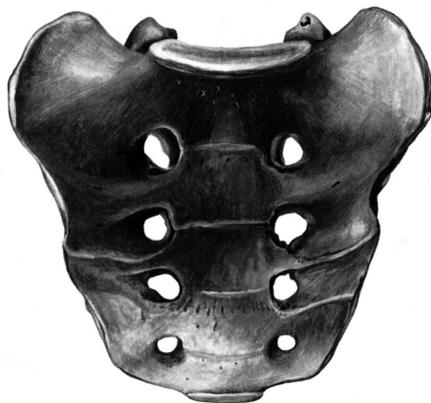
Organe sowie Organbereiche, wie beispielsweise Leber, Haut oder Gehirn werden in Atlanten sehr selten schraffiert. Ihre Illustration erfolgt durch andere Texturen und stilistische Mittel, wie dem *Shading*, wodurch eine größere Ähnlichkeit zu ihrer realen Erscheinung erreicht wird. Wenn allerdings ein größerer Abstraktionsgrad durch eine *Pen-and-Ink*-Darstellung erzielt werden soll, dann legt der Illustrator den Verlauf der Schraffurlinien so fest, dass die Form aus der momentanen Perspektive am besten hervorgehoben wird. Dazu zeigt zunächst Abb. 2.11 (a) die Illustration des Kleinhirns. Zur Hervorhebung der Beleuchtungssituation variiert der Linienstil, indem bei hellen Regionen die Linienzüge in eine Punktierung auslaufen und unterbrochen werden. Ein weiteres Beispiel ist in Abb. 2.11 (b) zu sehen, in der die Schraffurung des Magens und der umliegenden Blutgefäße dargestellt ist. Die Beleuchtung wird in diesem Fall durch Variation der Linienbreite und -länge erzeugt. Sehr deutlich wird dadurch auch der zylindrische Verlauf beider Strukturtypen hervorgehoben.



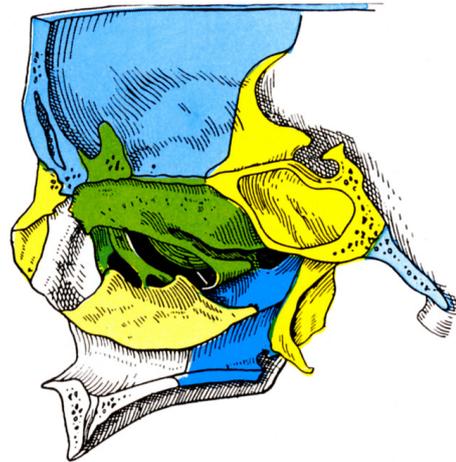
**Abb. 2.11:** (a) Schraffur von Kleinhirn und (b) Magen mit umgebenden Blutgefäßen. Durch jeweilige Variationen im Liniensstil, werden Oberflächenform und Beleuchtung zusätzlich betont.

## Knochen

Knochenstrukturen sind je nach Detaillierungsgrad unterschiedlich dargestellt. Für ein natürliches Aussehen werden in den meisten Atlanten keine Schraffuren verwendet, sondern eher schraffurähnliche Texturmuster in Kombination mit der schattierten Oberfläche. In Abb. 2.12 (a) ist dies beispielsweise an der Darstellung des Kreuzbeines zu sehen, wobei darin Beleuchtung, räumliche Tiefe und Oberflächenverlauf vorrangig durch die Schattierung beschrieben werden. Erkennbar sind jedoch auch Schraffurmuster, die konvexe sowie konkave Bereiche kennzeichnen und somit die Form aus der aktuellen Ansicht beschreiben. Eine Differenzierung einzelner Knochenstrukturen, wird mit Hilfe von Farbe hervorgehoben, wofür in Abb. 2.12 (b) ein Beispiel der rechten Nasenhöhle gezeigt ist. Darin dient die Schraffierung entlang der ersten Hauptkrümmungsrichtung zur abstrakten Beschreibung des Oberflächenverlaufes und der Beleuchtung. Für dunkle Bereiche werden Kreuzschraffuren verwendet und für helle Regionen Variation in Liniensstil, -länge und -abstand. Die unterschiedlichen Liniensbreiten betonen zudem die Stärke der lokalen Krümmung und heben ebenfalls die räumliche Tiefe hervor (vgl. Abschnitt 2.2.1).



(a) Quelle: WALDEYER UND MAYET [1993b]

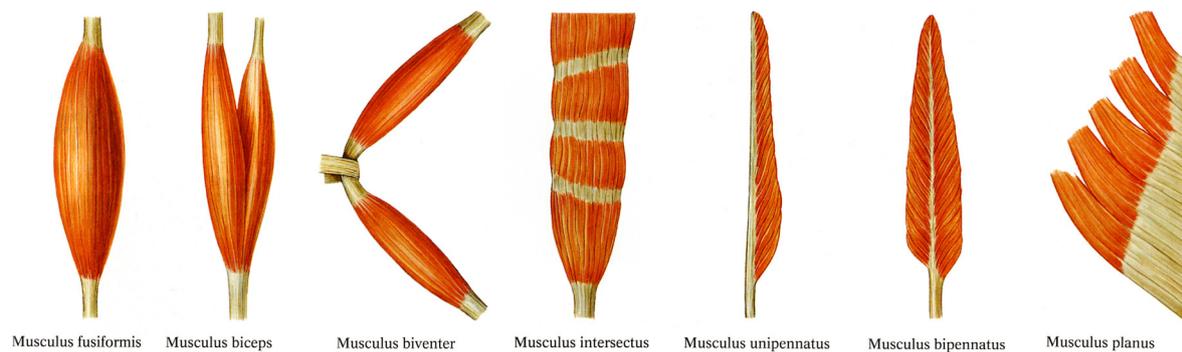


(b) Quelle: WALDEYER UND MAYET [1993a]

**Abb. 2.12:** In (a) dient vorrangig das *Shading* für die Beschreibung der Knochenoberfläche und die Schraffur nur als zusätzliche Information des Oberflächenverlaufes. Die Verwendung von Farbe erlaubt in (b) jedoch eine bessere Strukturdifferenzierung. Für Form und Beleuchtung wird ausschließlich Schraffur eingesetzt.

### Muskel-, Sehnen- und Bindegewebe

Muskel-, Sehnen- und Bindegewebe werden durchgängig in allen Atlanten schraffiert und sind darin auch die hauptsächlichen Anwendungsgebiete von Schraffuren. Im Gegensatz zu den bisherigen Strukturen erfolgt bei diesen Strukturen die Linienführung stets entlang der Kontraktions- und Dehnungsrichtung, da hiermit ihre funktionelle Anatomie beschrieben wird. Speziell beim Muskelgewebe zieht sich die Schraffur zwischen einen oder mehreren Ursprüngen und Ansätzen, die beide über Sehnen mit dem Knochen verbunden sind. Die Abb. 2.13 zeigt die verschiedenen Arten von Muskelformen und den jeweiligen Verlauf der Fasern. In der Anatomie werden die verschiedenen Formen in ring-, hohl-, spindel- und federförmige, mehrbäuchige sowie mehrköpfige Muskeln kategorisiert und legen damit Faser- bzw. Schraffurrichtung fest. Abhängig ist diese Einteilung u. a. von der Lagebeziehung zwischen Ursprung und Ansatz. Wie in der Darstellung weiterhin zu erkennen ist, fließen auch hier Krümmungsinformationen in den Linienverlauf ein, um beispielsweise spindelförmige Oberflächenverläufe hervorzuheben. Kombiniert werden die Schraffuren häufig mit Farbe und zusätzlichem *Shading* zur besseren Beschreibung von Beleuchtung und Form.



**Abb. 2.13:** Schraffurverlauf bei den verschiedenen Muskelformen. Definiert ist die Richtung durch die Sehnenursprünge und -ansätze des jeweiligen Muskels. Quelle: PUTZ UND PABST [1993b]

#### 2.2.4 Zusammenfassung und Schlussfolgerung

Mit Hilfe von illustrativen Darstellungen werden auf effektive Art und Weise Informationen über anatomische Strukturen vermittelt. Sie abstrahieren und reduzieren komplexe Zusammenhänge auf das Nötigste und erleichtern so das Verständnis. Bezüglich dieses Aspektes wurde gezeigt, dass sie dadurch gegenüber Bildgebungsverfahren und Fotografie entscheidende Vorteile besitzen. Jedoch können Illustrationen für diagnostische Zwecke nicht eingesetzt werden.

Eine Herausforderung ist die Wahl der richtigen Darstellungstechniken. Bewährt haben sich dabei u. a. Schraffur-Techniken. Veranschaulicht wurde, wie diese die Objektwahrnehmung beeinflussen und Informationen über den Oberflächenverlauf vermitteln. Es hat sich gezeigt, dass dabei die Orientierung der Schraffurlinien ein wichtiger Indikator ist. Der Linienverlauf erzielt die stärksten Wahrnehmungseffekte, wenn dieser den natürlichen, realen Oberflächenverlauf unterstreicht, was den beiden Hauptkrümmungsrichtungen entspricht. Von den betrachteten anatomischen Strukturen sind vorrangig Blutgefäße und Muskeln schraffiert. Bei ersteren verlaufen die Linien stets radial auf der Gefäßoberfläche. Im Falle von Muskeln existiert eine Vorzugsrichtung, welche durch die Kontraktions- und Dehnungsrichtung definiert ist. In diese fließen zusätzlich Krümmungsinformationen mit ein. Bezüglich des umzusetzenden Verfahrens müssen somit für die patientenspezifischen Oberflächenmodelle strukturindividuelle Schraffuren erzeugt werden. Das bedeutet einerseits eine geometrische Krümmungsapproximation der Oberflächenmodelle und andererseits Definierung von Ursprung und Ansatz bei Muskelstrukturen.

Als ebenfalls wichtig haben sich die Modifikationen von Linienstil, -abstand und -länge zur Hervorhebung von Tiefe und Beleuchtung gezeigt. Eine lokale Anpassung dieser Freiheitsgrade an den Oberflächenverlauf sollte von dem hier entwickelten Verfahren unterstützt werden.

## 2.3 Computergenerierte Schraffierungsansätze

Zur Simulation von krümmungs- und beleuchtungsabhängigen Schraffuren wurden im NPR-Bereich verschiedene Ansätze entwickelt. Diese lassen sich in bild-, objekt- und texturbasiert klassifizieren, wobei mit letzterem eine objektbezogene Verwendung von Texturen für die Schraffierung gemeint ist. Für das in dieser Arbeit umzusetzende Verfahren wird zunächst eine Anforderungsanalyse durchgeführt. Im Anschluss erfolgt eine nähere Betrachtung der klassifizierten Ansätze mit einer jeweiligen Diskussion hinsichtlich der getroffenen Anforderungen. Grundlage aller Verfahren sind dabei entweder Volumendatensätze oder polygonale Eingangsmodelle, da deren Einsatz in der medizinischen Visualisierung typisch ist.

### 2.3.1 Anforderungsanalyse

Die folgende Analyse dient zur klaren Definierung, was ein Verfahren zur Schraffierung patientenspezifischer Datensätze allgemein leisten muss. Die Anforderungen ergeben sich aus der interaktiven Exploration von 3D-Visualisierungen patientenspezifischer Strukturen.

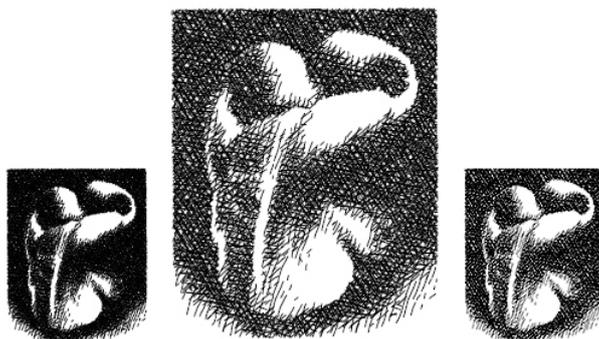
#### Frame-Kohärenz

Bei einer Interaktion in 3D-Visualisierungen hat der Betrachter die Möglichkeit, die Objekte von allen Seiten und Entfernungen zu betrachten. Das bedeutet zunächst für die Darstellung der schraffierten Oberfläche, dass sie kohärent gegenüber aufeinander folgende Bilder (*Frames*) sein muss. Diese Kohärenz bezieht sich dabei auf die Beleuchtung und Schraffur (nach PASTOR U. A. [2003]). Ersteres heißt, dass eine Änderung in der Objektbeleuchtung von Bild zu Bild durch sanfte Anpassung des Linienstiles und -abstandes geschehen muss, entsprechend der hellen und dunklen Bereiche. Auf Schraffurebene meint, dass die Platzierung der Linien fest mit dem Objekt verbunden werden muss und sich nicht von Bild zu Bild ändert. Ist diese Bedingung nicht gegeben, dann „springen“ oder auch „schwimmen“ die Linien auf der Oberfläche. Letzteres wird in der Computergraphik als *Shower-Door*-Effekt bezeichnet.

#### Skalierbarkeit

Eng verbunden mit der *Frame*-Kohärenz ist die Skalierbarkeit der Darstellung. Da durch Anordnung der Schraffurlinien und Anpassung ihrer Attribute (Breite, Länge usw.) die Objektbeleuchtung vermittelt wird, muss die dabei entstehende Tönung, beim Skalieren und Zoomen des Objektes oder auch einer Auflösungsänderung in der Darstellung, erhalten bleiben. Dazu zeigt Abb. 2.14 ein Beispiel, in dem in der Mitte die Originalbildgröße

einer Schraffierung dargestellt ist. Rechts und links ist davon eine Verkleinerung um den Faktor zwei abgebildet. Die dunklere Tönung im linken Bild wird durch gleich bleibende Linienanzahl und der damit verbundenen höheren Liniendichte hervorgerufen. Eine jeweilige Änderung in der Linienbreite würde diesen Effekt zwar verringern, wodurch aber Probleme in unterschiedlichen Bildschirmauflösungen entstehen, in Form von „Verklumpungen“ der Schraffurtextur. Daher muss das Schraffierungsverfahren in der Lage sein, die Liniendichte an die jeweilige Auflösung, Objektgröße und -entfernung anzupassen, wie dies in Abb. 2.14 rechts, zu sehen ist.



**Abb. 2.14:** Skalierung der mittleren Illustration, führt bei Nichtanpassung der Liniendichte zu einer veränderten Tönung in der linken Abbildung. Rechts wurde eine geringere Liniendichte simuliert, wodurch eine gleichbleibende Tönung zum Original erreicht wird. Quelle: nach SALISBURY U. A. [1996]

### **Echtzeitfähigkeit**

Eine vorhandene Echtzeitdarstellung der anatomischen Strukturen muss bei ihrer Schraffierung gewährleistet bleiben. Das bedeutet, dass der Berechnungsaufwand für die Schraffurgenerierung einzelner oder mehrerer komplexer Polygonmodelle, wie sie aus hochauflösten Datensätzen entstehen können, gering sein muss, um weiterhin interaktive Framerraten zu ermöglichen. Schraffuren dienen nur als Kontextinformation und sollen daher die Darstellung von Fokusstrukturen nicht verzögern.

### **Automatische Parameterbestimmung**

Da eine Schraffierung der Oberfläche die explorative Analyse der 3D-Visualisierung unterstützen soll, sind Benutzerinteraktionen für die Bilderzeugung durch Parametereinstellungen zu vermeiden. Notwendige Parameter müssen einmal global festgelegt und automatisch aus Modell- oder Metainformationen erzeugt werden können, um bei einer Interaktion reproduzierbare, visuelle Ergebnisse zu erhalten. Dies muss auch für mehrere

unterschiedliche Strukturen in der Darstellung gelten. Damit wird verhindert, dass die Aufmerksamkeit des Betrachters nicht ständig durch die Einstellungen für die Schraffurerzeugung abgelenkt wird.

### 2.3.2 Bildbasierter Ansatz

Bei den bildbasierten Ansätzen werden 2D-Eingabebilder der 3D-Modelle verwendet, auf denen 2D-Schraffierungslinien erzeugt werden. Zwar können bestimmte Modellinformationen im Objektraum gewonnen werden, diese dienen jedoch nur für die Schraffurgenerierung im Bildraum. Für diese hybriden Ansätze wird häufig der von SAITO UND TAKAHASHI [1990] eingeführte *G-Buffer* verwendet, der verschiedene geometrische Informationen, wie Tiefe oder Normalen, sichtbarer Polygonpixel in entsprechenden 2D-Bildern vereint und speichert.

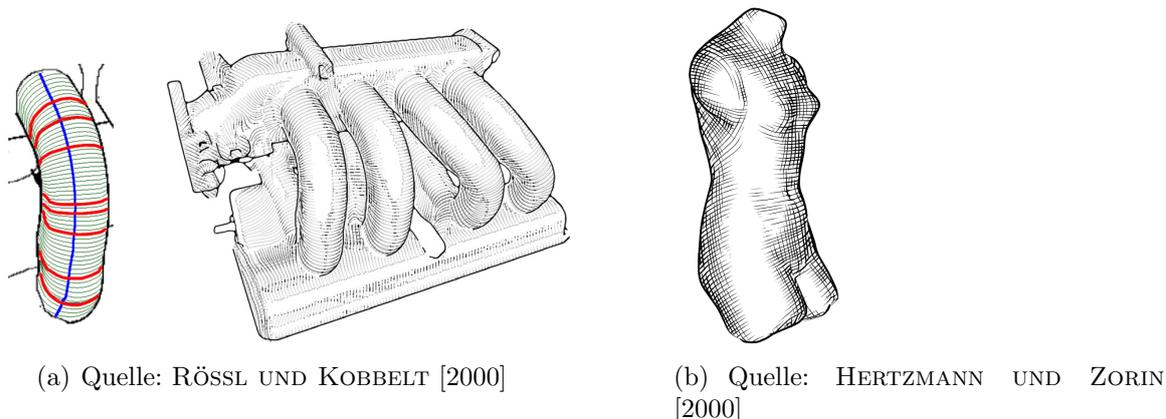
In der Arbeit von RÖSSL UND KOBELT [2000] werden in einem Vorverarbeitungsschritt zunächst krümmungsbasierte Vektorfelder für die Ausrichtung der Schraffurlinien erzeugt und in einem *Enhanced Frame Buffer* abgespeichert. Dieser ist eine Erweiterung des *G-Buffer*, der zusätzlich noch *Shading*-Informationen enthält. Auf Basis des gerenderten Bildes erfolgt eine semi-automatische Segmenteinteilung von Bildbereichen durch den Benutzer, die annähernd ähnliche Krümmungsrichtungen besitzen. Im Anschluss definiert dieser weiterhin eine Referenzlinie (*Backbone*) in jedem Segment und darauf *Seed Points*, von wo aus jeweils Referenzschraffurlinien (*Key-Ribs*) anhand des *Enhanced Frame Buffer* erzeugt werden. Das Vorgehen ist in Abb. 2.15 (a, links) dargestellt, in dem die blaue Linie die *Backbone* darstellt und rot die *Key-Ribs*. Dadurch entstehen überschneidungsfreie und gleichmäßig aufgeteilte Schraffurlinien, die zusätzlich durch Variation der Linienlänge und -breite die Beleuchtung wiedergeben (vgl. Abb. 2.15 (a, rechts)). Nachteilig bei diesem Verfahren ist die aufwendige Benutzerinteraktion zur Schraffurplatzierung, wodurch es sich daher nur für Einzelbilder und nicht für Animationen eignet.

Bei einem ähnlichen Verfahren von HERTZMANN UND ZORIN [2000], wo keine Benutzerinteraktionen mit der Szene erforderlich sind, wird für die Krümmungsapproximation das Modell zunächst im Objektraum geglättet und anschließend das Vektorfeld mit Hilfe eines Optimierungsproblemles für stark abweichende Krümmungsrichtungen gelöst. Nach der Projektion des Feldes in den Bildraum, werden daran 2D-Streamlines ausgerichtet. Das Verfahren erzeugt zwischen zwei Oberflächenpunkten automatisch geodätisch verlaufende Linien<sup>3</sup>, wie sie in Abb. 2.15 (b) zu sehen sind. Die Autoren geben allerdings an, dass für jedes Bild aufwändige Parametereinstellungen notwendig sind, um angemessene Ergebnisse zu erhalten.

Für eine automatische Schraffierung, ebenfalls ohne Benutzerinteraktion, wird in der Arbeit von LAKE U. A. [2000] jedes *Front Facing*-Polygon einer eindeutigen Beleuchtungsstufe zugeordnet und dessen Bildschirmkoordinaten bestimmt. Diese dienen als Textur-

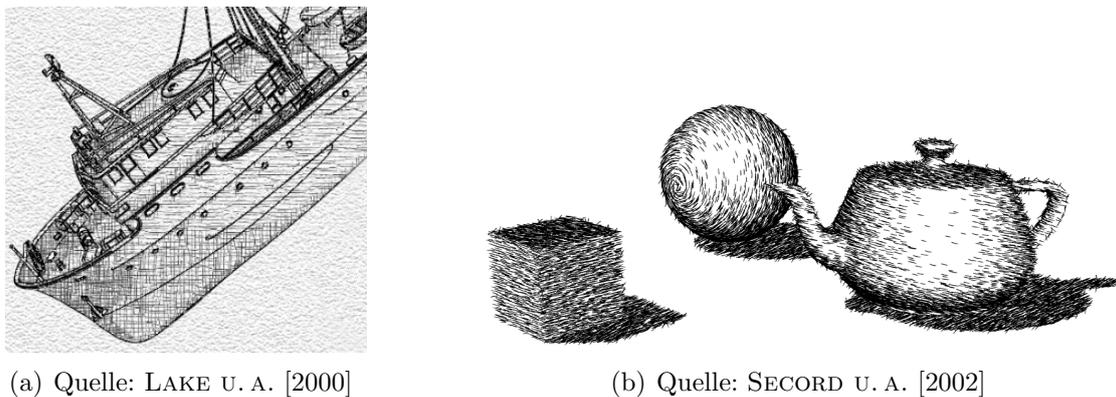
---

<sup>3</sup>Eine geodätische Linie wird als die lokal kürzeste Flächenkurve zwischen zwei Flächenpunkten bezeichnet, ohne die Fläche zu durchdringen.



**Abb. 2.15:** (a) Referenzlinie (blau), initiale (rot) und interpolierte Schraffurlinien (grün) sowie Ergebnisbild bei RÖSSL UND KOBELT [2000] und (b) bei HERTZMANN UND ZORIN [2000].

koordinaten für eine entsprechende Helligkeitstextur, die dann vom Bildschirm auf das Polygon projiziert wird. Durch diesen einfachen Ansatz ist eine echtzeitfähige Darstellung möglich. Aufgrund der Projektion fließen jedoch zum einen keine Krümmungsinformationen des Modells ein, wie dies am Rumpf des Schiffes in Abb. 2.16 (a) zu erkennen ist. Zum anderen sind die Texturen nicht mit dem Objekt verbunden, wodurch der *Shower-Door*-Effekt innerhalb einer Interaktion auftritt. Außerdem ist dieses Verfahren skalierungsabhängig, da die Texturprojektion ebenfalls unabhängig von der Objektgröße erfolgt.

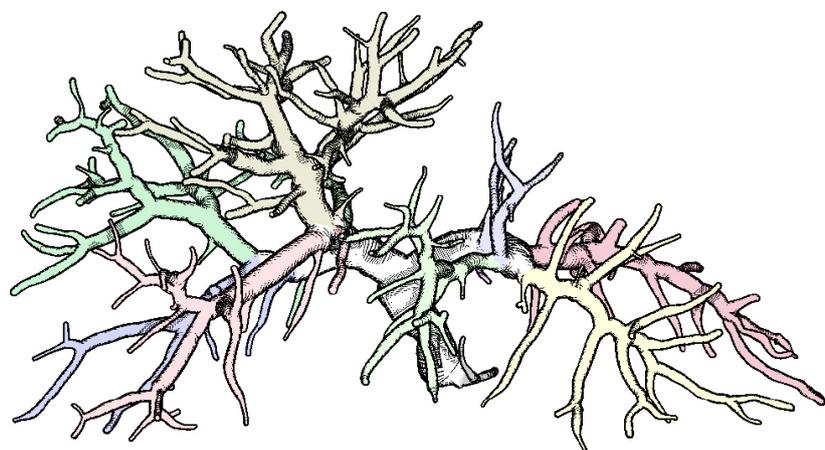


**Abb. 2.16:** Eine Variation unterschiedlicher Helligkeitsstufen wird in (a) durch die Projektion von Helligkeitstexturen und in (b) durch Anordnung einzelner Schraffurprimitive erzeugt.

SECORD U. A. [2002] stellen ein skalierungsunabhängiges Verfahren vor, bei dem Primitive (Punkte oder Schraffurlinien) anhand einer Dichtefunktion ( $PDF = Probability Density Function$ ), welche aus dem Eingangsbild ermittelt wird, auf dem Bild verteilt

werden. Zusätzlich lassen sich Geometrieinformationen aus einem *G-Buffer* auswerten. Dadurch lassen sich kontinuierliche Tönungen erzeugen, wobei hochfrequente Bereiche wie Kanten erhalten bleiben, wie dies in Abb. 2.16 (b) zu erkennen ist. Eine echtzeitfähige Darstellung ist nicht immer möglich, da diese abhängig von der PDF-Erzeugung, Anzahl und Ausrichtung der Primitive ist. Da bei einer Interaktion die PDF für das neu erzeugte Eingangsbild generiert wird und dadurch die Primitive unabhängig von der 3D-Oberfläche und dem vorherigen Bild platziert werden, ist dieses Verfahren zudem nicht *frame*-kohärent. Dies zeigt sich beispielsweise durch das „Wandern“ von Linien über die Oberfläche, wenn sich die Beleuchtung ändert.

In RITTER U. A. [2006] werden stilistische Schraffurlinien auf patientenspezifischen Gefäßmodellen durch Differenzbildung verschiedener *Depth-Buffer* Bilder der Modelle erzeugt (Abb. 2.17). Mit Hilfe von Hardwareshadern ist damit eine echtzeitfähige Darstellung möglich. Zusätzlich lassen sich die Linien, durch Kombination einer *Halftoningtechnik* nach FREUDENBERG U. A. [2004] mit prozedural und im Objektraum erzeugten Schraffurtexturen, auflockern. Die Darstellung der Schraffurlinien gibt allerdings nur die Orientierung der Oberfläche zur Kamera an, wodurch diese nicht *frame*-kohärent ist. Zudem wird durch das Verfahren keine Skalierbarkeit unterstützt. Da die Gefäßmodelle von Skelettlinien zu aneinander gesetzten Kegelstümpfen rekonstruiert werden (HAHN U. A. [2001]), findet eine modellspezifische Generierung von Texturkoordinaten statt. Für Oberflächenvisualisierungen von Patientendatensätzen mit den dabei entstehenden unebenen Oberflächenverläufen, wie sie in der vorliegenden Arbeit verwendet werden, ist die angegebene Koordinatengenerierung nicht anwendbar.

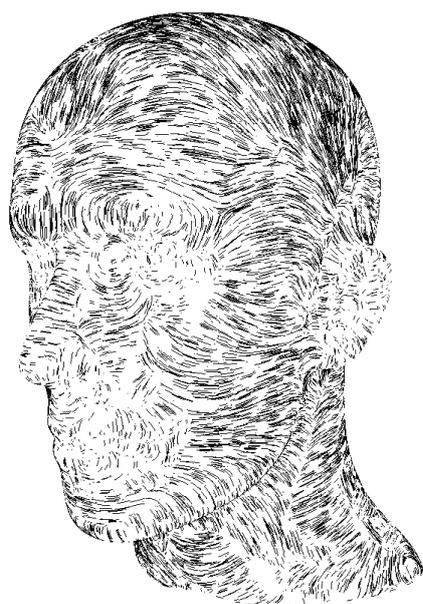


**Abb. 2.17:** Hardwarebeschleunigte Echtzeitschraffierung von patientenspezifischen Gefäßsystemen. Abstandskodierende Schatten visualisieren die Distanz zwischen Gefäßästen durch Variation von Schraffurattributen. Quelle: RITTER U. A. [2006]

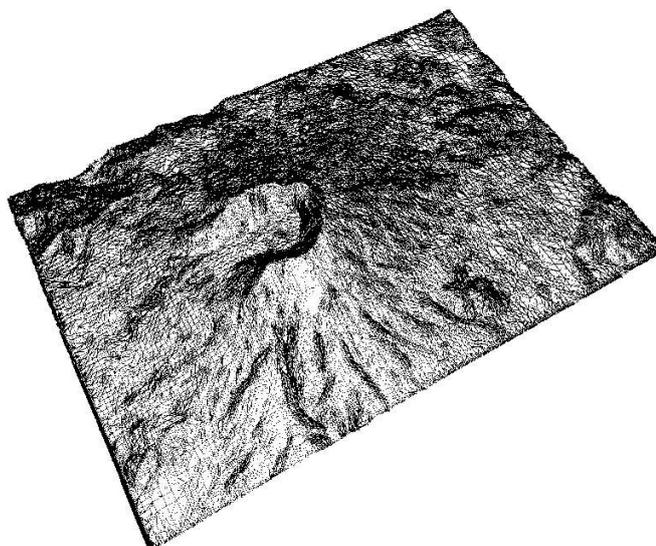
### 2.3.3 Objektbasierter Ansatz

Charakteristisch für objektbasierte Verfahren ist die direkte Schraffierung auf dem 3D-Modell. Die Schraffurlinien sind somit fest mit dem Objekt verbunden. Die Ausrichtung der Linien geschieht dabei über Geometrieinformationen des Modells.

Die Arbeiten von GIRSHICK U. A. [2000] und RÖSSL U. A. [2000] verwenden diesen Ansatz, in dem sie zunächst krümmungsbasierte Vektorfelder berechnen. Die Schraffurlinienerzeugung geschieht bei Girshick et al. durch 3D-Streamlines und bei Rössl et al. durch Polygonlinien. Beide Verfahren erlauben eine interaktive, *frame*-kohärente Darstellung. Die Skalierbarkeit und Tönung ist allerdings nur bei RÖSSL U. A. [2000] möglich und wird über Variation von Liniendichte und -länge realisiert. Dies hat zur Folge, dass bei komplexen, stark schattierten Objekten die Echtzeitfähigkeit nicht mehr gegeben ist, aufgrund der hohen zu verarbeitenden Polygoneometrie der Linien. Bei beiden Arbeiten entstehen zudem Strudel und ungleichmäßige Linienvläufe in Objektbereichen mit starken Krümmungsschwankungen, da die Vektorfelder nicht geglättet sind. Dies ist beispielsweise in den Augen-, Mund- und Ohrenbereichen des Kopfmodells in Abb. 2.18 (a) erkennbar.



(a) Quelle: RÖSSL U. A. [2000]

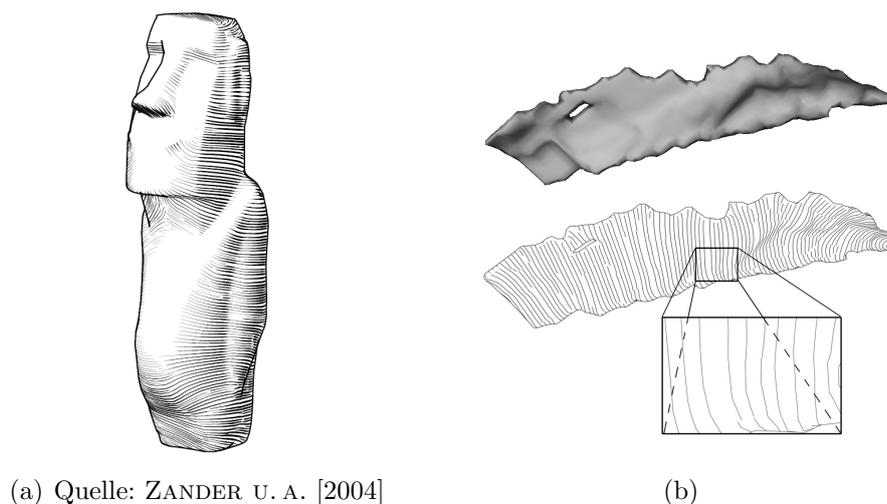


(b) Quelle: SOUSA U. A. [2003]

**Abb. 2.18:** Objektbasierte Schraffierungsergebnisse. In (a) sind dabei die Strudel in Regionen mit hohen Krümmungsschwankungen und in (b) Gitterartefakte bei planaren Bereichen zu erkennen.

ZANDER U. A. [2004] kombiniert die Krümmungsapproximation für jeden Eckpunkt nach RÖSSL U. A. [2000] mit einer Modifikation des Optimierungsproblems von HERTZMANN

UND ZORIN [2000]. Stark abweichende (unsichere) Krümmungsrichtungen werden an stabilen (zuverlässigen) Richtungen ausgerichtet, wodurch bei verhältnismäßig glatten Oberflächen ein geodätisch verlaufendes Vektorfeld entsteht. Zur Visualisierung werden 3D-Streamlines verwendet, die als Polygonzüge auf dem Objekt dargestellt werden und daher eine *frame*-kohärente Darstellung erlauben. Der Benutzer hat über eine sehr anspruchsvolle Parameterisierung die Möglichkeit Linienabstände, -längen, -breiten sowie -transparenzen die Schraffur an die jeweilige Beleuchtung und das ästhetische Empfinden anzupassen (vgl. Abb. 2.19 (a)). Bei der Anwendung auf segmentierten, medizinischen Oberflächen, wie sie in der vorliegenden Arbeit verwendet werden, zeigte es sich allerdings, dass die Schraffurlinien teilweise sichtbare Diskontinuitäten enthalten. Damit eine interaktive Darstellung durch die verwendete Streamlinegeometrie gewährleistet ist, muss zudem die Anzahl der Linien begrenzt werden. Das hat zur Folge, dass bei einer Nahaufnahme des Modells die Skalierbarkeit bezüglich der Beleuchtung nicht gegeben ist, da die Linien auseinander laufen. Die genannten Probleme sind in Abb. 2.19 (b) am Beispiel eines Muskeldatensatzes zur Verdeutlichung noch einmal gezeigt. Weiterhin bedeuten die vielfältigen Parametereinstellungen für die Schraffurlinien für den Anwender jeweils einen erhöhten Zeit- und Konzentrationsaufwand.



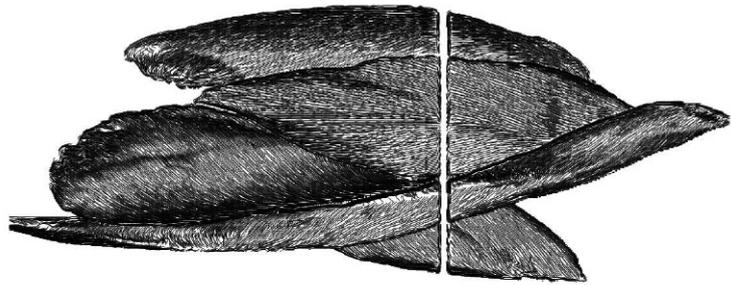
**Abb. 2.19:** (a) Geodätisch und gleichmäßig verlaufende Schraffurlinien bei ZANDER U. A. [2004]. Jedoch zeigt (b) die Sensibilität des Verfahrens bei unebenen Oberflächen und die fehlende Skalierbarkeit bei Vergrößerungen von Objektbereichen (Ausschnitt).

In SOUSA U. A. [2003] wird jede sichtbare Kante ebenfalls als Polygonzug dargestellt, dessen Breite und Stil von der lokalen Beleuchtung, Krümmung und den dazu entsprechenden Skalierungswerten des Benutzers gesteuert werden. Die Darstellung ist durch den festen Objektbezug der Polygonzüge *frame*-kohärent, jedoch müssen bei einer Änderung der Objektentfernung Anpassungen der *Stroke*-Stile über benutzerspezifische Parametereinstellungen getroffen werden. Da die *Strokes* zudem an Polygonkanten erzeugt werden,

treten teilweise Gitter- und Helligkeitsartefakte bei weniger dichten Polygonmodellen auf, wie sie in den planaren Regionen des Landschaftsmodells in Abb. 2.18 (b) zu erkennen sind. Fortgeführt und erweitert wurde dieser Ansatz in der Arbeit von SOUSA U. A. [2004], indem zum einen die *Strokes* anhand vorberechneter Krümmungsrichtungen ausgerichtet werden, wodurch die Gitterartefakte minimiert wurden. Zum anderen wird zu jeder Kante deren aktuelle Beleuchtung bezüglich eines *Spotlights* gespeichert. Darüber wird die Länge der *Strokes* automatisch gesteuert, wodurch sehr ausdrucksstarke Bilder entstehen, wie Abb. 2.20 (a) zeigt. Zusätzliche Variationen im *Stroke*-Stil, wie beispielsweise das Hinzufügen von Welligkeit im Linienverlauf, lassen die Schraffuren handgezeichneter wirken. Jedoch vermerken die Autoren, dass das Ergebnis noch von vielen Einstellungsparametern abhängig ist und der zusätzliche Berechnungsaufwand die echtzeitfähige Darstellung nicht mehr ermöglicht.



(a) Quelle: SOUSA U. A. [2004]



(b) Quelle: DONG U. A. [2003]

**Abb. 2.20:** (a) Expressionistische, echtzeitfähige Schraffierung mit Hilfe des *Edge-Buffer*. (b) Modellbasierte Schraffur entlang der Muskelfasern eines Volumendatensatzes.

Die bisherigen objektbasierten Ansätze verwendeten für die Ausrichtung der Schraffurlinien ein krümmungsbasiertes Vektorfeld. DONG U. A. [2003] stellen ein Verfahren zur automatischen Schraffierung von medizinischen Volumendaten vor, wo das Vektorfeld für Muskelstrukturen aus Modellinformationen erzeugt wird. In einem Vorverarbeitungsschritt wird zunächst an jedem Datenpunkt die Muskelfaserrichtung durch 3D-Bildverarbeitungsoperationen approximiert. Im nächsten Schritt erfolgt an dem Datenpunkt die Definierung der Schraffurlinien. Dazu schneidet eine Ebene, die durch Muskelfaserrichtung und Normale aufgespannt wird, einen Polygonpatch, das die lokale Geometrie an den Punkt approximiert. Anschließend werden nur die resultierenden *Strokes* auf die Bildebene projiziert und gezeichnet, die zum einen den Helligkeitswert des zugehörigen Voxels wiedergeben und zum anderen innerhalb eines benutzerspezifischen Abstandes zur Oberfläche liegen. Dieses Verfahren erzeugt *frame*-kohärente, kontrastreiche Schraffurlinien, die einerseits die Oberflächenform und andererseits auch die Kontraktions- und Dehnungsrichtung der Muskeln beschreiben, wie es in Abb. 2.20 (b) zu sehen ist. Aller-

dings verwenden die Autoren dabei den hochaufgelösten *Visible Human Datensatz* zu Extraktion der feinen Kontrastdetails der Muskelfasern. In den patientenindividuellen Datensätzen ist eine derartige Auflösung nicht gegeben. Aufwendig ist auch die Skalierbarkeit der Helligkeit bezüglich einer Änderung des Kameraabstandes, da die Anzahl der *Strokes* vom Benutzer nachträglich angepasst werden muss. Zudem ist die Darstellung aufgrund der Projektion der *Strokes* in den Bildraum nicht echtzeitfähig, da für diese der jeweilige Abstand zur Strukturoberfläche bei jedem Blickwinkel ermittelt wird.

### 2.3.4 Texturbasierter Ansatz

Bei texturbasierten Ansätzen werden die Schraffurlinien im Objektraum mittels Texturen repräsentiert und sind somit fest mit dem Modell verbunden, wodurch der *Shower-Door*-Effekt vermieden wird. Bei den Texturen können es sich um 2D-Bild- oder Prozedural-texturen sowie 3D-Texturvolumen handeln. Aufgrund der objektbezogenen Texturierung ist eine Parametrisierung der Oberfläche notwendig (vgl. Abschnitt 2.4.2).

PRAUN U. A. [2001] führen für die Echtzeitschraffierung von polygonalen Oberflächen die so genannten *Tonal Art Maps (TAM's)* ein, welche eine Erweiterung der *Art-Maps* von KLEIN U. A. [2000] sind. Eine *Art-Map* ist dabei eine *Mip-Map*-Textur, in der unterschiedliche Auflösungen einer Textur mit gleichem Helligkeitswert gespeichert sind. Eine *TAM*-Textur repräsentiert eine Folge von *Art-Maps* verschiedener Tönungen von Schraffierungsmustern und ist in Abb. 2.21 dargestellt. Zwei wichtige Aspekte sind dabei zum einen die Kohärenz zwischen den einzelnen Skalierungsstufen einer Schraffurtextur, da *Strokes* in niedriger aufgelösten Schraffurtexturen Teilmengen höherer Auflösungsstufen sind. Zum anderen sind hellere Schraffurtexturen stets Teilmenge von dunkleren Texturen, wodurch eine Kohärenz in den Helligkeitsstufen erzeugt wird.

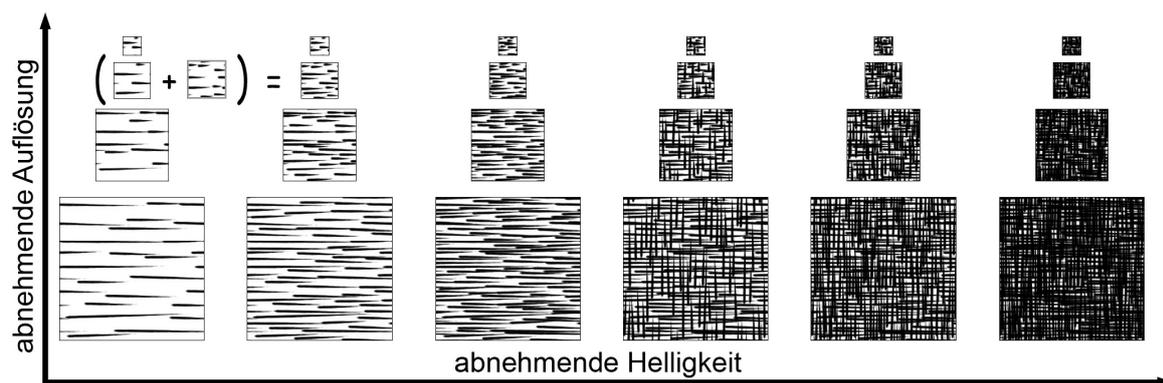
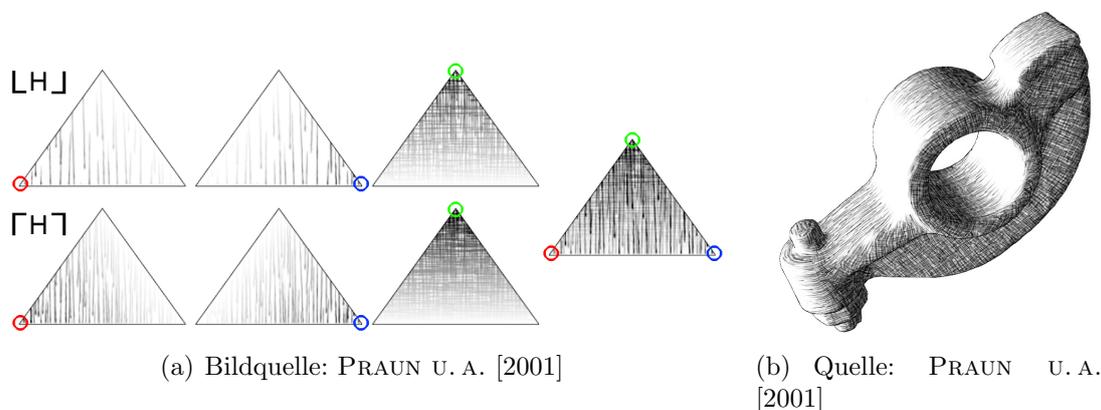


Abb. 2.21: Beispiel einer *TAM*-Textur für die Schraffierung. Bildquelle: PRAUN U. A. [2001]

In den Verfahren von PRAUN U. A. [2001] erhält jeder Eckpunkt eines Dreieckes zwei Texturen, die dessen minimale und maximale Helligkeitsstufe beschreiben. Dies ist in

Abb. 2.22 (a) durch die farbigen Markierungen in der oberen und unteren Reihe veranschaulicht. Die Überblendung über das Dreieck geschieht ähnlich dem *Gouraud-Shading* und wird mit Hilfe von *Hardware-Shadern* inklusive *Multitexturing* berechnet.

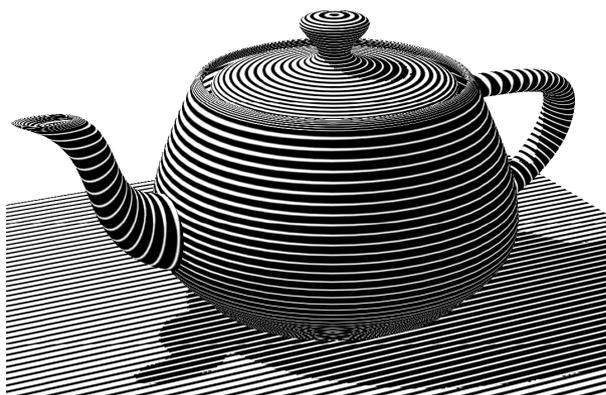


**Abb. 2.22:** (a) Prinzip der Texturinterpolation in einem Dreieck mit den *TAM*-Texturen. Die obere und untere Reihe zeigen jeweils die minimale und maximale Helligkeitstextur  $H$  an jedem Eckpunkt. (b) Ein aus der Interpolation resultierendes Ergebnisbild.

Das Konzept der *TAM*'s stellt eine *frame*-kohärente Darstellung bei der Interaktion mit dem 3D-Modell sicher, da die Linien, je nach Auflösung, Objektabstand und Helligkeit, sanft aus- und eingebledet werden. Für die eigentliche Texturierung wird in einem Vorverarbeitungsschritt zunächst auf einer niedrig aufgelösten Kopie des Eingangsmodells die erste Hauptkrümmungsrichtung für jede Fläche automatisch approximiert. Das daraus resultierende Vektorfeld wird nach einem ähnlichen Verfahren von HERTZMANN UND ZORIN [2000] geglättet und danach durch Resampling auf das originale Eingangsmodell übertragen. Anschließend erfolgt eine lokale Parametrisierung des Modells, indem einzelne Dreiecke zu *Patches* zusammengefasst werden und damit das gesamte Objekt mit separaten Texturstücken überdecken. Eine nähere Betrachtung dieser Parametrisierung erfolgt in Abschnitt 2.4.2. Um die sichtbaren Texturübergänge zu verringern, werden die Texturpatches ineinander überblendet. Zur Laufzeit wird nur die entsprechende *TAM*-Textur für jeden Eckpunkt bestimmt und diese bezüglich benachbarter Eckpunkte interpoliert, wodurch eine automatische und interaktive Darstellung der schraffierten Oberfläche möglich ist. In Abb. 2.22 (b) ist ein Texturergebnis des beschriebenen Verfahrens dargestellt. Probleme treten jedoch bei stark gekrümmten und komplexen Oberflächenverläufen auf, da dort einerseits das Vektorfeld semi-automatisch erstellt werden muss und andererseits kann es durch die Parametrisierung zu Texturdiskontinuitäten und -verzerrungen kommen. Zudem kann der Vorverarbeitungsschritt bei komplexen Modellen mehrere Minuten dauern.

In FREUDENBERG U. A. [2002] wird die Tönung des Objektes durch die Breite der Schraffurlinien erzeugt, anstatt durch die Liniendichte. Dazu verwenden und modifizieren die

Autoren die *Halftoning*-Methode und generieren die Schraffurtexturen prozedural. Dabei ersetzen sie die herkömmliche Stufenfunktion beim *Halftoning* durch eine lineare Schwellwertfunktion und erzeugen damit geglättete sowie unterschiedlich breiten Linien, um die Tönung einer Schraffur zu beschreiben (vgl. Abb. 2.23 (a)). Die *Halftoning*-Methode wird auf Pixelebene im Objektraum und durch *Hardware-Shader* in Echtzeit ausgeführt. Dieser Ansatz benötigt damit keinen zusätzlichen Texturspeicher, setzt allerdings vorher generierte Texturkoordinaten voraus. WEBB U. A. [2002] kombiniert die *Halftoning*-Methode mit dem *TAM* Konzept von PRAUN U. A. [2001] und der Möglichkeit moderner Graphikkarten Volumentexturen zu verarbeiten, um eine genauere Regulierung der Tönung durch Schraffuren zu erzeugen. Dafür verwenden die Autoren eine größere Auswahl von Tönungsstufen und integrieren zusätzlich farbige Schraffierungen. Durch die Texturierung auf Pixelebene ist die Schraffierung unabhängig von der zugrunde liegenden Triangulierung, wodurch Effekte wie die in Abb. 2.23 (b) erzeugte Weltkarte auf einer einfachen Kugel möglich sind. Allein die Helligkeit eines Pixels dient als Entscheidungskriterium ob und wie stark eine Linie gezeichnet wird.



(a) Quelle: FREUDENBERG U. A. [2002]

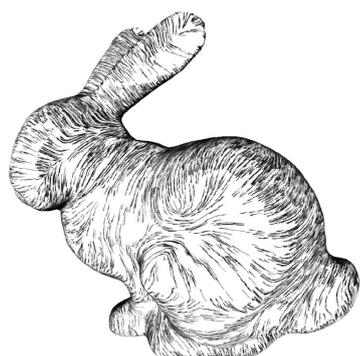


(b) Quelle: WEBB U. A. [2002]

**Abb. 2.23:** (a) Ein modifiziertes *Halftoning* stellt die Helligkeit über die Linienbreite dar und ermöglicht (b) durch Kombination mit höher aufgelösten *TAM*'s feinere Schraffierungen.

Die Arbeit von ECKARDT [2003] benutzt ebenfalls ein hardwarebeschleunigtes *Halftoning* zur Echtzeitschraffierung von polygonalen Oberflächen. Dazu werden jeweils zwei Oberflächenpunkte in den Bildraum projiziert, welche als Referenzpunkte zur Ausrichtung von einer prozedural erzeugten Schraffurlinie dienen. Die Tönung wird über das weiche Ein- und Ausblenden einzelner Texturelemente (Linienzüge) erzielt, indem die Beleuchtung der Referenzpunkte von einer Übergangsfunktion ausgewertet wird. Aufgrund des festen Objektbezuges der Schraffurlinien ist die Darstellung *frame*-kohärent. Eine Variation der Linienbreite ist durch das prozedurale Zeichnen ebenfalls möglich. Allerdings setzt das Verfahren bereits erzeugte Texturkoordinaten voraus.

Für die Hervorhebung der Oberflächenform verwenden GORLA U. A. [2003] Texturen, die mittels Textursynthese auf die Oberfläche abgebildet und dabei entlang einem Vektorfeld orientiert sind. Im Allgemeinen werden Textursyntheseverfahren genutzt, um Löcher in Bilder zu füllen oder neue, größere Texturmuster aus kleineren Textursamples zu erzeugen. In GORLA U. A. [2003] wird zunächst die Polygonoberfläche in Patches aufgeteilt, ähnlich wie bei PRAUN U. A. [2001]. Im Anschluss wird auf jedem Patch die Schraffurtextur pixelweise synthetisiert, was bis zu einigen Minuten an Berechnungsdauer führen kann. Das Verfahren erzeugt gleichmäßige Texturübergänge zwischen den Patches, benötigt geringe Parameterangaben und ist interaktiv (vgl. Abb. 2.24 (a)). Bei Änderung der Kameraentfernung und der damit notwendigen Texturauflösung muss die Berechnung allerdings von neuem durchgeführt werden. DONG UND CLAPWORTHY [2005] erweitern den Ansatz von DONG U. A. [2003], indem sie ebenfalls einen Textursyntheseansatz verwenden, diesen aber für eine 3D-Volumensynthese zur illustrativen Volumenvisualisierung von medizinischen Datensätzen modifizieren (vgl. Abb. 2.24 (b)). Dabei leidet das Verfahren unter den gleichen zeitaufwendigen Berechnungen bei sich ändernden Textursamples wie bei GORLA U. A. [2003].



(a) Quelle: GORLA U. A. [2003]



(b) Quelle: DONG UND CLAPWORTHY [2005]

**Abb. 2.24:** Schraffierungsergebnisse, die mittels Textursynthese auf (a) Polygonoberflächen und (b) Volumendatensätzen erzeugt wurden.

### 2.3.5 Zusammenfassung

Es hat sich gezeigt, dass bildbasierte Ansätze für das hier umzusetzende Verfahren nicht geeignet sind. Ihr Grundproblem ist die fehlende *Frame*-Kohärenz während der Interaktion aufgrund des nicht vorhandenen festen Bezuges von Modell und Schraffurlinien. Diese Anforderung erfüllen die objekt- und texturbasierten Verfahren. Bei Ersteren sind für eine echtzeitfähige Visualisierung allerdings oft effiziente Datenstrukturen notwendig,

die in einem Vorverarbeitungsschritt erzeugt werden müssen. Für die Schraffurgenerierung werden bei ihnen 3D-Geometrien verwendet, wodurch sich teilweise Begrenzungen in der Darstellungsgeschwindigkeit oder Skalierbarkeit ergeben. Zudem erfordern sie, zusammen mit bildbasierten Ansätzen, eine Vielzahl von Parametereingaben. Durch diese lassen sich jedoch sehr individuelle und natürlich wirkende Schraffurlinien erzeugen.

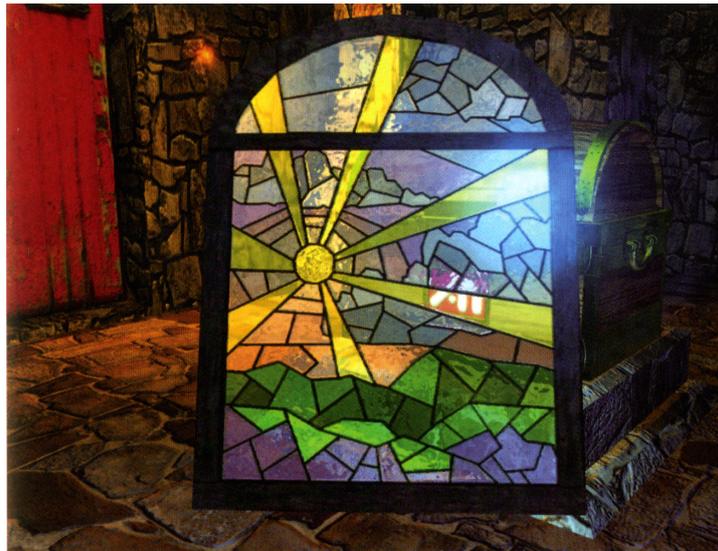
Am besten erfüllen die texturbasierten Ansätze auf polygonalen Oberflächenmodellen die gestellten Anforderungen. Von den vorgestellten Verfahren eignet sich dabei das von PRAUN U. A. [2001] am besten. Deren Parametrisierungsansatz erzeugt einzelne, sich überlappende *Patches*, welche lokal parametrisiert werden und dabei die Richtungsinformationen der Schraffurlinien integrieren. Durch das sanfte Ein- und Ausblenden von Schraffurtexturen mittels *TAM*-Texturkonzept werden *frame*-kohärente und skalierbare Darstellungen erzeugt. Die Echtzeitfähigkeit wird durch den Einsatz moderner Graphikhardware und *Shadern* ermöglicht. Dieser Ansatz soll daher für das hier umzusetzende Verfahren verwendet und weiterentwickelt werden. Modifikationen ergeben sich bezüglich der Erzeugung des Vektorfeldes und der Minimierung der Texturnähte. Letztere treten durch die Überlappungen an den *Patch*-Übergängen auf und werden in dem Verfahren durch Texturüberblendungen minimiert.

Die korrekte Ausrichtung der Schraffurlinien ist bei den vorgestellten Arbeiten ein allgemeines Problem. Die meisten Ansätze erzeugen krümmungsbasierte Vektorfelder, die aber nur bei relativ glatten Oberflächen einen geodätischen Verlauf erzeugen. Auf unebenen Oberflächen kommt es zu Diskontinuitäten oder das Feld muss vom Benutzer nachbearbeitet werden. Zudem wurde im Abschnitt 2.2.3 gezeigt, dass bei Muskelstrukturen die Linien entlang der Kontraktions- und Dehnungsrichtung verlaufen. In diese fließen die lokalen Krümmungsinformationen ein, wodurch die Linienorientierung zusätzlich anhand von Modellwissen bestimmt wird. Dieser Aspekt wurde nur in der Arbeit von RÖSSL UND KOBELT [2000] und DONG U. A. [2003] berücksichtigt.

## 2.4 Texturierung auf 3D-Modellen

Die Texturierung ist in der Computergraphik eine gängige und effiziente Methode, auf z. B. 3D-Oberflächen komplexe und feine Details scheinbar zu erzeugen, ohne sie geometrisch zu modellieren. Dadurch verleihen sie dem Objekt ein realistisches Erscheinungsbild und ermöglichen gleichzeitig eine effiziente Darstellungsgeschwindigkeit, da keine zusätzlichen Geometriedaten erzeugt und verarbeitet werden müssen. Dies spart zudem Modellierungs- und Speicheraufwand. Das dafür verwendete Verfahren wird als *Texture Mapping* bezeichnet, welches eine Textur auf eine Oberfläche abbildet. Eine Textur ist nach HECKBERT [1986] ein mehrdimensionales Bild, das in einen mehrdimensionalen Raum abgebildet wird. Das kleinste Element einer Textur wird als *Texel* (*TEXTure ELe-ment*) bezeichnet und dessen Wert kann verschiedene Beleuchtungs-, Geometrie und Materialparameter eines Oberflächenpunktes beeinflussen oder auch ersetzen. Dazu gehört u. a. die Oberflächenfarbe, Normalenrichtung (*Bump Mapping*), Transparenz, diffuser

Reflektionsanteil (*Environment Mapping*) oder Verschiebung entlang der Normalenrichtung (*Displacement Mapping*). So lassen sich beispielsweise viele natürliche Phänomene wie Lichtreflektion und -brechung simulieren, Materialoberflächen aus Holz oder Stein sowie scheinbare 3D-Strukturen wie Hautfalten auf den Oberflächen erzeugen. In Abb. 2.25 sind mehrere dieser Parameter durch *Texture Mapping* dargestellt, wie die Boden- und Wandfläche oder auch die Lichtbrechung in der Glasscheibe.



**Abb. 2.25:** Beispiel einer computergenerierten 3D-Szene, in der komplexe Materialeigenschaften und Lichtbrechungen durch Texturen simuliert sind. Quelle: AKENINE-MÖLLER UND HAINES [2002]

Eine Textur kann in einem ein-, zwei- oder dreidimensionalen Definitionsbereich und in zwei Repräsentationsformen vorliegen.

- Zum einen als diskrete Textur, wo die Texel als diskrete Werte abgespeichert sind und zu denen Photographien, computergenerierte Bilder oder Zeichnungen gehören. Sie werden am häufigsten bei der Texturierung verwendet, da mit ihnen auf einfache Weise komplexe Muster und Strukturen erzeugt werden können. Sie besitzen allerdings eine beschränkte Auflösung und benötigen, je nach Bildgröße, unterschiedlich viel Speicher für ihre Repräsentation.
- Zum anderen können Texturen prozedural generiert werden, durch die Auswertung einer mathematischen Funktion an jedem Oberflächenpunkt. Bedingt durch ihre mathematische Repräsentation besitzen sie eine uneingeschränkte Auflösung und benötigen keinen zusätzlichen Speicher. Die Definierung geeigneter Funktionen für komplexe Muster ist allerdings sehr schwierig und teilweise unmöglich. Zudem erhöht sich bei steigender Funktionskomplexität der Berechnungsaufwand.

Die weiteren Betrachtungen zur Texturierung beschränken sich von einer Abbildung diskreter 2D-Texturen auf polygonale 3D-Oberflächen, da dies für die weitere Arbeit von Interesse ist.

### 2.4.1 2D-Texture Mapping

Das 2D-*Texture Mapping* beschreibt die Projektion einer 2D-Bildtextur auf eine 3D-Oberfläche. AKENINE-MÖLLER UND HAINES [2002] geben dafür eine Texturierungspipeline an und unterteilen diese in fünf Stufen:

#### 1. Bestimmung der Objektraumposition

Für jeden Eckpunkt  $i$  eines Polygons werden die 3D-Koordinaten  $(x_i, y_i, z_i)$  im Objektraum, in dem die Oberfläche definiert ist, bestimmt (Abb. 2.26 (a)).

#### 2. Parametrisierung der Oberfläche<sup>4</sup>

Die Raumkoordinaten des Eckpunktes  $i$  werden durch eine Projektorfunktion  $\Phi$  in 2D-Koordinaten  $(u_i, v_i)$  des Parameterraumes überführt. Dies ist der *Mapping* Schritt und kann formal durch  $\Phi(x_i, y_i, z_i) \rightarrow (u_i, v_i)$  beschrieben werden. Die Koordinaten  $(u_i, v_i)$  werden als Parameterkoordinaten von  $i$  bezeichnet und ihr Wertebereich liegt zwischen  $[0, 1]$  (Abb. 2.26 (b)). Sie bestimmen den späteren Texturzugriff und damit welcher Texturausschnitt auf welchen Oberflächenbereich abgebildet wird.

#### 3. Umwandlung von Parameterraum in Texturraum<sup>5</sup>

Die Parameterkoordinaten  $(u_i, v_i)$  geben die relative Position eines Eckpunktes  $i$  in der Textur an. Um den genauen *Texel* zu bestimmen, werden diese mit einer Korrespondenzfunktion  $\Psi$  in den Texturraum umgewandelt und ergeben damit einen ganzzahligen Indexzugriff auf die Textur (Abb. 2.26 (c)).

#### 4. Transformation der *Texel*-Werte

Ein *Texel*-Wert kann durch eine Transformation, wie beispielsweise eine Glättungsfilterung, noch verändert werden.

#### 5. Modifikation der Oberflächenparameter

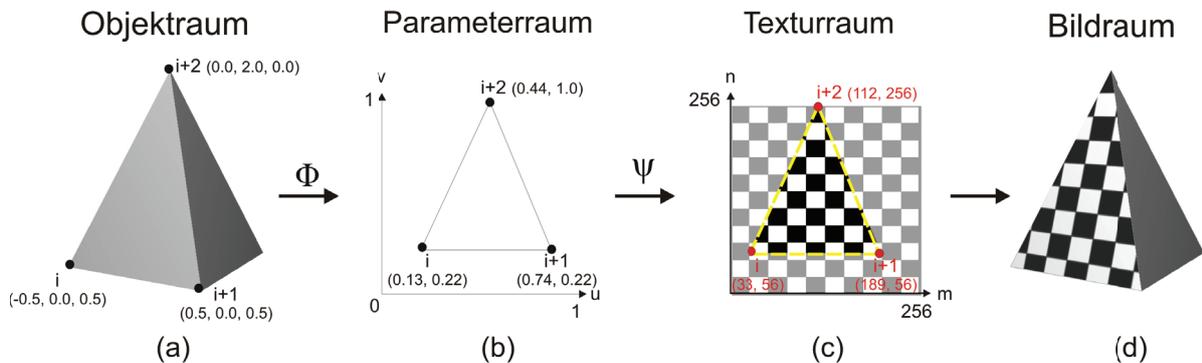
Der *Texel*-Wert modifiziert oder ersetzt bestimmte Parameter des Eckpunktes, wie seine Normalenrichtung oder Oberflächenfarbe, die letztendlich in die Beleuchtungsgleichung einfließen und die endgültige Pixelfarbe bestimmen.

---

<sup>4</sup>An dieser Stellen sei noch erwähnt, dass die Parametrisierung einer Oberfläche neben dem *Texture Mapping* auch noch weitere wichtige Anwendungen hat, wie beispielsweise in der Rekonstruktion von Polygonoberflächen (*Remeshing*) oder dem *Morphing* (GOTSMAN U. A. [2003]).

<sup>5</sup>Die Literatur ist in diesem Punkt nicht eindeutig. Teilweise werden Parameterraum und Texturraum gleichgesetzt, jedoch ist streng genommen hier eine Unterscheidung vorzunehmen.

Für die gesamte Polygonoberfläche werden schließlich die Texturkoordinaten interpoliert und die so texturierte Oberfläche in den Bildraum projiziert, was als *Projection Mapping* bezeichnet wird (Abb. 2.26 (d)). Somit wird jedem Oberflächenpunkt ein korrespondierender Punkt im Texturraum zugewiesen und umgekehrt.



**Abb. 2.26:** Verarbeitungsschritte beim 2D-*Texture Mapping* einer 3D-Oberfläche: (a) Einzelnen Eckpunkten eines Dreieckes im Objektraum werden durch Projektion korrespondierende Parameterkoordinaten zugeordnet (b). Diese stellen eine relative Position in der Textur dar und werden beispielsweise durch die Multiplikation mit der Texturgröße in eindeutige Texturindizes überführt (c). Die umschließenden *Texel* werden auf das Dreieck abgebildet und in den Bildraum projiziert. Die Werte in Klammern veranschaulichen am Beispiel die einzelnen Transformationen.

### Probleme beim Texture Mapping

Beim *Texture Mapping* können jedoch verschiedenen Probleme auftreten, die zu sichtbaren und störenden Texturartefakten führen. Dazu gehören:

- **Texturverzerrungen:**

Diese treten bei Streckung, Stauchung oder Skalierung der zu parametrisierenden Oberfläche durch die Projektorfunktion auf. Es kommt zu Winkel- und Flächenverzerrungen der projizierten Fläche in dem Parameter- beziehungsweise Texturraum, wodurch die Textur bei der Rückprojektion auf das Objekt gestaucht oder gestreckt werden muss. Daher sind eine oder mehrere geeignete Projektorfunktionen erforderlich, die diese Verzerrungen minimieren, worauf der nächste Abschnitt noch näher eingehen wird.

- **Diskontinuitäten an Texturgrenzen:**

Werden Polygonen oder Polygongruppen Texturen zugeordnet, so können sichtbare Nähte (*Seams*) zwischen benachbarten Texturübergängen auftreten. Bei einem kontinuierlichen Verlauf eines Texturmusters über die Oberfläche, wie beispielsweise den Schraffurlinien, ist dieser Effekt nicht gewünscht. Dieses Artefakt tritt auch

auf, wenn der Fall behandelt wird, bei dem eine Fläche außerhalb des Parameterbereiches  $[0, 1]$  projiziert wird. Die Textur wird dann in den meisten Fällen wiederholt auf der Fläche dargestellt. Sind die Texturen dabei nicht kachelfähig aufgebaut, so sind sichtbare Diskontinuitäten an den Texturübergängen zu erkennen.

- **Samplingartefakte:**

In den seltensten Fällen stimmt die Texturauflösung mit der in den Bildraum projizierten und texturierten Polygonfläche überein. Meist muss die Textur gefiltert auf die projizierte Fläche aufgebracht werden. Dabei treten zwei Fälle auf, die als *Magnification* und *Minification* bezeichnet werden. Ersteres tritt auf, wenn die Texturauflösung kleiner als die projizierte Polygonfläche ist. Die Textur wird dabei vergrößert, wodurch blockartige Artefakte (*Pixelation*) auftreten. Durch bilineares Filtering der Textur lässt sich dieser Effekt minimieren. *Minification* tritt auf, wenn die Texturauflösung größer als die projizierte Polygonfläche ist, so dass mehrere *Texel* einen Pixel überdecken. Die Schwierigkeit besteht dabei in der Auswahl und dem Einfluß der *Texel* für den Pixel. Eine gängige Lösungsmethode ist die Reduktion der Texturauflösung, welche beispielsweise durch *Mip Mapping* erreicht wird. Dabei wird eine Textur in verschiedenen Auflösungen erzeugt, in einer Art Texturpyramide gespeichert und je nach Auflösung der projizierten Polygonfläche erfolgt die Auswahl der passenden Texturgröße.

## 2.4.2 Parametrisierung

Wie aus dem vorhergehenden Abschnitt ersichtlich, ist eine geeignete Parametrisierung der Oberfläche zur Vermeidung von Texturverzerrungen erforderlich. Nach FLOATER UND HORMANN [2004] handelt es sich bei einer Parametrisierung um die bijektive Abbildung einer Oberfläche  $S$  auf eine zweite Oberfläche  $S^*$ , beschrieben durch eine Abbildungsfunktion (Projektorfunktion)  $f$ . Die Parametrisierung eines Punktes  $p \in S$  ist dabei so definiert, dass dieser die gleichen Koordinaten hat wie sein Bildpunkt  $f(p) \in S^*$ . Die Abbildung  $f$  ist zulässig, wenn die Parametrisierung umkehrbar eindeutig ist. Bezüglich dem 2D-*Texture Mapping* auf einer polygonalen Oberfläche  $S_T \in \mathbb{R}^3$ , entspricht die Parametrisierung der Suche nach einer geeigneten, stückweise linearen und kontinuierlichen Abbildung  $f : S_T(x, y, z) \leftrightarrow S^*(u, v)$  mit  $S^* \in \mathbb{R}^2$ , die ebenfalls polygonal ist.

Um Texturverzerrung zu vermeiden, wird eine isometrische (längentreue) Abbildung gesucht, da diese konform (d. h. winkeltreu) und flächentreu ist. In der Praxis ist eine derartige Abbildung, außer bei planar abwickelbaren Oberflächen, wie beispielsweise bei einem offenen Zylinder, nur sehr selten möglich (AKENINE-MÖLLER UND HAINES [2002]). Dies trifft gerade auf Polygonmodelle mit komplexer Geometrie zu, wo versucht wird, eine einzige Projektorfunktion für eine isometrische Abbildung der ganzen Oberfläche zu finden. Daher wird eine konforme (winkeltreue), flächentreue oder verzerrungsminimierte Kombination dieser beiden Abbildungen gesucht.

Das Finden einer geeigneten Oberflächenparametrisierung für die Texturierung, unter den oben beschriebenen Anforderungen, ist ein breites Forschungsgebiet in der Computergraphik. Da in dieser Arbeit polygonale Oberflächen texturiert werden sollen, erfolgt für deren Parametrisierung im Folgenden eine Vorstellung von zwei grundsätzlichen Ansätzen, die für diese Arbeit in indirekte und direkte Parametrisierung unterschieden werden.

### Indirekte Parametrisierung

Beim Ansatz der indirekten Parametrisierung wird eine so genannte Zwei-Phasen Projektion von BIER UND SLOAN [1986] durchgeführt. In der ersten Phase wird die Textur auf ein 3D-Zwischenobjekt  $S$  projiziert mit  $T(u, v) \rightarrow S(x_i, y_i, z_i)$ , das sich mathematisch einfach parametrisieren lässt und das eigentlich zu texturierende Objekt  $O$  umschließt. In der zweiten Phase erfolgt eine Abbildung der projizierten Textur von  $S$  auf  $O$  mit  $S(x_i, y_i, z_i) \rightarrow O(x, y, z)$ . Als Zwischenobjekt finden dabei häufig Kugel, Zylinder oder Quader Verwendung, deren Projektion jeweils als *Sphere*, *Cylinder* oder *Cube Mapping* bezeichnet wird. Ein Zylinder lässt sich beispielsweise durch:  $T(u, v) = (r \cos(2\pi u), r \sin(2\pi u), v/h)$  parametrisch beschreiben, wobei  $r$  und  $h$  jeweils den Radius und die Höhe des Zylinders angeben. Für die zweite Phase der Texturierung gibt es verschiedene Strategien, wie die Texturabbildung von  $S$  auf  $O$  erfolgt, basierend auf den Schnitt eines Objektstrahles mit dem Zwischenobjekt. Die Wahl des Zwischenobjektes hängt von der Form des zu umschließenden Objektes ab und sollte so gewählt werden, das es dieses bestmöglich umschließt, um Texturverzerrungen zu minimieren. Je komplexer die Objektform ist und damit abweichender vom Zwischenobjekt oder weiter entfernter das Objekt von diesem ist, desto stärker treten diese Verzerrungen auf. Zur Minimierung dieser Artefakte kann zum einen das komplexe Objekt in einfachere Objektformen aufgeteilt werden, an die sich dann jeweils ein passendes Zwischenobjekt anwenden lässt. Dabei können jedoch Texturnähte an den Übergängen entstehen. Zum anderen führt TARINI U. A. [2004] sogenannte *Polycubes* ein, wo das ganze Objekt in Einheitswürfel unterteilt wird und diese dann für ein *Cube Mapping* verwendet werden. Die Anzahl der Unterteilungen steigt mit der Komplexität des Objektes, um Verzerrungen zu minimieren. Damit verbunden ist allerdings auch ein erhöhter Texturspeicherbedarf, da für jeden Einheitswürfel eine eigene Textur verwendet wird (TARINI U. A. [2004], BAER [2005]).

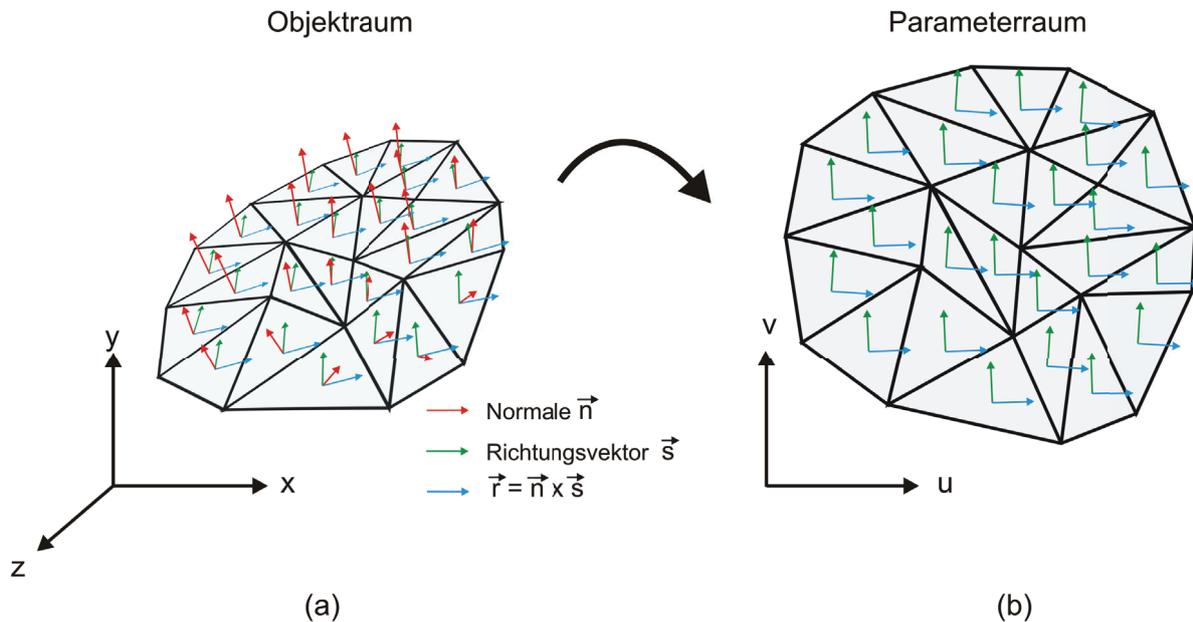
### Direkte Parametrisierung

Beim direkten Parametrisieren wird jeder Eckpunkt direkt in den Parameterraum überführt ohne die Texturabbildung von einem Zwischenobjekt. Ein Ansatz besteht in der Faltung (*Unfolding*) oder Projektion (*Flattening*) der 3D-Polygonfläche in eine planare Polygonfläche. Dabei wird die einfache Parametrisierung eines planaren Polygons, genauer gesagt eines Dreieckes, ausgenutzt. Die Abbildung von den  $\mathbb{R}^3$  in den  $\mathbb{R}^2$  erfolgt entweder durch die Rotation eines Dreieckes um die gemeinsame Kante seines Nachbardreieckes in

dessen Ebene (SAMEK U. A. [1986]) oder durch Projektion jedes Dreieckes in die Ebene (FLOATER [1997]). Um dabei Dreiecksüberlappungen oder starke Verzerrungen an hohen Krümmungsbereichen zu vermeiden, wird das gesamte Objekt oft in einzelne, annähernd planare Bereiche partitioniert, so genannte *Patches* oder auch *Charts*, welche dann einzeln in die Ebene überführt werden. In der Ebene wird anschließend der Texturraum auf die Polygonfläche projiziert und jedem Eckpunkt eine entsprechende  $(u, v)$ -Koordinate zugewiesen. In diesem Zusammenhang werden auch Texturatlantenteile eingesetzt, wo in einer 2D-Textur einzelne Texturen für die *Patches* abgespeichert sind.

Der Vorteil dieses Ansatzes liegt in der einfachen und dadurch schnellen Texturkoordinatenzuweisung, wobei es jedoch durch die Schnittkanten zu Diskontinuitäten an den *Patch*-Übergängen und damit zu Texturnähten kommt. Außerdem sind teilweise Benutzerinteraktionen notwendig für die Definierung von Schnittkanten bei komplexen Objekten sowie für die Verwendung von Texturatlantenteilen bei der Ausrichtung der darin abgebildeten planaren *Patches*. PRAUN U. A. [2000] verwenden diesen Ansatz bei der Parametrisierung beliebiger Oberflächen für die echtzeitfähige Texturierung sich wiederholender Texturmuster (*Lapped Textures*). PRAUN U. A. [2001] überträgt diesen Ansatz für die Oberflächenschraffierung. Die Partitionierung erfolgt automatisch, indem um ein nicht zugewiesenes Dreieck (Referenzdreieck) das *Patch* durch das Hinzufügen benachbarter Dreiecke wächst, wobei es bei jedem Wachstumsschritt in den Texturraum projiziert wird. Zusätzlich fließt das zugrunde liegende Vektorfeld in die Parametrisierung mit ein. Für jedes Patchdreieck ergibt sich ein lokales Koordinatensystem, welches aus Richtungsvektor  $\hat{s}$ , der Flächennormale  $\hat{n}$  und dem Kreuzprodukt  $\hat{r}$  der beiden ersten Vektoren entsteht (vgl. Abb. 2.27 (a)). Über ein Optimierungsproblem werden die  $\hat{s}$  und  $\hat{r}$  Vektoren der im *Patch* enthaltenen Dreieckes anschließend an die  $u$ - und  $v$ -Achsen des Parameterraumes ausgerichtet (vgl. Abb. 2.27 (b)). Die dabei entstehenden 2D-Koordinaten für jeden Eckpunkt entsprechen dessen Parameterkoordinaten. Das Patchwachstum bricht ab, wenn einerseits das Texturmuster vollständig überlagert ist, wodurch sich die einzelnen Texturpatches auf dem Modell überlagern können. Andererseits, wenn bei der Projektion zu starke Texturverzerrungen entstehen, wodurch es zu harten Texturübergängen kommen kann. Um die Nahtartefakte zu minimieren, werden die Texturpatches an den Grenzen ineinander überblendet.

Ein zweiter relativ neuer Ansatz direkter Parametrisierung besteht in der Abbildung geschlossener und löcherloser (Genus-0) Dreiecksnetze auf eine kontinuierliche Kugel (*Spherical Parameterization*). Dabei wird einerseits die topologische Äquivalenz dieser Oberflächen zu Kugeln ausgenutzt und andererseits lässt sich die mathematisch einfache Parametrisierung einer Kugel auf die abgebildeten Eckpunkte anwenden. Im Vergleich zum vorhergehenden Ansatz treten dadurch weniger Verzerrungen auf und es wird keine Partitionierung benötigt, wodurch keine Nähte entstehen (GOTSMAN U. A. [2003], PRAUN UND HOPPE [2003]). Eine Herausforderung ist dabei das Finden einer geeigneten Abbildung von einer komplexen und stark gekrümmten Oberfläche auf die Kugel, so dass Überlappungen sowie konforme Verzerrungen vermieden beziehungsweise gemindert werden. In diesem Zusammenhang entstehen weitere Probleme mit numerischen Unge-



**Abb. 2.27:** (a) Darstellung eines *Patch* im Objektraum mit zugehörigem lokalem Koordinatensystem für jedes Dreieck. (b) Projiziertes *Patch* in den Texturraum, wobei die  $\hat{s}$ - und  $\hat{r}$ -Achsen jedes Dreieckes an die  $u$ - und  $v$ -Achsen des Parameterraumes ausgerichtet sind.

naugigkeiten und die Verfahren besitzen einen relativ hohen Berechnungsaufwand, der einige Minuten beträgt (PRAUN UND HOPPE [2003]). Des weiteren fließen keine lokalen Richtungsinformationen in die Parametrisierung ein, aufgrund der globalen Abbildung. Letztlich kann dieser Ansatz nur auf löcherlose Dreiecksnetze angewendet werden. Für die Parametrisierung von Polygonoberflächen mit beliebigem Genus wird der Partitionierungsansatz verwendet (FLOATER UND HORMANN [2004]).

### Zusammenfassung

Für eine kontinuierliche und verzerrungsfreie Texturierung ist eine isometrische, bijektive Parametrisierung der Dreiecksfläche notwendig. Es hat sich allerdings gezeigt, dass für beliebig geformte Oberflächen, wie sie auch bei medizinischen Oberflächenrekonstruktionen entstehen, kein allgemeiner Ansatz existiert. Die verschiedenen Verfahren versuchen daher möglichst konforme und flächentreue Abbildungen zu erreichen. Die indirekte Parametrisierung eignet sich dafür nur, wenn die Objektform dem Zwischenobjekt ähnelt, da es sonst zu starken Verzerrungen kommt. Das *Polycube*-Verfahren würde diesen Effekt vermindern, jedoch ist eine Anwendung aus zwei Gründen nicht geeignet. Zum einen wäre für die Anpassung an die anatomischen Oberflächenmodelle eine hohe *Cube*-Auflösung notwendig und damit ein erhöhter Texturspeicherbedarf. Zum anderen können keine Richtungsinformationen in die Parametrisierung einfließen, wonach sich

die Ausrichtung der Schraffurtexturen orientiert. Die einzelnen *Cubes* müssten entsprechend der lokalen Richtungen rotiert werden, was einerseits zu Diskontinuitäten an den Würfelübergängen führt und andererseits deren Konzept der achsenparallelen Würfelseitenausrichtung nicht entspricht.

Die direkte Parametrisierung durch Abbildung auf eine Kugel eignet sich ebenfalls nicht für die verwendeten Modelle, da diese durchaus Löcher enthalten können. Weiterhin scheitert dieser Ansatz bei der Anwendung von anisotropen Texturen, die an einem Vektorfeld ausgerichtet werden sollen. Zum einen fließt dieses in die globale Parametrisierung nicht mit ein, so dass keine lokale Anpassung der Texturorientierung erfolgt. Zum anderen muss die Textur vorher verzerrt werden, damit sie bei der Rückprojektion von der Kugel auf das Objekt unverzerrt abgebildet wird.

Bei der Partitionierung der Oberfläche in *Patches* tritt das Problem von Nähten und Verzerrungen auf. Jedoch zeigen die Arbeiten von PRAUN U. A. [2000] und PRAUN U. A. [2001], dass durch den Einsatz von *Hardware-Shadern* für die Texturierung die sichtbaren Nahtartefakte vermindert werden können. Zusätzlich fließt ein zugrunde liegendes Vektorfeld in die Parametrisierung ein, wodurch eine lokale Ausrichtung der Texturen erfolgt.

## 2.5 Zusammenfassung und Schlussfolgerungen

Medizinische Illustrationen haben sich über die Jahrhunderte zur Darstellung anatomischer Sachverhalte und Verwendung in der Lehre bewährt. Die dabei verwendeten illustrativen Darstellungstechniken ermöglichen zum einen artefaktfreie und abstrahierte Abbildungen komplexer Zusammenhänge. Zum anderen erlauben sie eine frei wählbare Sichtbarkeit sowie Hervorhebung oder Abschwächung darzustellender Objektbereiche. Eine effektive und seit langem eingesetzte Darstellungstechnik ist dabei die Schraffur, welche in diesem Kapitel u. a. untersucht und bezüglich der Anwendung in anatomischen Illustrationen analysiert wurde. In diesem Zusammenhang wurden verschiedene Gestaltungsparameter herausgearbeitet und es konnte gezeigt werden, dass Schraffuren Informationen über den Oberflächenverlauf vermitteln. Dabei ist die Wahrnehmung am genauesten, wenn die Linien entlang der beiden Hauptkrümmungsrichtungen verlaufen. Von Vorteil ist dieser Aspekt beispielsweise bei der Beurteilung der räumlichen Beziehung von eingebetteten Oberflächenvisualisierungen. Eine halbtransparente Darstellung in Kombination mit Schraffurlinien des umschließenden Objektes bietet dabei einen besseren Oberflächeneindruck als eine ausschließlich halbtransparente Darstellung.

Die Untersuchung anatomischer Atlanten und Illustrationen zeigten in diesem Kapitel, dass vorrangig Gefäße und Muskelstrukturen schraffiert werden. Bei ersteren verlaufen die Schraffurlinien entlang der ersten Hauptkrümmungsrichtungen, d. h. radial auf der Gefäßoberfläche. Für Muskelstrukturen entspricht der Linienverlauf vorrangig der Kontraktions- und Dehnungsrichtung, wobei allerdings das lokale Krümmungsverhalten

mit einfließt. Aus dieser Analyse heraus lassen sich zwei Schlussfolgerungen für ein computergeneriertes Schraffierungsverfahren ableiten. Zum einen ist es sinnvoll ebenfalls eine Beschränkung der zu schraffierenden Strukturen vorzunehmen, da dies für den Mediziner am vertrautesten und dadurch anwendungsorientierter ist. Zum anderen ist die Erzeugung eines krümmungs- und modellbasierten Vektorfeldes notwendig, welches abhängig von der zugrunde liegenden Struktur ist. An diesem orientieren sich die späteren Schraffurlinien.

Des Weiteren wurden in diesem Kapitel bild-, objekt- und texturbasierte Verfahren zur computergenerierten Schraffierung untersucht. Dabei zeigt sich, dass ein texturbasierter Ansatz für das hier umzusetzende Verfahren am besten geeignet ist. Es stellt die notwendige *Frame*-Kohärenz, Echtzeitfähigkeit und Skalierbarkeit sicher, die in den anderen Ansätzen nur teilweise erfüllt werden. Der Einsatz von Texturen erfordert jedoch eine möglichst verzerrungsfreie Parametrisierung der Oberfläche. Aufgrund des komplexen Oberflächenverlaufes der in dieser Arbeit verwendeten patientenspezifischen Datensätze, genügen die Verfahren für eine indirekte Parametrisierung dieser Anforderung nicht. Die direkte Parametrisierung durch Aufteilung der Oberfläche in *Patches* und deren Ausrichtung an das Vektorfeld nach PRAUN U. A. [2000] stellt eine möglichst isomorphe Abbildung sicher. Dieser Ansatz wird daher in Kombination mit dem von PRAUN U. A. [2001] eingeführten *TAM*-Texturen zur Sicherstellung der Skalierbarkeit weiterverfolgt. Bevor jedoch auf den Entwurf des eigenen Verfahrens eingegangen wird, erfolgt im nächsten Kapitel eine mathematische Betrachtung von Oberflächenkrümmung und Vorstellung einiger Approximationsverfahren, die diese auf diskreten Oberflächen bestimmen.

## 3 Krümmungsapproximation auf 3D-Modellen

Die meisten betrachteten Schraffierungsverfahren aus Abschnitt 2.3 verwenden krümmungsbasierte Vektorfelder zur Ausrichtung der Schraffurlinien. Dafür werden an jedem Oberflächenpunkt Krümmungsinformationen benötigt, die das lokale Krümmungsverhalten in Form von Richtungsvektoren beschreiben. Für die vorliegende Arbeit wird der Einsatz von Vektorfeldern ebenfalls aufgegriffen und erweitert (vgl. Abschnitt 4.3). Dieser Abschnitt geht dazu auf die mathematischen Grundlagen von Krümmungsbestimmungen auf Oberflächen ein. Darauf aufbauend werden anschließend überblicksartig Approximationsverfahren beschrieben, die diese Informationen auf polygonalen Netzen bestimmen. Für diese Arbeit liegen diese in Form von Dreiecksnetzen vor. Die Verfahren beziehen sich dabei auf die Arbeit von SCHULZ [2005], in der verschiedene Approximationsmethoden umgesetzt und untersucht wurden, inwiefern sie sich auf anatomische Oberflächenmodelle anwenden lassen.

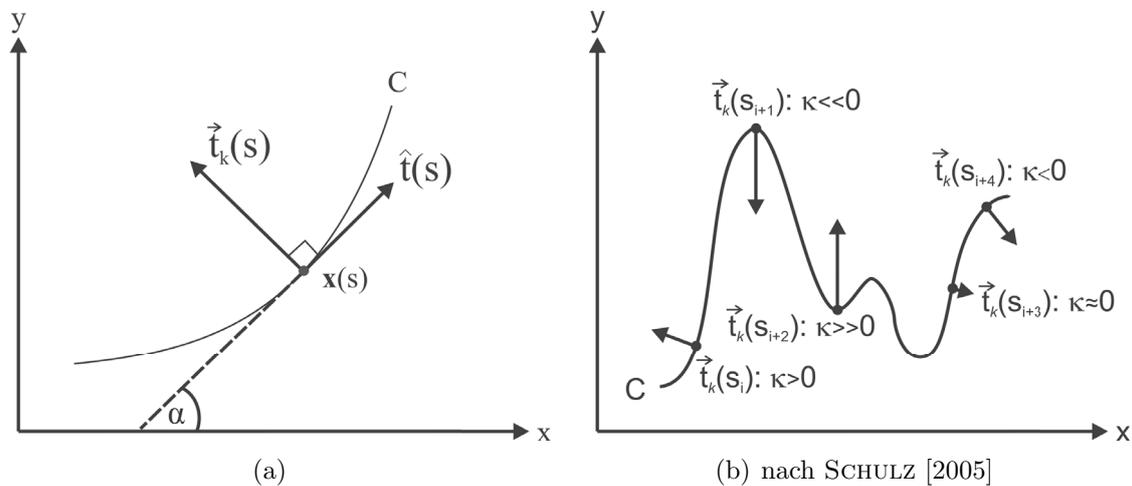
### 3.1 Mathematische Grundlagen

Analytisch beschriebene Kurven und Flächen können durch Ableitungen erster und zweiter Ordnung charakterisiert werden. Krümmungsinformationen an einen Punkt  $Q$  lassen sich dabei durch die zweite Ableitung an  $Q$  gewinnen und beschreiben dessen lokales Krümmungsverhalten bezüglich seiner Umgebung. Je nachdem, ob Krümmungsinformationen für Kurven oder Flächen bestimmt werden sollen, beinhalten sie verschiedene Krümmungsgrößen. Es erfolgt zuerst eine Betrachtung von Ebenenkurven, wo zum Begriff der *Krümmung*, als eine Krümmungsgröße, hingeführt wird und anschließend die Erweiterung auf Flächen. Letzteres beinhaltet die für diese Arbeit notwendigen weiteren Krümmungsgrößen. Folgenden mathematische Notationen werden diesbezüglich festgelegt, die auch für die weitere Arbeit gelten: Ein Vektor ist mit einem „ $\rightarrow$ “ und seine Normierung durch ein „ $\wedge$ “ gekennzeichnet. Punkte auf einer Fläche oder Linie sowie Dreieckseckpunkte werden als indizierte  $p$  bzw.  $v$  bezeichnet. Das Skalarprodukt wird durch „ $\cdot$ “ ausgedrückt.

#### 3.1.1 Krümmung auf einer Kurve

Es sei  $\mathbf{x} = \mathbf{x}(s)$  eine Parameterdarstellung einer Ebenenkurve  $C$ . So beschreibt die erste Ableitung  $d\mathbf{x}/ds = \mathbf{x}'$  die Tangente  $\vec{t}(s)$  an  $C$  im Punkt  $\mathbf{x}(s)$ . Der dazugehörige normierte Tangenteneinheitsvektor sei  $\hat{t}(s)$ , mit  $\mathbf{x}'/|\mathbf{x}'|$ . Die Ableitung von  $\hat{t}(s)$  ist dann

$d\hat{t}/ds = \hat{t}'(s) = \mathbf{x}''(s)$  und bezeichnet den Krümmungsvektor  $\vec{t}_\kappa$ . Ist dieser verschieden vom Nullvektor  $\vec{0}$ , so zeigt  $\vec{t}_\kappa(s)$  im Punkt  $\mathbf{x}(s)$  in die Richtung, in die sich die Kurve krümmt und steht senkrecht zu  $\hat{t}(s)$  (vgl. LIPSCHUTZ [1980]). Der Betrag von  $\vec{t}_\kappa(s)$  wird mit  $\pm|\vec{t}_\kappa(s)|$  bezeichnet und gibt die Krümmung  $\kappa$  an. Sie ist ein Maß für die Änderungsgeschwindigkeit der Tangentenrichtung an der Stelle  $\mathbf{x}(s)$  und steigt je größer die Richtungsänderung ist. In Abb. 3.1 (a) wird dieser Zusammenhang am Beispiel einer Ebenenkurve verdeutlicht. An der Stelle  $\mathbf{x}(s)$  ist der zugehörige Tangenteneinheits- und Krümmungsvektor abgebildet. Zusätzlich ist der Winkel  $\alpha$  zwischen  $\hat{t}(s)$  und der positiven x-Achse dargestellt. Die Krümmung ist positiv definiert, wenn der Winkel mit wachsendem Parameter  $s$  ebenfalls steigt und negativ, wenn der Winkel kleiner wird. Zur Verdeutlichung zeigt dazu Abb. 3.1 (b) verschiedene Stellen auf einer Ebenenkurve mit Angabe des lokalen Krümmungsvektors und zugehöriger, relativer Krümmung.



**Abb. 3.1:** (a) Darstellung des Krümmungsvektors  $\vec{t}_\kappa(s)$  in Beziehung zum Tangentenvektor  $\hat{t}(s)$  an einen Kurvenpunkt  $x(s)$  mit positiver Krümmung. (b) Beispiele von sich variierende Krümmungsvektoren und deren Betrag an verschiedenen Kurvenpunkten mit steigendem  $s$ .

### 3.1.2 Krümmung auf einer Fläche

Das Krümmungsverhalten eines Oberflächenpunktes  $p$  auf einer Fläche  $S$  wird bestimmt durch das Krümmungsverhalten aller Flächenkurven  $C$ , die durch  $p$  verlaufen. Das bedeutet, dass die Krümmungsinformationen von Ebenenkurven übertragen werden, auf jede Kurve  $C$ , die  $p$  berührt und in  $S$  liegt. Es lassen sich daraus weitere Krümmungsgrößen ableiten, die das lokale Krümmungsverhalten charakterisieren, von denen die folgenden die wichtigsten sind:

- *Hauptkrümmungen*  $\kappa_1$  und  $\kappa_2$ ,

- Hauptkrümmungsrichtungen  $\hat{t}_{\kappa 1}$  und  $\hat{t}_{\kappa 2}$ ,
- mittlere Krümmung  $H$  sowie
- Gauß'sche Krümmung  $K$ .

Es erfolgt eine kurze Herleitung dieser Größen und Erläuterung ihres Einflusses auf eine Klassifizierung der Oberflächentypen.

Die lokale Krümmung  $\kappa$  ist durch die Normalenkrümmung  $\kappa_N$  und geodätische Krümmung  $\kappa_G$  an  $p$  beschreibbar. An dieser Stelle soll zunächst nur  $\kappa_N$  weiter betrachtet werden. Auf  $\kappa_G$  wird bei den Approximationsverfahren Bezug genommen. Die Normalenkrümmung ist definiert durch die Projektion des Krümmungsvektors  $\vec{t}_\kappa$  auf die Flächennormale  $\hat{n}$  von  $p$  (vgl. Abb. 3.2 (a)) mit

$$\kappa_N = \vec{t}_\kappa \cdot \hat{n} \quad (3.1)$$

und dem dazugehörigen Normalenkrümmungsvektor

$$\vec{t}_{\kappa N} = \kappa_N \hat{n}. \quad (3.2)$$

Die Normalenkrümmung ist abhängig von der Richtung der Tangente an  $p$  und entspricht der Krümmung der Kurve  $C$ , die durch den Normalenschnitt von  $S$  an  $p$  definiert ist. Die Normalenschnittebene  $\epsilon(\hat{n})$  wird aufgespannt durch  $\hat{n}$  und der Tangente an  $p$ .

Mit Hilfe der Weingartenabbildung  $W$  und gegebener Tangentenrichtung kann die Normalenkrümmung auf einer parametrisierten Fläche  $S$  im  $R^3$  bestimmt werden (vgl. SCHULZ [2005]). Ein Oberflächentyp  $p$  entspricht dabei der Abbildung  $x : R \times R \rightarrow R^3$  mit

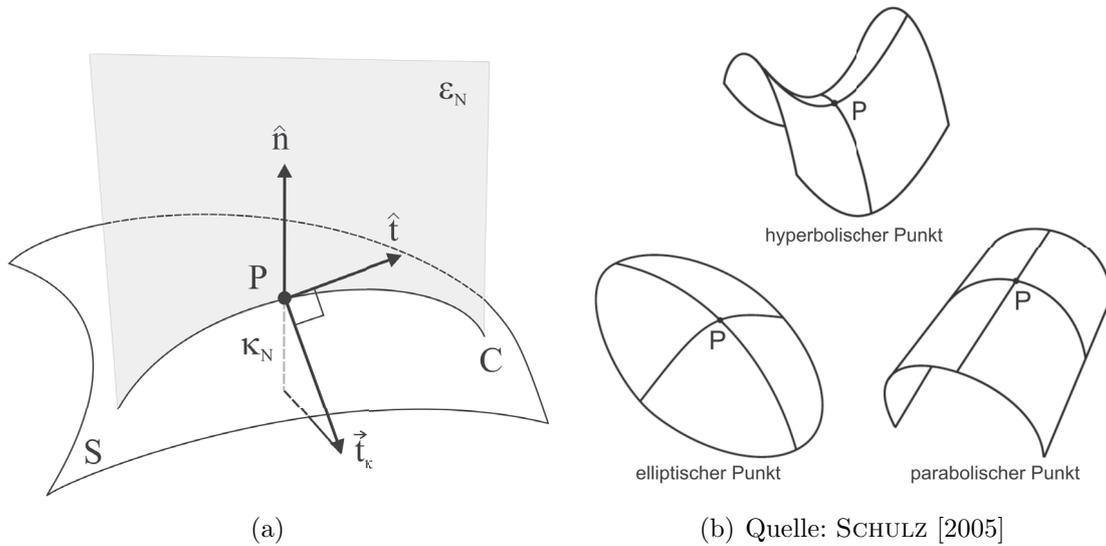
$$p : x(u, v) = \begin{pmatrix} x_1(u, v) \\ x_2(u, v) \\ x_3(u, v) \end{pmatrix} \quad (3.3)$$

Das  $(u, v)$  Koordinatenpaar wird als Gauß'sche Koordinate bezeichnet, das den Schnittpunkt einer  $u$ - und  $v$ -Flächenkurve auf  $S$  an  $p$  darstellt. Die Elemente der Weingartenmatrix beschreiben eine  $2 \times 2$ -Matrix und bestehen aus den Größen der ersten (I) und zweiten (II) Fundamentalform der Flächentheorie mit:

$$\begin{aligned} I : E &= x_u \cdot x_u & F &= x_u \cdot x_v & G &= x_v \cdot x_v \\ II : L &= x_{uu} \cdot \hat{n} & M &= x_{uv} \cdot \hat{n} & N &= x_{vv} \cdot \hat{n} \\ W &= \frac{1}{EG - F^2} \begin{bmatrix} MF - LG & LF - ME \\ NF - MG & MF - NE \end{bmatrix} \end{aligned} \quad (3.4)$$

Handelt es sich bei  $x_u$  und  $x_v$  um orthonormale Tangentialvektoren, die die Basis der Tangentialebene  $T(p)$  von  $p$  bilden, so entspricht  $W$  einer symmetrischen Matrix mit

$$W = \frac{1}{EG - F^2} \begin{bmatrix} L & M \\ M & N \end{bmatrix}. \quad (3.5)$$



**Abb. 3.2:** (a) Geometrische Interpretation der Normalenkrümmung  $\kappa_n$  an einen Flächenpunkt  $p$  auf einer Flächenkurve  $C$ . Die Kurve  $C$  entsteht durch den Normalschnitt von  $S$  und der von dem Tangentenvektor  $\hat{t}$  an  $p$  abhängigen Normalenschnittebene  $\epsilon(\hat{n})$ . (b) Beispiele der drei Klassifikationsgruppen eines Oberflächenpunktes  $p$  mit den entsprechenden Hauptkrümmungslinien.

Mit Hilfe eines weiteren Tangentialvektors  $\hat{l} = ax_u + bx_v$  innerhalb von  $T(p)$ , kann die Normalenkrümmung bezüglich der gegebenen Basis durch

$$\kappa_N = [a \ b] W \begin{bmatrix} a \\ b \end{bmatrix} \quad (3.6)$$

berechnet werden. Voraussetzung für die Bestimmung von I und II ist dabei eine  $C^2$  stetige Fläche, da die Fundamentalgrößen Ableitungen 1. und 2. Ordnung enthalten.

Die dabei entstehende quadratische Form von  $\kappa_N$  ergibt einen maximalen und minimalen Krümmungswert (als Extremwerte), welche den Eigenwerten von  $W$  entsprechen. Werden nun die Normalenkrümmungen aller Flächenkurven  $C_i$  betrachtet, die durch  $p$  verlaufen, so ergeben sich eine maximale und minimale Hauptkrümmung, die als  $\kappa_1$  und  $\kappa_2$  bezeichnet werden. Die zugehörigen Eigenvektoren von  $W$  entsprechen den Hauptkrümmungsrichtungen  $\hat{t}_{\kappa_1}$  und  $\hat{t}_{\kappa_2}$ , welche orthogonal zueinander stehen. Die Hauptkrümmungen geben somit die stärkste und geringste Krümmung (Biegung) von  $S$  an  $p$  an. Die Hauptkrümmungsrichtungen zeigen entsprechend in die Richtung der stärksten und geringsten Krümmung. Die Kurven, die entlang der Hauptkrümmungsrichtungen verlaufen, werden als Hauptkrümmungslinien bezeichnet.

Ebene und kugelige Oberflächen stellen hier jedoch eine Ausnahme dar, da praktisch jeder Oberflächenpunkt jeweils die gleiche Krümmung besitzt und damit jede Richtung als Hauptkrümmungsrichtung interpretiert werden kann.

Aus den beiden Hauptkrümmungen lassen sich die mittlere und Gauß'sche Krümmung durch

$$H = \frac{(\kappa_1 + \kappa_2)}{2} \quad (3.7)$$

sowie

$$K = \kappa_1 \kappa_2 \quad (3.8)$$

bestimmen. Mit Hilfe der Krümmungsgröße  $K$  ist nun eine Klassifizierung möglich, die einen Oberflächenpunkt in folgende drei Gruppen einteilt:

- *elliptisch*:  $K \approx 0$ ,  $\kappa_1 \wedge \kappa_2 \approx 0$  (planar) o.  $K \neq 0$ :  $\kappa_1 \approx \kappa_2 \wedge \kappa_1 \approx \kappa_2 \neq 0$  (sphärisch)
- *parabolisch*:  $K \approx 0$ ,  $\max(|\kappa_1|, |\kappa_2|) > 0 \wedge \min(|\kappa_1|, |\kappa_2|) \approx 0$  und
- *hyperbolisch*:  $K < 0$ ,  $\max(\kappa_1, \kappa_2) > 0 \wedge \min(\kappa_1, \kappa_2) < 0$ .

Wie zu erkennen ist, zählen zu elliptischen Oberflächenpunkten auch planare und sphärische Flächen. Des Weiteren werden bei elliptischen und parabolischen Umgebungen zwischen konkaven und konvexen Oberflächenorientierungen unterschieden. Beispiele zu allen drei Gruppen sind in Abb. 3.2 (b) dargestellt, wobei zusätzlich die beiden Hauptkrümmungslinien an  $p$  jeweils eingezeichnet sind.

## 3.2 Verfahren zur Krümmungsapproximation

Die Krümmungsbestimmung auf Dreiecksgeometrien erfordert eine Übertragung der oben beschriebenen Methoden von kontinuierliche auf diskrete Oberflächen. Aufgrund des diskreten Charakters der zugrunde liegenden Gitterstrukturen können die Krümmungsinformationen allerdings nur eine Approximation des tatsächlichen Oberflächenverlaufes darstellen. Für deren Bestimmung existieren zwei Klassen von Verfahrensansätzen, das *Curve Fitting* und das *Surface Fitting*. Dazu werden im Folgenden einige Techniken vorgestellt, die in der Diplomarbeit von SCHULZ [2005] implementiert worden und für diese Arbeit verwendet werden. Zur Bestimmung der Krümmungsinformationen für einen Dreieckseckpunkt  $v_i$  ist in beiden Ansätzen die Definition einer Umgebung  $nhd(v_i)$  notwendig. In dieser sind neben  $v_i$  selbst, alle Nachbarnpunkte  $v_j$  enthalten, die in einem Radius  $r$  um  $v_i$  liegen. Der Radius bezieht sich dabei auf den topologischen Abstand. Zusätzlich muss als Bedingung für die Verfahren eine 2-Mannigfaltigkeit der Oberfläche gegeben sein. Dies bedeutet, dass eine ausgehende Kante an  $v_i$  stets 2 angrenzende Flächen besitzen muss. Ist dies an  $v_i$  nicht erfüllt, so wird dieser als degenerierter Punkt bezeichnet und es können keine Krümmungsinformationen an ihm bestimmt werden.

Zusätzlich wurde in SCHULZ [2005] das erzeugte Feld aus den Hauptkrümmungsrichtungen  $\hat{t}\kappa_1$  und  $\hat{t}\kappa_2$  durch ein Optimierungsverfahren nach HERTZMANN UND ZORIN [2000] geglättet. Dazu wurden die Hauptkrümmungsrichtungen an den Eckpunkten als „fest“

markiert, die eine stark parabolische Umgebung beschreiben. Parabolische Punkte sind charakterisiert durch eine hohe Krümmung in eine der beiden Hauptkrümmungsrichtungen bei gleichzeitig sehr geringer Krümmung der anderen Hauptkrümmungsrichtung (vgl. Abschnitt 3.1.2). Daher müssen als Kriterium, ob ein Punkt parabolisch ist, folgende zwei Bedingungen erfüllt sein:

$$\frac{\max(|\kappa_1|, |\kappa_2|)}{\min(|\kappa_1|, |\kappa_2|)} > \delta_{ratio} \wedge \max(|\kappa_1|, |\kappa_2|) > \delta_{reliable} \quad (3.9)$$

Die Schwellwerte  $\delta_{ratio}$  und  $\delta_{reliable}$  werden dabei vom Benutzer eingestellt. Die Richtungen an den übrigen Eckpunkten wurden anschließend gegen die „fest“-markierten ausgerichtet.

### 3.2.1 Curve Fitting

Beim Ansatz des *Curve Fitting* werden die Normalenkrümmungen  $\kappa_{Nij}$  an den inzidenten Dreieckskanten  $e_{ij}$  eines Punktes  $v_j \in nhd(v_i)$  untersucht und daraus die Hauptkrümmungswerte ermittelt. Die Abschätzung von  $\kappa_{Nij}$  geschieht dabei jeweils durch:

$$\kappa_{Nij} = 2 \frac{\hat{n}(v_i) \cdot (v_j - v_i)}{\|v_j - v_i\|} \quad (3.10)$$

In SCHULZ [2005] wurden drei Verfahren für das *Curve Fitting* umgesetzt, welche im Folgenden kurz vorgestellt werden.

#### NormalOP

Bei diesem Operator von GOLDFEATHER [2003] werden die Krümmungsinformationen anhand eines *Least Square* Verfahren zur Lösung von Gleichung 3.6 approximiert bestimmt. Die Normalenkrümmung  $\kappa_{\hat{n}ij}$  wird durch 3.10 approximiert und der Tangentialvektor  $\hat{t}_{ij}$  durch die Projektion von  $e_{ij}$  nach  $T(p_i)$  erzeugt. Nach Umstellung von 3.6 ergibt sich folgendes Gleichungssystem:

$$\begin{bmatrix} \dots & \dots & \dots \\ a_{ij}^2 & 2a_{ij}b_{ij} & b_{ij}^2 \\ \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dots \\ \kappa_{Nij} \\ \dots \end{bmatrix}. \quad (3.11)$$

Jede Zeile der linken Matrix repräsentiert eine Kante von  $v_i$  zu  $v_j \in nhd(v_i)$ . Die Skalare  $a_{ij}$  und  $b_{ij}$  beschreiben dabei den Tangentenvektor der Kante  $e_{ij}$  in einer beliebigen orthonormalen Basis innerhalb der Tangentialebene. Die Krümmungsinformationen entsprechen dann den oben erwähnten Eigenwerten und Eigenvektoren der approximierten Elemente der Weingartenmatrix.

### TaubinOP

In TAUBIN [1995] werden die Krümmungsinformationen anhand des Zusammenhanges zwischen der orthonormalen Basis  $(\hat{t}_{\kappa_1}, \hat{t}_{\kappa_2}, \hat{n}(v_i))$  und einer beliebigen anderen orthonormalen Basis  $(\hat{x}_1, \hat{x}_2, \hat{x}_3)$  bestimmt. Dazu wird ebenfalls eine Matrix  $M_{P_i}$  aufgestellt, in der Form:

$$M_{P_i} = \sum_{nhd(v_i)} w_{ij} \kappa_{N_{ij}} \hat{t}_{ij} \hat{t}_{ij}^T, \quad (3.12)$$

wobei  $nhd(v_i)$  nur die direkten Nachbarknoten von  $v_i$  enthält. Die Normalenkrümmung wird anhand von 3.10 abgeschätzt und der Tangentialvektor  $\hat{t}_{ij}$  entspricht wieder der Projektion von  $e_{ij}$  in die Tangentialebene. Zusätzlich fließt noch ein Wichtungsfaktor  $w_{ij}$  ein, der abhängig ist vom Flächeninhalt der angrenzenden Dreiecke an  $e_{ij}$ . Die Eigenvektoren von  $M_{P_i}$  entsprechen den Hauptkrümmungsrichtungen und mit Hilfe der Eigenwerte  $\lambda_1, \lambda_2$  können die Hauptkrümmungswerte durch

$$\begin{aligned} \kappa_1 &= 3\lambda_1 - \lambda_2 \\ \kappa_2 &= 3\lambda_2 - \lambda_1 \end{aligned} \quad (3.13)$$

bestimmt werden.

### MaubeugeOP

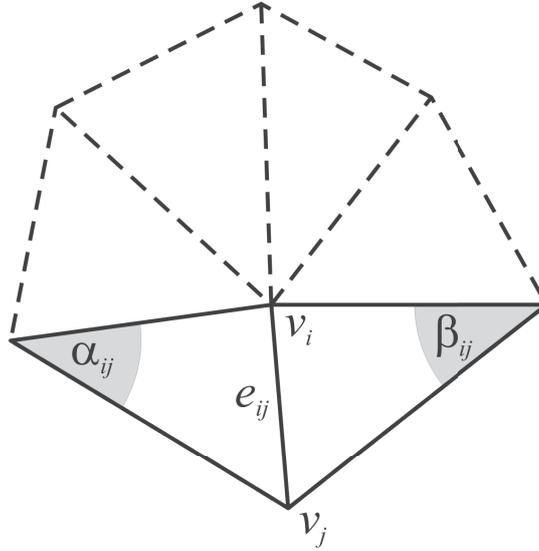
Für den *MaubeugeOP*-Operator von MEYER U. A. [2003] werden zunächst die Gauß'sche und mittlere Krümmung berechnet, aus denen sich dann  $\kappa_1$  und  $\kappa_2$  bestimmen lassen. Zur Approximation von  $H$  wird der Zusammenhang von lokaler Flächenänderung in einem Umgebungsbereich  $U_{P_i}(r)$  eines Punktes  $v_i$  bezüglich dessen mittleren Krümmungsvektors  $\bar{t}_{\kappa i}$  ausgenutzt. Der Änderungsgradient  $\nabla A$  wird dazu wie folgt abgeschätzt:

$$\nabla A = \frac{1}{2} \sum_{nhd(v_i)} (\cot \alpha_{ij} + \cot \beta_{ij})(v_i - v_j). \quad (3.14)$$

Die Winkel  $\alpha_{ij}$  und  $\beta_{ij}$  entsprechen dabei den gegenüberliegenden Winkeln der zwei Dreiecke, die an der Kante  $e_{ij}$  benachbart sind (vgl. Abb. 3.3). Eine Approximation des mittleren Krümmungsvektors erfolgt dann mittels:

$$\bar{t}_{\kappa i} = \frac{2\nabla A}{A(U_{P_i}(r))} \quad (3.15)$$

Die Berechnung des Flächeninhaltes  $A(U_{P_i}(r))$  geschieht dabei nur über die Flächenbereiche, die innerhalb einer stückweise linearen Begrenzungslinie um  $v_i$  liegen. Diese verläuft in der Mitte jeder inzidenten Kante  $e_{ij}$  und innerhalb der inzidenten Dreiecke entweder durch den Dreiecksschwerpunkt (baryzentrisch, im Fall von stumpfen Dreiecken) oder durch den Umkreismittelpunkt (Voronoi-Zerlegung, im Fall von spitzen Dreiecken).



**Abb. 3.3:** Darstellung der gegenüberliegenden Winkel  $\alpha_{ij}$  und  $\beta_{ij}$  an einer Kante  $e_{ij}$ . Mit diesen erfolgt die Bestimmung des Änderungsgradienten  $\nabla A$  des Flächeninhaltes innerhalb der direkten Nachbarschaft eines Eckpunktes  $v_i$ .

Die Berechnung der Gauß'schen Krümmung  $K$  erfolgt anhand der Gaußabbildung. Diese weist jedem Flächenpunkt  $v_i$  eines Flächenbereiches  $B$  einen Punkt auf der Einheitskugel zu, entlang seiner Normalen  $\hat{n}(v_i)$ . Für  $K$  wird nun das Verhältnis zwischen dem sphärischen Flächeninhalt  $A_G(B)$  auf der Kugel zum Flächeninhalt  $A(B)$  auf der Oberflächen untersucht. Eine Approximation erfolgt durch:

$$K = \frac{A_G(B)}{A(B)} = \frac{2\pi - \sum_{nhd(v_i)} \phi_j}{A(B)}, \quad (3.16)$$

wobei  $\phi_j$  der Winkel des inzidenten Dreieckes an  $v_i$  ist. Mit Hilfe von  $H$  und  $K$  sind die Hauptkrümmungen:

$$\begin{aligned} \kappa_1 &= H + \sqrt{H^2 - K} \\ \kappa_2 &= H - \sqrt{H^2 - K} \end{aligned} \quad (3.17)$$

bestimmbar und die Hauptkrümmungsrichtungen werden anhand von 3.11 ermittelt. Für das *Least Square* Problem wird  $H$  als lineare Bedingung verwendet.

### 3.2.2 Surface Fitting

Beim Ansatz des *Surface Fitting* wird eine geeignete Passfläche für den Oberflächenverlauf an einem Punkt  $v_i$  gesucht, deren Beschreibung zur Bestimmung der Krümmungsinformationen genutzt wird. Das Vorgehen erfolgt dabei in drei Schritten (vgl. GOLDFEATHER UND INTERRANTE [2004], SCHULZ [2005]):

### 1. Parametrisierung der Punktumgebung

Diese kann topologieunabhängig oder topologieabhängig erfolgen. Ersteres bedeutet keine Beachtung von Kanten und Flächenbeziehungen zwischen Punkten in  $nhd(v_i)$ , wozu beispielsweise die Projektion von  $nhd(v_i)$  in die Tangentialebene  $T(v_i)$  gehört. Die Projektion geschieht dabei entlang der Normalen  $\hat{n}(v_i)$ , welche bei den *Curve Fitting*-Verfahren ebenfalls angewendet wird. Bei einer topologieabhängigen Parametrisierung wird  $nhd(v_i)$  auf einen isomorphen und kreuzungsfreien Graphen im  $R^2$  abgebildet (vgl. ECK U. A. [1995]).

### 2. Abbildung der Parametrisierung auf eine Passfläche

Die  $(u, v)$ -Parameter aus 1. werden über eine Abbildung  $F : R^2 \rightarrow R^3$  zur Erzeugung einer kontinuierlichen Fläche verwendet, welche die diskrete Umgebungsgeometrie an  $v_i$  am genauesten beschreibt. Das Finden der Abbildung  $F$  wird dabei über ein Optimierungsproblem gelöst.

### 3. Bestimmung der Krümmungsinformationen

Die aus der Abbildung  $F$  gewonnenen Parameter  $F(u_i, v_i)$  werden nun zur Bestimmung der Gleichungen in 3.5 verwendet, woraus sich wiederum die Krümmungsgrößen ableiten lassen.

Im Folgendem werden vier Krümmungsoperatoren vorgestellt, die auf dem *Surface Fitting* Ansatz basieren und sich hinsichtlich Schritt 1 und 2 unterscheiden.

## QuadraticOP

Bei dieser Methode von GOLDFEATHER [2003] wird zunächst eine topologieunabhängige Parametrisierung von  $nhd(v_i)$  erzeugt. Durch diese wird im Anschluss versucht, eine quadratische Fläche (Funktion) bestmöglich an die Umgebung von  $v_i$  anzulegen. Die quadratische Approximation jedes  $v_j \in nhd(v_i)$  hat die Form:

$$z_{ij} = F(x_{ij}, y_{ij}) = \frac{1}{2}Ax_{ij}^2 + Bx_{ij}y_{ij} + \frac{1}{2}Cy_{ij}^2, \quad (3.18)$$

wobei  $(x_{ij}, y_{ij})$  das Parameterpaar  $(u_{ij}, v_{ij})$  von  $v_j$  ist. Für jeden Punkt von  $nhd(v_i)$  ergibt sich unter Verwendung von 3.18 folgendes Gleichungssystem:

$$\begin{bmatrix} \dots & \dots & \dots \\ \frac{1}{2}x_{ij}^2 & x_{ij}y_{ij} & \frac{1}{2}y_{ij}^2 \\ \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \dots \\ z_{ij} \\ \dots \end{bmatrix}, \quad (3.19)$$

das sich wie Gleichung 3.11 durch ein *Least Square* Verfahren lösen lässt. Der Unterschied zum *NormalOP*-Operator besteht darin, dass bei diesem die Krümmung anhand von Krümmungskreisen approximiert wird und beim *QuadraticOP*-Operator mit Hilfe von parabolischen Flächen.

### CubicOP

In GOLDFEATHER [2003] wird eine Erweiterung des *QuadraticOP* Operators vorgeschlagen, indem zusätzlich die Normaleninformationen der inzidenten Punkte verwendet werden. Damit wird eine kubische Fläche an  $v_i$  erzeugt, mit folgender Form:

$$z_{ij} = F(x_{ij}, y_{ij}) = \frac{1}{2}Ax_{ij}^2 + Bx_{ij}y_{ij} + \frac{1}{2}Cy_{ij}^2 + Dx_{ij}^3 + Ex_{ij}^2y_{ij} + Fx_{ij}y_{ij}^2 + Gy_{ij}^3 \quad (3.20)$$

Die Normale an jedem Punkt  $v_j \in nhd(v_i)$  ist durch die erste Ableitung von 3.20 mit  $N(x_{ij}, y_{ij}) = (f_x(x_{ij}, y_{ij}), f_y(x_{ij}, y_{ij}), -1)$  gegeben, was

$$(Ax_{ij} + By_{ij} + 3Dx_{ij}^2 + 2Ex_{ij}y_{ij} + Fy_{ij}^2, Bx_{ij} + Cy_{ij} + Ex_{ij}^2 + 2Fx_{ij}y_{ij} + 3Gy_{ij}^2, -1) \quad (3.21)$$

entspricht. Daraus lässt sich ein Gleichungssystem aufstellen, das für jeden Punkt  $v_j \in nhd(v_i)$  und seiner Normalen drei Gleichungen besitzt. Dieses lässt sich ebenfalls durch ein *Least Square* Verfahren lösen:

$$\begin{bmatrix} \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{1}{2}x_{ij}^2 & x_{ij}y_{ij} & \frac{1}{2}y_{ij}^2 & x_{ij}^3 & x_{ij}^2y_{ij} & x_{ij}y_{ij}^2 & y_{ij}^3 \\ x_{ij} & y_{ij} & 0 & 3x_{ij}^2 & 2x_{ij}y_{ij} & y_{ij}^2 & 0 \\ 0 & x_{ij} & y_{ij} & 0 & x_{ij}^2 & 2x_{ij}y_{ij} & 3y_{ij}^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} X = \begin{bmatrix} \dots \\ z_{ij} \\ -\frac{a_{ij}}{c_{ij}} \\ -\frac{b_{ij}}{c_{ij}} \\ \dots \end{bmatrix} \quad (3.22)$$

Die Normale sei dabei gegeben durch  $(a_{ij}, b_{ij}, c_{ij})$  und  $(A, B, C, D, E, F, G)^T$  als  $X$ .

### ParametricOP

Bei diesem Operator von RÖSSL U. A. [2000] erfolgt die Parametrisierung im Gegensatz zu den bisherigen Verfahren durch die Approximation geodätischer Polarkoordinaten von  $v_j \in nhd(v_i)$ . Die Parametrisierung bezieht sich dabei nur auf die direkte Nachbarschaft. Dazu werden alle Winkel  $\phi_{ij}$  zwischen  $e_{ij}$  und  $e_{ij+1}$  (beide in  $nhd(v_i)$ ) aufsummiert und anschließend einheitlich skaliert, so dass die Summe  $2\pi$  ergibt. Die sich daraus ergebenden neuen skalierten Winkel  $\phi'_{ij}$  in Kombination mit den euklidischen Abständen  $r_{ij}$  zwischen  $v_j$  und  $v_i$  sind die Parameterkoordinaten  $(\phi'_{ij}, r_{ij})$ , die im Folgenden als  $(u, v)$  bezeichnet werden. Mit Hilfe eines quadratischen Taylorpolynoms der gesuchten Fläche:

$$F(u, v) = uF_u + vF_v + \frac{1}{2}u^2F_{uu} + uvF_{uv} + \frac{1}{2}v^2F_{vv}. \quad (3.23)$$

und der Normalen  $\hat{n}$  mit  $\hat{n} = (F_u \times F_v) / \|F_u \times F_v\|$ , wird die Weingartenmatrix wie folgt gebildet:

$$\begin{bmatrix} F_{uu}^T \hat{n} & F_{uv}^T \hat{n} \\ F_{uv}^T \hat{n} & F_{vv}^T \hat{n} \end{bmatrix}. \quad (3.24)$$

Damit wird wiederum ein Gleichungssystem aufgestellt, in welchem die gesuchten Fundamentalgrößen  $F = F_{uu}, F_{uv}, F_{vv}$  durch ein *Least Square* Verfahren ermittelt werden:

$$\begin{bmatrix} u_j & v_j & \frac{1}{2}u_j^2 & u_j v_j & \frac{1}{2}v_j^2 \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} F = \begin{bmatrix} \dots \\ x_j \\ \dots \end{bmatrix} \begin{bmatrix} \dots \\ y_j \\ \dots \end{bmatrix} \begin{bmatrix} \dots \\ z_j \\ \dots \end{bmatrix} \quad (3.25)$$

Das Koordinatenpaar  $(x_j, y_j, z_j)$  bezeichnet die Position von  $v_j \in nhd(v_i)$  in 3D.

### BiQuadBezOP

Dieser Operator verwendet eine biquadratische *Bèzier* Oberfläche, um  $nhd(v_i)$  zu beschreiben (vgl. RAZDAN UND BAE [2005]). Eine derartige Passfläche hat folgende formale Beschreibung:

$$F(u, v) = \sum_{k=0}^2 \sum_{l=0}^2 B_k^2(u) B_l^2(v) v_{k,l}. \quad (3.26)$$

Hierbei stellen  $v_{k,l}$  die Kontrollpunkte der Fläche dar und die Koeffizienten  $B_k^2(u), B_l^2(v)$  die quadratischen Bernstein Polynome. Die Parametrisierung erfolgt topologieabhängig (harmonisch) durch die Projektion der Randpunkte von  $nhd(v_i)$  auf einen Einheitskreis. Die Verteilung der restlichen Punkte innerhalb des Kreises wird mit Hilfe eines *Least Square* Verfahrens durchgeführt (vgl. ECK U. A. [1995]). Ziel ist es, zunächst die Kontrollpunkte  $v_{k,l}$  der Passfläche zu finden, die die Punktumgebung von  $v_i$  am besten approximiert. Dazu wird für alle  $n$  Punkte  $v_j \in nhd(v_i)$  folgendes Gleichungssystem aufgestellt, wo die gesuchten Kontrollpunkte ebenfalls über ein *Least Square* Verfahren gefunden werden:

$$\begin{bmatrix} B_0^2(u_0)B_0^2(v_0) & B_0^2(u_0)B_1^2(v_0) & \dots & B_2^2(u_0)B_2^2(v_0) \\ & & \vdots & \\ B_0^2(u_n)B_0^2(v_n) & B_0^2(u_n)B_1^2(v_n) & \dots & B_2^2(u_n)B_2^2(v_n) \end{bmatrix} \begin{bmatrix} v_{0,0} \\ \vdots \\ v_{2,2} \end{bmatrix} = \begin{bmatrix} v_0 \\ \vdots \\ v_n \end{bmatrix}. \quad (3.27)$$

Die ermittelten Kontrollpunkte  $v_{k,l}$  und Parameterkoordinaten  $(u_i, v_i)$  für den Punkt  $v_i$  werden für die Ableitungsberechnung der Fundamentalgrößen (vgl. Gleichung 3.4) verwendet. Dabei wird folgender Zusammenhang ausgenutzt (vgl. PIEGL UND TILLER [1995]):

$$\frac{d\left(\sum_{k,n} (u_i) v_k\right)}{du_i} = n \sum_{k=0}^{n-1} (u_i) (v_{k+1} - v_k). \quad (3.28)$$

Anders als bei den bisherigen Verfahren lässt sich keine Weingartenmatrix aufstellen, um die Krümmungsinformationen zu bestimmen. Laut SCHULZ [2005] ist es nicht garantiert,

dass eine orthonormale Basis für die Tangentialebene existiert. Jedoch zeigt SCHÖNE [1975], dass aus den abgeleiteten Fundamentalgrößen zwei quadratische Gleichungen aufgestellt werden können, deren zwei Lösungen jeweils  $\kappa_1, \kappa_2$  sowie  $\lambda_1, \lambda_2$  ergeben:

$$\begin{aligned} (EG - F^2)\kappa^2 - (EN + GL - 2FM)\kappa + (LN - M^2) &= 0 \\ (FN - GM)\lambda^2 + (EN - GL)\kappa + (EM - FL) &= 0. \end{aligned} \quad (3.29)$$

Die beiden Werte  $\lambda_1, \lambda_2$  stellen das Verhältnis der Koordinaten von  $\hat{t}_{\kappa_1}, \hat{t}_{\kappa_2}$  dar und beschreiben damit die Hauptkrümmungsrichtungen.

### 3.3 Zusammenfassung und Schlussfolgerungen

Krümmungsinformationen beschreiben den ebenen Verlauf einer Kurve oder Fläche an einen Punkt  $v_i$  sowie dessen Oberflächenorientierung. Von Interesse für diese Arbeit ist dabei die Krümmungsapproximation auf triangulierten 3D-Oberflächen, da die ermittelten Informationen für die Generierung eines Vektorfeldes verwendet werden können. An diesem orientieren sich die späteren Schraffurlinien.

Wichtige Krümmungsgrößen sind zum einen die beiden Hauptkrümmungen  $\kappa_1, \kappa_2$ , die jeweils die maximale und minimale Krümmung an  $v_i$  angeben sowie die entsprechenden Hauptkrümmungsrichtungen  $\hat{t}_{\kappa_1}, \hat{t}_{\kappa_2}$ . Zum anderen lässt sich ein Oberflächenpunkt über die Gauß'sche Krümmung  $K$  zu einem der drei Umgebungsbeschreibungen elliptisch, parabolisch oder hyperbolisch zuordnen. Zur Berechnung der Krümmungsgrößen wird die Normalenkrümmung an  $v_i$  mit Hilfe der Weingartenmatrix  $W$  und gegebener Tangentenrichtung  $\hat{t}$  berechnet. Die Eigenwerte von  $W$  entsprechen dabei  $\kappa_1, \kappa_2$  und die Eigenvektoren  $\hat{t}_{\kappa_1}, \hat{t}_{\kappa_2}$ . Zur Berechnung von  $W$  werden die Größen der  $I$  und  $II$  Fundamentalform gesucht, welche durch ein *Least Square* Verfahren ermittelt werden. Dies setzt eine  $C^2$  stetige Oberfläche voraus, da für die Fundamentalgrößen Ableitungen der 1. und 2. Ordnung notwendig sind.

Verfahren, die die Krümmung auf diskreten Oberflächen bezüglich einer definierten Umgebung  $nhd(v_i)$  approximieren, unterscheiden sich hinsichtlich einem *Curve Fitting* oder *Surface Fitting* Ansatz. Die erste Variante ermittelt die Normalenkrümmung anhand der inzidenten Kanten durch Abschätzung von entsprechenden Krümmungskreisen. Für die zweite Variante wird zunächst eine Parametrisierung von  $nhd(v_i)$  erzeugt und diese durch ein *Least Square* Verfahren auf eine geeignete Passfläche abgebildet, die die definierte Umgebung von  $v_i$  approximiert. Auf dieser werden dann die Krümmungsinformationen ermittelt. Die Parametrisierung kann dabei topologieabhängig oder topologieunabhängig erfolgen. Die vorgestellten und von SCHULZ [2005] umgesetzten Verfahren sind in Tabelle 3.1 nocheinmal kurz zusammengefasst.

Verfahren	Ansatz	Parametrisierung	$nhd(v_i)_{min}$
NORMALOP	<i>Curve Fitting</i>	Projektion	3
TAUBINOP	<i>Curve Fitting</i>	Projektion	3
MAUBEUGEOP	<i>Curve Fitting</i>	keine	3
QUADRATICOP	<i>Surface Fitting</i> (quadratische Passfläche)	Projektion	3
CUBICOP	<i>Surface Fitting</i> (kubische Passfläche)	Projektion	3
PARAMETRICOP	<i>Surface Fitting</i> (quadratische Passfläche)	geodätische Polarkoordinaten	5
BIQUADBEZOP	<i>Surface Fitting</i> (biquadratische <i>Bèzier</i> -Fläche)	isomorph	9

**Tabelle 3.1:** Übersicht verschiedener Krümmungsapproximationsverfahren, die in SCHULZ [2005] umgesetzt und untersucht wurden. Sie unterscheiden sich in *Curve Fitting* und *Surface Fitting* Ansätzen, der Art ihrer Parametrisierung und Mindestanzahl von Eckpunkten in ihrer Nachbarschaft.

## Schlussfolgerungen

Die vorgestellten Verfahren zur Krümmungsapproximation unterscheiden sich hinsichtlich der Anforderungen an die Punktumgebung eines Eckpunktes, der damit verbundenen Robustheit in der Krümmungsapproximation und der Komplexität sowie Qualität ihrer Berechnungen. Folgende drei Aspekte sind davon bei der Bestimmung von Krümmungsinformationen für diese Arbeit wichtig:

- **Robustheit**

Die Robustheit bezieht sich auf die Bestimmbarkeit von Krümmungsinformationen für einen Eckpunkt  $v_i$ . Zum einen muss dabei die Situation berücksichtigt werden, wenn für ein Verfahren die Mindestanzahl von Nachbarpunkten in  $nhd(v_i)$  aufgrund eines zu geringen Radius nicht gegeben ist. Zum anderen beinhaltet die Robustheit einen angemessenen Umgang mit degenerierten Bereichen (vgl. Abschnitt 3.2) an  $v_i$ .

- **Genauigkeit**

Die approximierten Krümmungsinformationen an einem Punkt  $v_i$  sollten diesen angemessen genau als hyperbolischen, zylindrischen oder elliptischen Bereich beschreiben.

- **Berechnungszeit**

Für die Bestimmung der Krümmungsinformationen benötigen die Verfahren einen unterschiedlichen Berechnungsaufwand. Je komplexer das Aufstellen und Lösen von Gleichungssystemen sowie die Art der Parametrisierung ist, desto höher ist

die Berechnungszeit. Für diese Arbeit sind schnellere und einfachere Verfahren vorzuziehen, wenn deren Krümmungsinformationen gegenüber aufwändigeren aber genaueren Verfahren nur minimal abweichen.

In SCHULZ [2005] erfolgten verschiedene Untersuchungen bezüglich der Eignung der umgesetzten Approximationsverfahren auf anatomischen Oberflächenmodellen. Die verwendeten Strukturen sind vergleichbar mit denen, die in dieser Arbeit zur Verfügung stehen. Die Untersuchungen betrachteten einfache wie auch komplexe Oberflächen, wobei die Komplexität sich auf die Anzahl von Oberflächenpunkten mit hyperbolischer, elliptischer und zylindrischer Umgebung bezieht. Die Ergebnisse zeigten, dass bei komplexen Oberflächen, wie einem Schädel oder der *Carotis* (Halsschlagader), topologieabhängige Approximationsverfahren besser geeignet sind als nicht topologieabhängige. Die ermittelten Krümmungsinformationen wiesen eine geringere Abweichung bezüglich den zu erwartenden Krümmungswerten auf. Für einfachere Strukturen wie Muskeln, bei denen sich einzelne Oberflächenorientierungen über einen großen Bereich erstrecken, liefern bereits die nicht topologieabhängigen Verfahren geeignete Werte.

In SCHULZ [2005] werden für komplexe Objekte die *ParametricOP* oder *BezierQuadOP* Operatoren vorgeschlagen, wenn die Anzahl der direkten Nachbarn eines Eckpunktes fünf oder mehr ist. Damit ist eine korrekte topologische Parametrisierung garantiert und eine angemessen genaue Beschreibung des Krümmungsverhaltens gegeben. Dabei steigt die Genauigkeit mit erweiterter Nachbarschaft, d. h.  $r > 1$ , wodurch jedoch auch die Berechnungszeit stark zunimmt, insbesondere beim *BezierQuadOP*-Operator. Für einfache Strukturen liefern bereits die Operatoren TAUBINOP und QUADRATICOP mit ihrer projektiven Parametrisierung geeignete Krümmungsinformationen. Voraussetzung dafür ist allerdings eine korrekte, topologische Parametrisierung von  $nhd(v_i)$ . Ist dies nicht gegeben, so weicht das ermittelte Krümmungsverhalten zu sehr von dem tatsächlichen ab. Obwohl der TAUBINOP-Operator das schnellste Verfahren darstellt, unterstützt es als einziges keine erweiterte Nachbarschaft. Nachteilig wirkt dies sich auf die Genauigkeit aus, da nur die direkten Nachbarn bei der Krümmungsapproximation einfließen.

Es zeigt sich, dass für die spätere Auswahl, welches Verfahren für die anatomischen Strukturen am besten geeignet ist, kein eindeutiges festgelegt werden kann. Die oben betrachteten Aspekte verlangen nach einem Kompromiss zwischen Robustheit, Genauigkeit und Geschwindigkeit in der Berechnung. Einfluss hat dabei die Komplexität der Oberflächenmodelle, welche je nach anatomischer Struktur unterschiedlich stark ausfällt.

## 4 Entwurf für Schraffierung medizinischer Datensätze

Ziel und Inhalt dieses Kapitels ist die konzeptionelle Erarbeitung und Präsentation eines texturbasierten Verfahrens zur modell- und krümmungsbasierten Schraffierung von medizinischen Datensätzen. Dazu wird zunächst eine Vorbetrachtung bezüglich der zu verwendenden Datensätze und der aus dem vorhergehenden Kapitel erarbeiteten Grundlagen durchgeführt. Danach erfolgt eine Analyse der sich ergebenden Anforderungen für das Verfahren. In dem sich anschließenden Entwurf werden einzelne Teilschritte aufgezeigt und entwickelt, die für die texturbasierte Schraffierung notwendig sind. Der Schwerpunkt liegt dabei weniger auf einer genauen Nachahmung von Oberflächenschraffuren, als vielmehr auf deren echtzeitfähigen und automatischen Realisierung. Für die notwendigen Krümmungsapproximationen der zu schraffierenden Oberflächen werden im Weiteren die Verfahren aus Abschnitt 3.2 verwendet.

### 4.1 Vorbetrachtungen

Im Abschnitt 2.2.3 sind wichtige Aspekte der Schraffierung von anatomischen Strukturen aufgezeigt worden. Da es sich in der vorliegenden Arbeit speziell um die Schraffierung patientenspezifischer, anatomischer Strukturen innerhalb einer interaktiven Visualisierung handelt, sind dafür weitere Betrachtungen notwendig. Diese beziehen sich einerseits auf die zugrunde liegenden Datensätze und andererseits auf die korrekte Ausrichtung der Schraffur. Wichtig in diesem Zusammenhang ist die Beantwortung der Frage, was „korrekt“ im Sinne des Betrachters ist. Dieser Abschnitt zeigt Probleme und Fragen aber auch Schlussfolgerungen auf, die sich aufgrund der Datensätze und der strukturentypischen Schraffierung ergeben.

#### 4.1.1 Analyse der patientenspezifischen Datensätze

Die patientenspezifischen Datensätze liegen zunächst als 3D-Volumendatensatz vor, der sich aus mehreren 2D-Schichten von radiologischen MRT- oder CT-Aufnahmen zusammensetzt. Für einzelne Strukturen wie Organe, Gefäße oder Muskeln werden in diesen Daten binäre Segmentierungsmasken erzeugt. Mit Hilfe dieser Masken wird festgelegt, welcher Voxel zu welcher Struktur zugeordnet ist. Anschließend erfolgt auf Basis dieser Segmentierung eine Triangulierung der Oberfläche durch gängige Verfahren wie *Marching Cube* (LORENSEN UND CLINE [1987]). Daraus resultieren folgende zwei charakteristische Erscheinungen der Oberflächengeometrie:

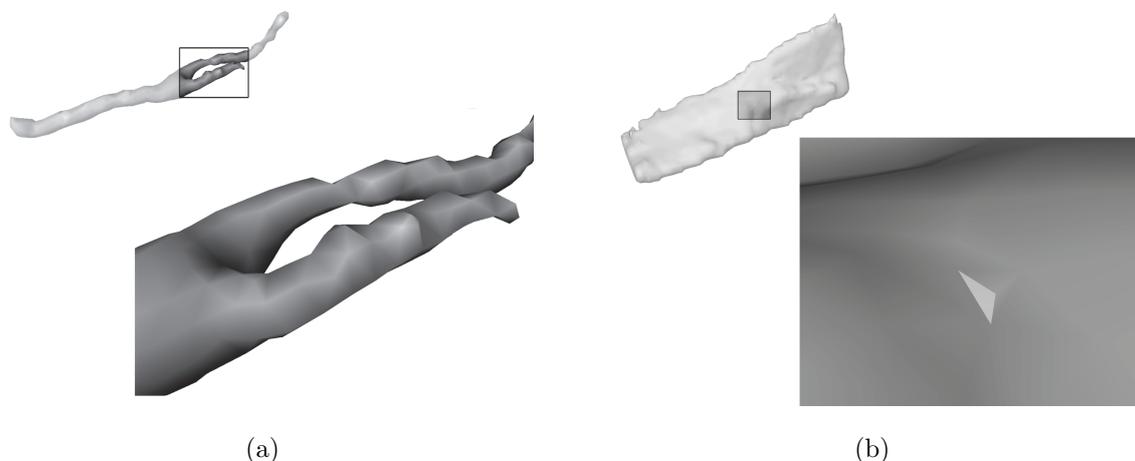
- **Raue Modelloberflächen**

Die resultierenden Oberflächen weisen teilweise einen unregelmäßigen Verlauf auf, wie in der Vergrößerung von Abb. 4.1 (a) am Beispiel einer Halsschlagader zu sehen ist. Der Grund dafür liegt zum einen an der binären Segmentierung der Objekte und zum anderen an den gering aufgelösten Schichtbildaufnahmen. Die binäre Zuordnung eines Voxels zu einem Objekt während der Segmentierung bewirkt eine teilweise scharfkantige Approximation der tatsächlichen Oberfläche. Der auftretende Partialvolumeneffekt erschwert zusätzlich die Segmentierung und hat einen negativen Einfluss auf das Ergebnis. Für die anschließende Triangulierung eines Objektes wird ausschließlich dessen Segmentierungsmaske in Kombination mit den relativ großen und anisotropen Voxelabständen verwendet. Dadurch treten jedoch grobe, scharfkantige sowie teilweise treppenstufenartige Oberflächenverläufe auf. In den meisten Fällen erfolgt keine Glättung der Modelle, da diese für Abstandsbestimmungen im Kontext einer kombinierten Visualisierung mit anderen Strukturen verwendet werden. Eine Glättung der Oberfläche würde hier zu Lasten der Genauigkeit führen.

- **Artifizielle Bereiche**

Artifizielle Bereiche entstehen entweder während der Triangulierung oder nachträglichen Operationen zur Geometrieaufbereitung. Sie beinhalten die bereits oben erwähnten scharfen Objektkanten sowie degenerierte Oberflächenpunkte. Letztere äußern sich in Form von Löchern und Randkanten wie dies in Abb. 4.1 (b) dargestellt ist. Die Vergrößerung zeigt den Oberflächenausschnitt eines Muskels, in dem ein Loch durch eine fehlende Dreiecksfläche zu sehen ist. Entstehen kann dieses durch eine nachträgliche Geometrievereinfachung oder nicht topologieerhaltene Glättung.

Die beiden oben aufgezeigten Charakteristiken verursachen zwei Probleme. Zum einen erschweren sie eine korrekte Krümmungsapproximation. Die scharfen Objektkanten sowie die stark abweichenden Oberflächenverläufe verfälschen die lokale Krümmung. Der Grund dafür liegt in der unregelmäßigen Punktumgebung  $nhd(P_i)$ , auf deren Basis die Approximationsverfahren die Krümmung bestimmen. An degenerierten Punkten können zudem keine Krümmungsinformationen bestimmt werden, da dort keine 2-Mannigfaltigkeit besteht (vgl. Abschnitt 3.2). Zum anderen erschwert die raue, unregelmäßige Oberfläche sowie nicht einheitliche Objektform die Ermittlung einer angemessenen Parametrisierung der Oberfläche, um eine verzerrungsminimierte Texturierung zu ermöglichen. In Abschnitt 2.4.2 wurde bereits erläutert, dass ein indirekter Parametrisierungsansatz für alle anatomischen Strukturen dabei ungeeignet ist, weshalb für diese Arbeit an das Verfahren von PRAUN U. A. [2000] angeknüpft wird.



**Abb. 4.1:** Beispiele der in dieser Arbeit verwendeten Oberflächenmodelle. In (a) ist die Vergrößerung einer Halsschlagader zu sehen, wo der raue Oberflächenverlauf erkennbar ist. Ein Beispiel für degenerierte Bereiche in Form eines Loches im Dreiecksgitter wird in (b) durch einen vergrößerten Halsmuskulbereich gezeigt. Entstehen können diese Bereiche durch nachträgliche Operationen auf der Geometrie, wie einer Geometrievereinfachung oder Glättung.

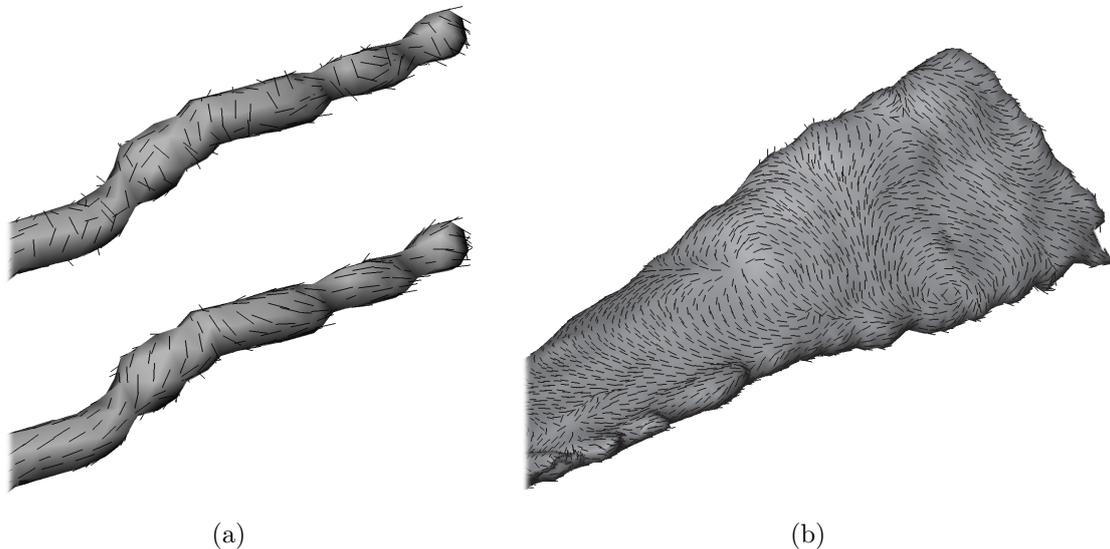
#### 4.1.2 Beschreibung der Schraffurrichtung anatomischer Strukturen

Die meisten der untersuchten Schraffierungsverfahren verwenden für die Beschreibung des Linienverlaufes ein Vektorfeld, das einen möglichst geodätischen Verlauf besitzt. In dem Parametrisierungsansatz von PRAUN U. A. [2000] fließt ebenfalls ein zugrundeliegendes Vektorfeld in die Ausrichtung der Texturen ein. Da in der vorliegenden Arbeit dieser Ansatz weiterverfolgt wird, ist auch hier die Erzeugung eines derartigen Vektorfeldes notwendig. Dieses muss im Sinne des Betrachters korrekt bestimmt werden. Die Korrektheit bezieht sich dabei auf die wahrnehmungspsychologischen und anatomischen Untersuchungen aus den Abschnitten 2.2.2 bzw. 2.2.3. Diese zeigten, dass die Schraffurrichtung zum einen Informationen über den Oberflächenverlauf vermittelt und zum anderen für anatomische Strukturen nicht immer gleich verläuft. Daher muss das Vektorfeld einerseits korrekt die Oberflächeninformation vermitteln und andererseits dem vom Betrachter erwarteten und vertrauten Verlauf entsprechen.

Das dafür allein Krümmungsinformationen nicht ausreichen, soll im Folgenden für die in diesem Kapitel betrachteten anatomischen Strukturen Blutgefäße bzw. allgemein Gefäßsysteme, Muskeln, Organen und Knochen gezeigt werden.

## Gefäßsysteme

Bei Gefäßsystemen verläuft die Schraffurrichtung stets entlang der radialen Richtung um das Gefäß. Da es sich bei diesen um zylinderähnliche Formen handelt, entspricht deren Schraffurverlauf im Allgemeinen der ersten Hauptkrümmungsrichtung (vgl. Abschnitt 3.1.2). In Abb. 4.2 (a) ist dieser Richtungsverlauf dargestellt. Es ist zu erkennen, dass aufgrund der rauen Oberfläche kein geodätisches Vektorfeld erzeugt wird. Durch Glättung des Feldes über eine  $k$ -Nachbarschaftsumgebung wird der Verlauf zwar besser aneinander angeglichen, jedoch weicht der Richtungsverlauf von dem radialen Umlauf ab. Zudem werden markante Oberflächenverläufe durch die Glättung nicht mehr korrekt beschrieben. Hier besteht somit die Notwendigkeit, Modellwissen über den radialen Verlauf einfließen zu lassen.



**Abb. 4.2:** Darstellung eines krümmungsbasierten Vektorfeldes für (a) eine Halsschlagader und (b) einen Halsmuskel. In beiden Abbildungen ist jeweils die ersten Hauptkrümmungsrichtung abgebildet. Zusätzlich ist diese in (b) und der unteren Abbildung von (a) über eine  $k$ -Nachbarschaft geglättet. Erkennbar sind starke Richtungsabweichungen und kritische Punkte im Vektorfeld, die bei beiden Strukturen nicht der jeweils gewünschten Vorzugsrichtung entsprechen.

## Muskelstrukturen

Die Schraffurrichtung bei Muskelstrukturen entspricht in der Regel der Ausrichtung entlang der Kontraktions- und Dehnungsrichtung. Festzustellen ist, dass diese dabei durch Lagebeziehung von Muskelursprung zu -ansatz definiert wird. Wie Abb. 4.2 (b) jedoch

zeigt, kann die Lagebeziehung nicht über Krümmungsinformationen bestimmt werden. Zu sehen ist die Oberfläche einer Muskelstruktur auf der die geglättete erste Hauptkrümmungsrichtung dargestellt ist. Zusätzlich ist die eigentliche Kontraktions- und Dehnungsrichtung durch einen weißen Pfeil angedeutet. Erkennbar sind kritische Punkte, wie Strudel und Quellen in dem erzeugten Vektorfeld, die zwar die lokale Krümmung widerspiegeln, jedoch nicht die erwartete Muskelfaserrichtung. Auch lassen sich durch die Krümmungsinformationen, die aus Abschnitt 2.2.3 getroffene Muskelformen nicht beschreiben (vgl. Abb. 2.13 auf Seite 20). Bei diesen wird jedoch deutlich, dass Krümmungsinformationen in den Linienverlauf einfließen, ihn aber nicht dominieren. Im Zusammenhang mit den Muskelformen lassen sich folgende zwei Klassifikationstypen unterscheiden:

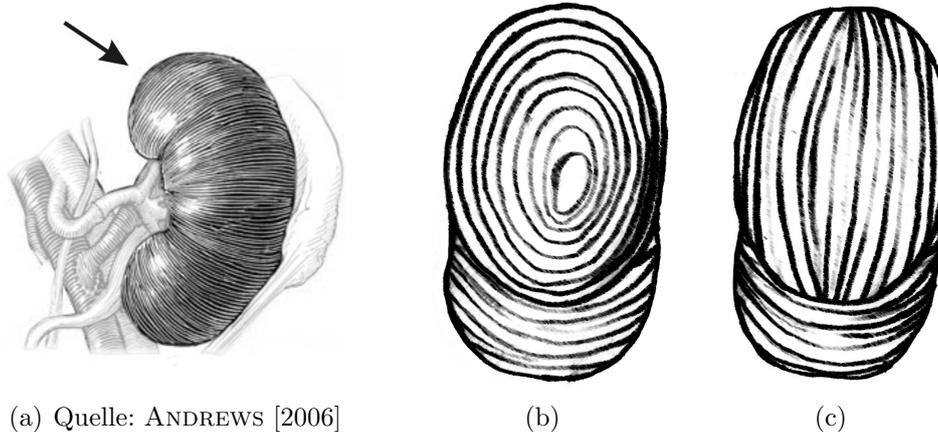
- **Elongierte Muskeln:** Ursprung und Ansatz sind parallel oder konzentrisch zueinander und bilden damit eine teilweise segmentartige Unterteilung des Muskels (z. B. *Musculus intersectus*). Die Muskelfasern verlaufen in der Regel entlang der längsten Objektausdehnung und nahezu orthogonal zu Ursprung und Ansatz. Dazu gehören ring-, hohl- und spindelförmige, mehrbäuchige sowie mehrköpfige Muskeln.
- **Federförmige Muskeln:** Ursprung und Ansatz liegen schräg zueinander, wodurch sich eine schräge Faserrichtung bezüglich Ursprung und Ansatz ergibt (z. B. *Musculus bibennatus*). Des Weiteren existiert ein hohes Verhältnis von Objektbreite zu Objektstärke, d. h. der einschließende Bereich ist im Allgemeinen flach. Als Muskelform zählen dazu die federförmigen Muskelstrukturen.

Es zeigt sich also, dass für das zu erzeugende Vektorfeld bei Muskelstrukturen vorrangig Modellinformationen über Muskelursprung und -ansatz notwendig sind. Ergänzt werden diese durch Krümmungsinformationen, um lokale Oberflächenmerkmale einfließen zu lassen. Inspirierend sind dazu die Ansätze von RÖSSL UND KOBELT [2000] sowie DONG U. A. [2003] bezüglich der Erzeugung eines Vektorfeldes durch Kombination von Krümmungsinformationen und definierter Vorzugsrichtung zur individuellen Beschreibung von Modelloberflächen (vgl. Abschnitt 2.3.3).

## Organe und Knochen

Die Analyse von Schraffuren in anatomischen Illustrationen aus Abschnitt 2.2.3 zeigt, dass die Linien bei Organen und Knochen im Gegensatz zu Gefäßen und Muskeln blickpunkt-abhängig gezeichnet werden. Der Illustrator wählt aus der momentanen Perspektive den Linienverlauf, der die Oberflächenform am besten beschreibt. Am Beispiel einer Magenillustration in Abb. 4.3 wird diese Beobachtung erläutert. In (a) ist die Originalschraffur des Magens dargestellt. Wird das Organ aus dem gekennzeichneten Blickwinkel unter Beibehaltung des Linienverlaufes betrachtet, würde sich dieser wie in (b) ergeben. Diese Illustration wirkt allerdings verwirrend für den Betrachter und vermittelt durch die konzentrischen Ringe keine korrekten Oberflächeninformationen. Ein Illustrator würde daher den Magen bezüglich dieser Perspektive ähnlich der Abbildung (c)

zeichnen und einen anderen Linienverlauf im betreffenden Organbereich wählen. Durch diesen wird der konvexe Oberflächenverlauf besser wahrgenommen. Das gleiche trifft auch für Knochenstrukturen zu. Für eine Einzelbilddarstellung, wie sie in einer anatomischen



**Abb. 4.3:** Der in (a) schraffierte Magen würde aus dem gekennzeichneten, neuen Blickwinkel, bei gleichem Linienverlauf, ungefähr der in (b) dargestellten Zeichnung entsprechen. Die konzentrisch angeordneten Schraffurlinien beschreiben nicht korrekt den konvexen Oberflächenverlauf. Ein Illustrator würde für diesen Organbereich den in (c) gezeigten Linienverlauf wählen, um den Oberflächenverlauf hervorzuheben.

Illustration zu finden ist, ergeben sich aus dem sich ändernden Richtungsverlauf keine Probleme. Diese entstehen jedoch, wenn nach dem gleichen Prinzip Organe oder Knochen in einer interaktiven Visualisierung schraffiert werden sollen. Die Schraffurlinien würden bei jeder Änderung des Blickpunktes ihre Position auf der Oberfläche ändern, wodurch die *Frame*-Kohärenz nicht mehr gewährleistet ist. Wie im Abschnitt 2.3.1 festgelegt wurde, ist dieser Aspekt eine wichtige Anforderung in einer interaktiven Darstellung.

Diese Beobachtung stellt somit ein Problem bezüglich der Generierung eines einheitlichen Vektorfeldes dar. Erschwerend für eine formale bzw. modellbasierte Ableitung der Schraffurrichtung ist dabei, dass der Richtungsverlauf neben Krümmungsinformationen durch das ästhetische Empfinden des Illustrators bestimmt wird. In dieser Arbeit wird dennoch ein Konzept vorgeschlagen, das eine *Frame*-Kohärenz sicherstellt und dabei den in Abb. 4.3 (c) gezeigten Richtungsverlauf anstrebt. Die weitere Betrachtung dieser Strukturen wird auch durch die Untersuchungen von INTERRANTE U. A. [1996], INTERRANTE U. A. [1997] sowie INTERRANTE [1997] bezüglich eingebundener Oberflächenvisualisierungen motiviert. Die darin festgestellte Unterstützung räumlicher Lagebeziehungen durch Schraffierung der halbtransparenten Außenobjekte stellen auch für die illustrative Darstellung in der medizinischen Visualisierung eine interessante Anwendung dar.

## 4.2 Anforderungsanalyse und Übersicht des Entwurfes

Aus den bisherigen Untersuchungen und Vorbetrachtungen ergeben sich für das hier umzusetzende Schraffierungsverfahren weitere Anforderungen, zusätzlich zu denen einer interaktiven Visualisierung. Im Folgenden werden diese zusammengefasst. Daraus resultieren einzelne, zu lösende Teilprobleme, die der Abschnitt 4.2.2 in Form eines *Work-flow*-Diagrammes darstellt. Jeder dieser Verarbeitungsschritte wird dann in dem sich anschließenden Entwurf vorgestellt.

### 4.2.1 Anforderungsanalyse

Die folgende Analyse bezieht sich auf die Anforderungen bezüglich der interaktiven Schraffierung, dem zu erzeugenden Vektorfeld, die Oberflächenparametrisierung sowie die anschließende Texturierung.

#### Interaktive Schraffierung

Zunächst müssen wesentliche Anforderungen erfüllt sein, die an eine interaktive Visualisierung gestellt werden. Diese wurden in Abschnitt 2.3.1 eingeführt und dort auf eine interaktive Schraffierung übertragen. Dazu zählen die:

- **Frame-Kohärenz,**
- **Skalierbarkeit,**
- **Echtzeitfähigkeit** sowie
- **automatische Parameterbestimmung.**

#### Krümmungs- und modellbasiertes Vektorfeld

Die Ausrichtung der Schraffurtexturen orientiert sich an ein zugrunde liegendes Vektorfeld. Daher ist dessen korrekter Verlauf eine wichtige Voraussetzung der späteren Parametrisierung und Texturierung. Das Vektorfeld besteht dabei aus Krümmungs- und Modellinformationen der zu schraffierenden anatomischen Struktur. Dafür muss zunächst eine Krümmungsapproximation der Oberfläche erfolgen, die robust gegenüber den rauen Oberflächenverlauf und degenerierten Bereichen ist. Des Weiteren ist eine mögliche modellspezifische Vorzugsrichtung zu ermitteln, d. h. bei Gefäßen den radialen Verlauf sowie bei Muskelstrukturen die Kontraktions- und Dehnungsrichtung, abhängig von der Lagebeziehung von Ursprung und Ansatz. Gemäß der Anforderung an eine automatische Parameterbestimmung muss diese Richtungsinformation automatisch extrahiert werden.

Im Anschluss ist eine geeignete Zusammenführung dieser beiden Richtungsinformationen notwendig, so dass ein geodätisch verlaufendes Vektorfeld erzeugt werden kann. Für Organe und Knochen existiert keine derartige, modellspezifische Vorzugsrichtung.

### **Oberflächenparametrisierung und Texturierung**

Der texturbasierte Ansatz erfordert eine geeignete Oberflächenparametrisierung zur korrekten Abbildung der Schraffierungstextur auf die Objektoberfläche. Geeignet bedeutet in diesem Sinne einerseits die Minimierung der Texturverzerrungen und andererseits das Integrieren des zugrunde liegende Vektorfeldes. Letzteres ist notwendig, damit die Texturen Informationen über den Oberflächenverlauf vermitteln. Auch hier muss die Parametrisierung robust gegenüber den Oberflächenunebenheiten sein sowie ohne Benutzerinteraktion stattfinden.

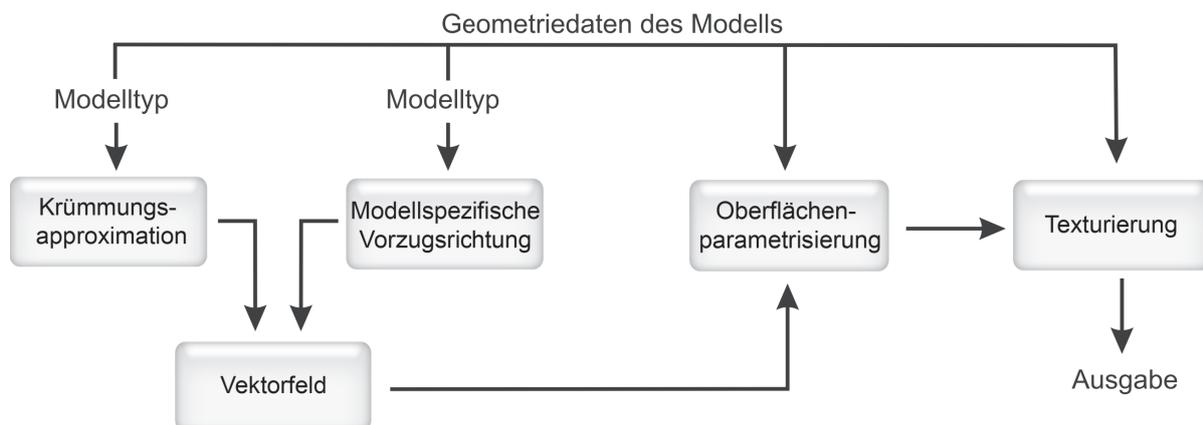
Da mit Schraffuren, neben dem Oberflächenverlauf, auch Informationen über die Beleuchtungssituation vermittelt werden können (vgl. Abschnitt 2.2.1), muss diese ebenfalls angemessen integriert werden. Ein gängiges Gestaltungsmittel ist dabei die Erhöhung der Liniendichte mit abnehmender Beleuchtung. Dieses Prinzip soll für das Verfahren übernommen werden. Des Weiteren muss für die Interaktion die Kohärenz der Skalierbarkeit für die Schraffierung gegenüber dem Objektabstand zum Betrachter gewährleistet sein. Dies entspricht einer sich ändernden Anzahl von Schraffurlinien, abhängig vom Objektabstand. Somit sind entsprechende Helligkeits- und Auflösungstexturen notwendig. Hierfür wird das *TAM*-Konzept von PRAUN U. A. [2001] übernommen (vgl. Abschnitt 2.3.4).

### **Zusammenfassung**

Das in dieser Arbeit vorgestellte Verfahren soll Schraffierungen auf triangulierten, patientenspezifischen Oberflächenmodellen erzeugen. Grundlage für das Verfahren ist dabei der texturbasierte Ansatz von PRAUN U. A. [2000], wo die Schraffurtexturen anhand eines Vektorfeldes ausgerichtet werden. Für die hier betrachteten anatomischen Strukturen setzt sich dieses bei Gefäß- und Muskelstrukturen aus Krümmungsinformationen und einer modellspezifischen Vorzugsrichtung zusammen. Im Fall von Organen und Knochen entspricht dies der ersten Hauptkrümmungsrichtung. Wichtig ist dabei ein korrekter, geodätischer Richtungsverlauf. Die dafür notwendigen Informationen müssen robust und automatisch aus den Geometriedaten sowie anhand des Modelltypes erzeugt werden. Des Weiteren bietet der lokale Parametrisierungsansatz von PRAUN U. A. [2000] ein geeignetes Vorgehen bei der verzerrungsminimierten Erzeugung von Parameterkoordinaten auf den rauen Modelloberflächen.

### 4.2.2 Entwurfsübersicht

Die aus dem vorhergehenden Abschnitt erarbeiteten Anforderungen repräsentieren verschiedene Verarbeitungsschritte des Verfahrens, welche im Weiteren als Komponenten betrachtet werden. Zusammengesetzt werden diese zu einem *Workflow*, der in Abb. 4.4 dargestellt ist. Außer für die Berechnung des endgültigen Vektorfeldes dienen zunächst die 3D-Geometriedaten als Basiseingabe für die einzelnen Komponenten. Zu Beginn werden daraus die Krümmungsinformationen der Oberfläche approximiert und zur initialen Erzeugung des Vektorfeldes verwendet. Die Wahl der entsprechenden Approximationsmethode richtet sich dabei nach dem Typ des zu schraffierenden, anatomischen Modells. Mithilfe des Modelltyps wird weiterhin die entsprechende Vorzugsrichtung auf der Struktur ermittelt. Diese fließt ebenfalls in die Komponente zur Erzeugung des Vektorfeldes ein, in Kombination mit den Krümmungsinformationen. In der darauf folgenden Oberflächenparametrisierung werden die Parameterkoordinaten für jeden Oberflächenpunkt erzeugt, wobei das Vektorfeld für deren korrekten Ausrichtung im Parameterraum verwendet wird. Mit Hilfe der Parametrisierung erfolgt anschließend bei der Texturierung die Abbildung der entsprechenden Schraffurtexturen auf das Modell, abhängig von der lokalen Beleuchtung und Kameraentfernung.



**Abb. 4.4:** *Workflow*-Darstellung des in dieser Arbeit vorgestellten Verfahrens zur Schraffierung patientenspezifischer Oberflächenmodelle. Die Übersicht zeigt kooperierende Komponenten zur Erzeugung des Vektorfeldes, der Parametrisierung und anschließenden Texturierung. Bezüglich des Vektorfeldes legt das jeweilige anatomische Modell die Wahl des Krümmungsapproximationsverfahrens und der Vorzugsrichtung fest.

Bis auf die Texturierung bilden die genannten Verarbeitungsschritte eine Vorverarbeitung für jede anatomische Oberfläche. Diese muss nur einmal erfolgen und kann für jede erneute Darstellung wieder verwendet werden. Im Folgenden wird der Entwurf jeder Komponente beschrieben, wobei dies in einer generellen Einteilung von Erzeugung des Vektorfeldes, der Oberflächenparametrisierung sowie der Texturierung erfolgt.

### 4.3 Generierung des krümmungs- und modellbasierten Vektorfeldes

Dieser Abschnitt behandelt die einzelnen Verarbeitungsschritte zur Erzeugung eines krümmungs- und modellbasierten Vektorfeldes. Wie bereits erwähnt, sind für das hier zu entwickelnde Verfahren einerseits Krümmungsinformationen an jedem Oberflächenpunkt und andererseits die Vorzugsrichtung für Gefäß- und Muskelstrukturen notwendig. Die Krümmungsapproximation auf triangulierten Oberflächen von radiologischen Schichtbilddaten anatomischer Strukturen basiert dabei auf der Diplomarbeit von SCHULZ [2005]. In dieser findet zusätzlich eine Untersuchung und Bewertung der umgesetzten Verfahren für die verwendeten anatomischen Strukturen statt. Ausgangsbasis für die zusätzliche Ermittlung der jeweiligen Vorzugsrichtungen liefern die Beobachtungen aus Abschnitt 2.2.3 und die am Anfang dieses Kapitels erfolgten Vorbetrachtungen. Des Weiteren wird auf die Zusammenführung der gewonnenen Informationen zum endgültigen Vektorfeld und dessen nachträglicher Optimierung eingegangen.

Ein Aspekt, der für die Bestimmung der Krümmungsapproximation berücksichtigt werden muss, ist der beobachtete raue und teilweise artifizielle Oberflächenverlauf. Dadurch spiegeln die ermittelten Krümmungsinformationen zwar das lokale Krümmungsverhalten wider, weichen jedoch bezüglich ihrer Nachbarschaft teilweise stark von dieser ab. Zusätzlich verursachen „Ausreißer“, in Form von scharfen Ecken und Kanten, in den Oberflächenpunkten lokale Verfälschungen in den Krümmungsapproximationen. In der Gesamtheit ergibt sich daraus ein Rauschen in den ermittelten Krümmungsinformationen. Daher erfolgt für die Krümmungsapproximationen zunächst eine Glättung der Oberfläche, um das Rauschen zu minimieren.

#### 4.3.1 Glättung der Oberfläche für die Krümmungsapproximation

Die Glättung der Oberfläche bezieht sich zum einen auf die Dreieckseckpunkte und zum anderen auf deren Normalenrichtung. Die Unregelmäßigkeit in den Normalenrichtungen ist dabei ebenfalls eine Folge der Oberflächenextraktion aus den Segmentierungsmasken der Volumendaten. Mit der Glättung der Oberflächenpunkte wird eine Verminderung der Rauigkeit und scharfkantigen Bereiche bewirkt. Die Glättung der Normalen ist notwendig, da die meisten Approximationsverfahren aus Abschnitt 3.2, insbesondere alle *Curve Fitting* Verfahren, die Normalenrichtung für ihre Krümmungsapproximation benötigen.

Für die Glättung der Eckpunkte wird ein topologieerhaltender Ansatz in Form einer iterativen Laplace-Glättung verwendet. Diese verschiebt einen Oberflächenpunkt  $v_i$  in das gewichtete Positionszentrum seiner direkten Nachbareckpunkte  $v_j \in nhd(v_i)$ :

$$P'(v_i) = P(v_i) + w \left( \left( \frac{1}{n} \sum_{j=0}^n P(v_j) \right) - P(v_i) \right) \quad (4.1)$$

Der Wichtungsfaktor  $w \in [0, 1]$  gibt dabei die Stärke der Verschiebung an, wobei 0 keine und 1 Verschiebung zum Zentrum bedeutet. Neben  $w$  ist die Glättung abhängig von der Anzahl der Iterationen von Gleichung 4.1 auf alle Eckpunkte der Oberfläche. Je mehr Durchgänge verwendet werden, desto stärker wird die Oberfläche geglättet. Jedoch schrumpft diese dabei auch, wodurch eventuell wichtige Oberflächendetails, wie beispielsweise *Aneurysmen* (krankhafte Gefäßerweiterungen), nicht mehr als solche wahrgenommen werden. Daher ist eine angemessene Einstellung der beiden Parameter zu wählen.

Die Glättung der Normalenrichtung  $\hat{n}(v_i)$  für einen Eckpunkt  $v_i$  erfolgt ebenfalls topologierhaltend in Form einer normierten Mittelung seiner direkten Nachbarnormalen  $\hat{n}(v_j) \in nhd(v_i)$ :

$$\begin{aligned} \vec{n}(v_i) &= \frac{1}{n} \sum_{j=0}^n \hat{n}(v_j) \\ \hat{n}(v_i) &= \vec{n}(v_i) / \|\vec{n}(v_i)\| \end{aligned} \tag{4.2}$$

Die Mittelung wird iterativ ausgeführt, so dass die Anzahl der Iterationsschritte ebenfalls angemessen gewählt werden muss. Ein zu starkes Glätten würde die Normalenrichtung an  $v_i$  zu sehr seiner Umgebung angleichen und damit verfälschen.

Für die Krümmungsapproximation wird zunächst eine Kopie der Oberfläche angelegt, auf welcher die Glättung nach dem oben beschriebenen Vorgehen angewendet wird. Anschließend werden die Krümmungsinformationen für jeden Eckpunkt auf der geglätteten Kopie approximiert und auf sein Originaleckpunkt übertragen. Die Weiterverwendung der Originaloberfläche ist deshalb notwendig, da bei der Laplace-Glättung schon bei einem Iterationsschritt das Modell schrumpft. Dieser Effekt ist aber für die weitere Verwendung der Strukturen nicht erwünscht bzw. führt auch zu falschen Rückschlüssen. Letzteres tritt beispielsweise bei der Abstandsbestimmung in einer späteren kombinierten Darstellung mit anderen Kontextstrukturen auf.

### 4.3.2 Krümmungsapproximationen der anatomischen Strukturen

Damit die späteren Schraffurlinien Informationen über den Oberflächenverlauf vermitteln können, müssen diese Informationen zunächst ermittelt werden. Für diese Arbeit von Interesse sind dabei die beiden Hauptkrümmungsrichtungen  $t_{\kappa_1}, t_{\kappa_2}$ , da diese den lokalen Oberflächenverlauf an einem Oberflächenpunkt beschreiben. Die Arbeit von SCHULZ [2005] dient in diesem Zusammenhang als Vorarbeit für die hier umzusetzende Aufgabenstellung. Die einzelnen Verfahren wurden im Abschnitt 3.2 beschrieben und zusätzlich sind dazu erste Schlussfolgerungen für deren Anwendung getroffen worden. Für die jeweilige anatomische Struktur müssen davon geeignete ausgewählt werden. Dabei sind die in Abschnitt 3.3 getroffenen drei Aspekte zu beachten:

- **Robustheit**

- **Genauigkeit** sowie
- **Berechnungszeit.**

Für den Kompromiss zwischen Genauigkeit und Berechnungszeit der Approximationsverfahren gilt dabei, dass bei Gefäß- und Muskelstrukturen schnellere Verfahren den aufwendigeren aber genaueren Methoden vorgezogen werden, wenn diese nur leichte Abweichungen in den Krümmungsinformationen aufweisen. Dies begründet sich darin, dass für diese Strukturen Krümmungsinformationen nur einen partiellen Einfluss auf das Vektorfeld haben und in die Vorzugsrichtung einfließen, also nicht dominierend sind. Jedoch ist hier die Komplexität (Anzahl von elliptischen, parabolischen und hyperbolischen Punkten) bezüglich des Oberflächenverlaufes zu berücksichtigen und dafür geeignete Verfahren auszuwählen. Hingegen besteht das Vektorfeld bei Organen und Knochen nur aus der ersten Hauptkrümmungsrichtung. Diese muss daher so genau wie möglich ermittelt werden, wodurch genauere aber langsamere Verfahren den einfacheren vorzuziehen sind.

Folgende zwei Möglichkeiten bieten sich für die Krümmungsapproximation an, die die obigen drei Aspekte berücksichtigen.

### Interpolation fehlender Krümmungswerte

Ein relativ einfaches Vorgehen besteht zunächst in der Wahl einer Approximationsmethode, die für die gegebene Struktur geeignete Informationen bestimmt. Treten dabei Oberflächenbereiche auf, wo dies aufgrund mangelnder Nachbarpunkte oder degenerierter Bereiche nicht möglich ist, so werden die fehlenden Krümmungsinformationen durch Interpolation ermittelt. Dazu findet an einem solchen Punkt  $v_i$  eine Mittelung der Krümmung aus der direkten, validen Nachbarschaft statt. Dazu wird eine Nachbarschaft  $nhd(v_i)$  mit dem topologischen Radius „ $r=1$ “ definiert, in dem alle direkten Nachbarpunkte  $v_j$  enthalten sind, für die gültige Krümmungsinformationen vorliegen. Dieses Vorgehen beruht auf der Annahme, dass das Krümmungsverhalten an einen Punkt  $v_i$  sich nicht stark von dem seiner direkten Nachbarn unterscheidet. Die Mittelung von  $\kappa_1, \kappa_2$  an einem Punkt  $v_i$  geschieht dabei durch

$$v_i(\kappa_1, \kappa_2) = \frac{1}{n} \sum_{j=0}^n v_j(\kappa_1, \kappa_2), \quad (4.3)$$

wobei  $v_j \in nhd(v_i)$  und  $n = |nhd(v_i)|$ . Die Mittelung der Hauptkrümmungsrichtungen  $\hat{t}_{\kappa_1}, \hat{t}_{\kappa_2}$  geschieht über die projizierten Richtungen von  $v_j$  in die Tangentialebene von  $v_i$ , welche durch seine Position und Normale aufgespannt wird:

$$v_i(\hat{t}_{\kappa_1}, \hat{t}_{\kappa_2}) = \frac{1}{n} \sum_{j=0}^n T_i(v_j(\hat{t}_{\kappa_1}, \hat{t}_{\kappa_2})). \quad (4.4)$$

$T_i$  beschreibt dabei die Projektion der benachbarten Richtungsvektoren in die Tangentialebene von  $v_i$ .

Der Vorteil dieses Vorgehens liegt in der schnellen Berechnung der Krümmungsinformationen für die gesamte Oberfläche. Aufgrund der Mittelung kann es allerdings bei den betroffenen Oberflächenpunkten zu Ungenauigkeiten kommen, die größer werden, je stärker sich das Krümmungsverhalten ändert. Letzteres tritt beispielsweise bei scharfen Kanten oder Spitzen auf.

### **Adaptive Krümmungsapproximation**

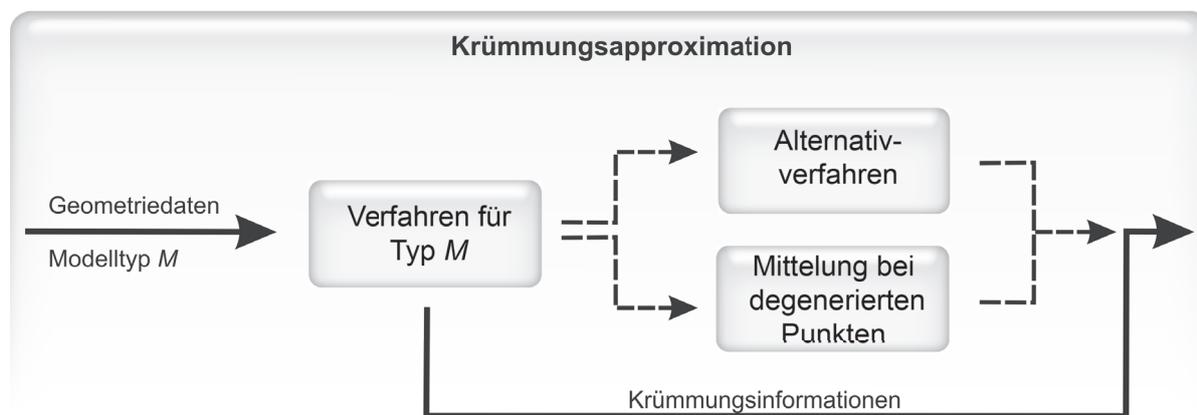
Eine andere Möglichkeit stabile Krümmungsinformationen zu erzeugen, besteht in einem adaptiven Verfahren, welches in Abb. 4.5 dargestellt ist. Je nach Art der Struktur wird zunächst wieder das Verfahren gewählt, das bezüglich der zu erwartenden Oberflächenkomplexität, geeignete Krümmungsinformationen ermittelt. An Oberflächenpunkten, wo aus den gleichen Gründen wie beim Interpolationsansatz keine Krümmungen bestimmt werden konnte, erfolgt eine Separierung und weitere Behandlung dieser Bereiche. Die Punkte, die nicht genügend Nachbarpunkte für das erste Verfahren besitzen, werden einem zweiten Verfahren übergeben, das mit der gegebenen Anzahl auskommt.

Die Krümmung an degenerierten Punkten kann mit keinem der vorhandenen Verfahren ermittelt werden. In diesem Fall erfolgt eine Mittelung der Krümmungsinformationen durch die direkten, nicht degenerierten Nachbarpunkte. Die Mittelung erfolgt nach dem gleichen Prinzip, wie es in der obigen Interpolation fehlender Krümmungswerte beschrieben ist. Auch hier gilt wieder die gleiche Annahme wie bei der Interpolation. Zusätzlich begründet sich dieses Vorgehen damit, dass in der Regel degenerierte Punkte durch Rekonstruktionsartefakte entstehen und diese als artifiziell zu betrachten sind. Die Krümmungsinformationen müssen daher mittels der Umgebung approximiert werden. Die Motivation dieses adaptiven Ansatzes liegt gegenüber der vorherigen Interpolation in einer höheren Genauigkeit der berechneten Krümmungswerte an kritischen Oberflächenpunkten. Erkauft wird die Genauigkeit durch einen erhöhten Berechnungsaufwand, da die Berechnungen für das zweite Verfahren in der Regel aufwendiger sind als eine einfache Mittelung.

#### **4.3.3 Auswahl der Approximationsverfahren**

Für das hier umzusetzende Verfahren wird der adaptive Ansatz verwendet. Zum einen begründet sich dies in der höheren Genauigkeit der ermittelten Krümmungsinformationen. Zum anderen beruht die Entscheidung auf der Beobachtung, die auch in SCHULZ [2005] zu finden ist, dass die Anzahl von kritischen Oberflächenpunkten, die separat behandelt werden müssen, im Vergleich zu Gesamtheit aller Oberflächenpunkte sehr gering ist (weniger als 0,2 %). Somit wird keine bemerkbare Erhöhung der Berechnungszeit für die Krümmungsapproximation erwartet.

An dieser Stelle sollen nun die Approximationsverfahren von Abschnitt 3.2 bestimmt werden, die für die hier zu schraffierenden Strukturen am besten geeignet sind. Diese

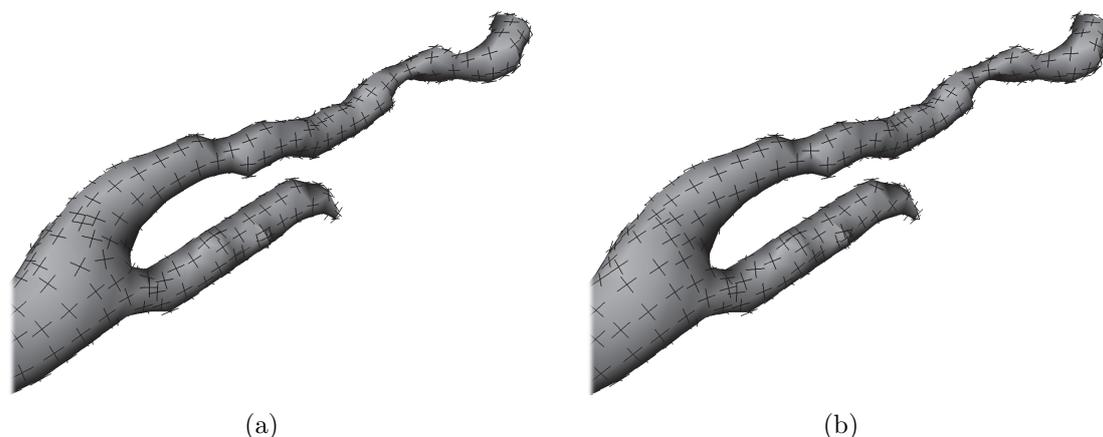


**Abb. 4.5:** Darstellung einer adaptiven Krümmungsapproximation. Für die Oberflächengeometrie eines anatomischen Modells  $M$  werden die Krümmungsinformationen zunächst mit Hilfe eines geeigneten Verfahrens bestimmt. An Punkten, die die geforderte Mindestanzahl an Nachbarpunkten für das Verfahren nicht besitzen, wird ein weiteres Verfahren gewählt, das mit der vorhandenen Anzahl auskommt. Für degenerierte Bereiche werden die Krümmungsinformationen anhand von  $nhd(v_i)$  gemittelt.

sind für die beiden oben beschriebenen Ansätze gleich. Zusätzlich muss für das adaptive Vorgehen ein alternatives Verfahren festgelegt werden, bei denen Krümmungsinformationen an den Oberflächenpunkten bestimmt werden, deren Anzahl von Nachbarpunkten für das „Vorzugsverfahren“ nicht ausreichte.

## Gefäßstrukturen

Die verwendeten Gefäßstrukturen besitzen zum Teil eine komplexe Geometrie in Form von vielen Oberflächenpunkten mit hyperbolischer, elliptischer und zylindrischer Umgebung. Daher wird für diese Strukturtypen ein topologieabhängiges Verfahren gewählt. Laut Tabelle 3.1 aus Abschnitt 3.2 zählen zu diesen Verfahren die *ParametricOP* und *BezierQuadOP* Operatoren. Einen Vergleich zwischen beiden Operatoren bezüglich der ermittelten Hauptkrümmungsrichtungen zeigt Abb. 4.6. Zu sehen sind jeweils die beiden ermittelten Hauptkrümmungsrichtungen an jedem Eckpunkt auf den Oberflächenausschnitt einer Halsschlagader. In (a) wurden diese mit dem *ParametricOP* Operator bestimmt und in (b) mittels dem *BezierQuadOP* Operator. Trotz gleicher Parametereinstellungen sind kaum erkennbare Unterschiede festzustellen. Der *BezierQuadOP* Operator benötigt allerdings im Durchschnitt für die Berechnung ca. fünf mal so lange wie der *ParametricOP* Operator. Aus diesem Grund wird für Gefäßstrukturen der *ParametricOP* Operator verwendet.



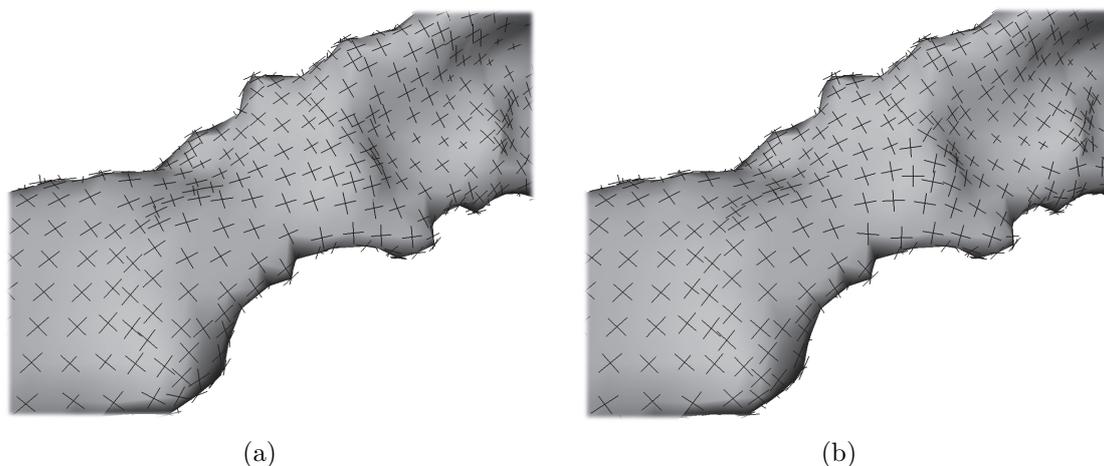
**Abb. 4.6:** Vergleich zwischen den *ParametricOP* (a) und *BezierQuadOP* (b) Operatoren zur Krümmungsapproximation auf einen vergrößerten Ausschnitt einer Halsschlagader. An jedem Oberflächenpunkt sind jeweils nur die beiden Hauptkrümmungsrichtungen dargestellt.

## Muskelstrukturen

Wie im Abschnitt 4.3.2 schon erwähnt, ist die Oberflächengeometrie bei Muskelstrukturen gegenüber Blutgefäßen weniger komplex. Nach SCHULZ [2005] sind dafür schon Verfahren geeignet, die eine topologieunabhängige Parametrisierung der Nachbarschaft besitzen. Aufgrund der großflächigeren Ausbreitung einer Oberflächenorientierung, bieten sich dazu *Surface Fitting* Verfahren an, da hierbei geeignete Passflächen bestimmt werden können. Die *QuadraticOP* und *CubicOP* Operatoren zeigen auf den verwendeten Datensätzen, ein geeignetes Approximationsverhalten mit geringen Abweichungen zueinander. In Abb. 4.7 ist ein Vergleich für eine vergrößerte Halsmuskeloberfläche zu sehen. Zu sehen sind jeweils die beiden ermittelten Hauptkrümmungsrichtungen an jedem Eckpunkt bei Anwendung des *QuadraticOP* (a) und *CubicOP* Operator (b). Da der erste Operator fast doppelt so schnell arbeitet wie der zweite, wird dieser für die Muskelstrukturen verwendet.

## Organe und Knochenstrukturen

Die in dieser Arbeit verwendeten Organe und Knochenstrukturen zeigen überwiegend einen komplexen Oberflächenverlauf. Hierfür eignen sich wiederum die Approximationsverfahren, die eine topologieerhaltene Parametrisierung besitzen. Im Gegensatz zu den Gefäßstrukturen wird hierfür der *BezierQuadOP*-Operator mit erweitertem Nachbarschaftsradius ( $r > 1$ ) verwendet. Dies begründet sich zum einen aus der Beobachtung, dass besonders Knochenstrukturen einen komplexeren Verlauf aufweisen als Gefäßstrukturen. Zum anderen wird Abschnitt 4.3.4 zeigen, dass das resultierende Vektorfeld für Or-

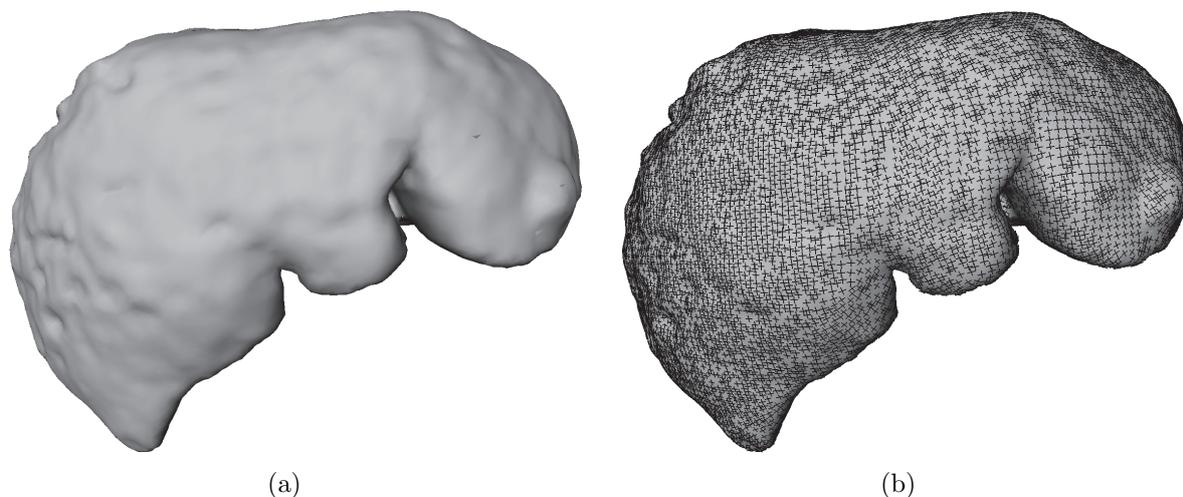


**Abb. 4.7:** Vergleich zwischen den *QuadraticOP* (a) und *CubicOP* (b) Operatoren zur Krümmungsapproximation auf einen vergrößerten Ausschnitt eines Halsmuskels. Auch hier sind an jedem Oberflächenpunkt nur die beiden Hauptkrümmungsrichtungen dargestellt.

gane und Knochen nur aus bestehenden Krümmungsinformationen generiert wird. Daher ist eine möglichst genaue Approximation der Krümmung notwendig. Laut SCHULZ [2005] ist bei *BezierQuadOP*-Operator eine korrekte topologische Parametrisierung garantiert, wenn die erforderliche Anzahl von Nachbarpunkten gegeben ist. Des Weiteren besitzt das zugrundeliegende Berechnungsverfahren eine hohe Flexibilität und damit höhere Genauigkeit in der Oberflächenbeschreibung einer Punktumgebung als der *ParametricOP*-Operator. Die Abb. 4.8 (a) zeigt dazu die Oberfläche einer patientenspezifischen Leber, welche durch einen komplexen Oberflächenverlauf gekennzeichnet ist. In (b) ist zusätzlich das Krümmungsfeld dargestellt, welches durch den *BezierQuadOP*-Operator ermittelt wurde. Im Beispiel ist dafür ein topologischer Radius von  $r = 3$  verwendet worden. Erkennbar sind wieder die beiden Hauptkrümmungsrichtungen an jedem Eckpunkt. Anhand einer visuellen Beurteilung erfolgt eine korrekte Beschreibung der lokalen Oberflächenkrümmung. Der in diesem Zusammenhang erhöhte Berechnungsaufwand ist gegenüber der erhöhten Genauigkeit zu vertreten.

### Operator für den adaptiven Ansatz

Für den adaptiven Ansatz wird als zweites Verfahren der *TaubinOP* Operator verwendet. Zum einen liegt der Vorteil bei diesem darin, dass das Verfahren schon mit drei direkten Nachbarpunkten für die Berechnung von Krümmungsinformationen auskommt. Zum anderen stellt das zugrunde liegende Verfahren in SCHULZ [2005] den schnellsten Operator dar, da die Lösung von Gleichungssystemen nicht erforderlich ist.

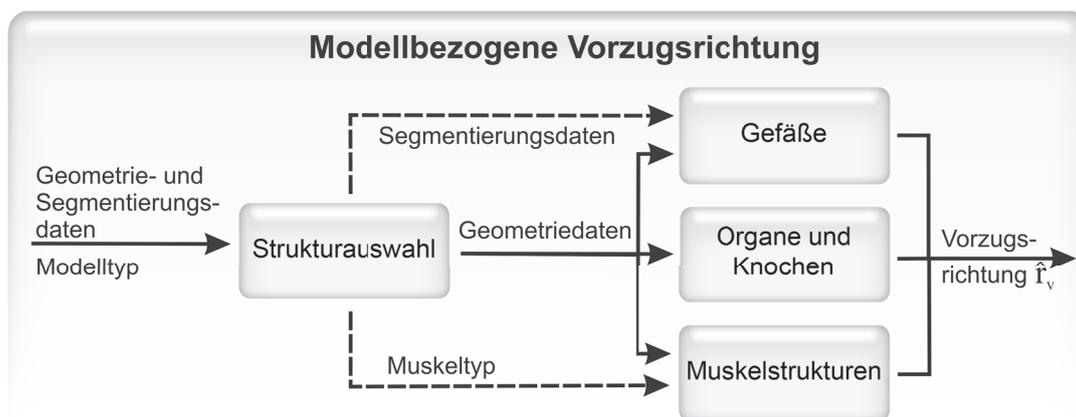


**Abb. 4.8:** (a) Darstellung einer patientenspezifischen Leberoberfläche für deren Eckpunkte in (b) die beiden Hauptkrümmungsrichtungen mit Hilfe des *BezierQuadOP*-Operator bestimmt wurden.

#### 4.3.4 Ermittlung der modellbezogenen Vorzugsrichtung

In diesem Abschnitt wird die Erzeugung der modellbezogenen Vorzugsrichtung für Gefäß- und Muskelstrukturen erarbeitet, entsprechend den Beobachtungen aus den Abschnitten 2.2.3 und 4.1.2. Die Abb. 4.9 zeigt dazu eine Übersicht der verschiedenen Verarbeitungsschritte. Anhand des Modelltyps wird zunächst entschieden für welche der zu schraffierenden Struktur die Vorzugsrichtung bestimmt werden soll. Diese Metainformationen werden bei der Segmentierung der anatomischen Strukturen durch den Benutzer einmal definiert und sind den Daten angefügt. Die Strukturauswahl unterscheidet dabei zwischen Gefäß- und Muskelstrukturen sowie Organe und Knochen. Die beiden letzten Strukturen nehmen eine Sonderstellung in diesem Abschnitt ein. Organe und Knochen weisen nach den Vorbetrachtungen aus Abschnitt 4.1.2 keine eindeutige modellspezifische Vorzugsrichtung auf. In diesem Abschnitt soll dennoch ein Vorgehen beschrieben werden, welches aus den Geometriedaten ein Vektorfeld erzeugt, das einerseits modellspezifisch den Oberflächenverlauf beschreibt und andererseits die *Frame*-Kohärenz sicherstellt.

Neben den Geometriedaten werden für die Bestimmung der Vorzugsrichtung bei Gefäßen und Muskelstrukturen weitere Informationen benötigt. Für Gefäße sind die 2D-Segmentierungsmasken erforderlich und bei den Muskelstrukturen muss zusätzlich angegeben sein, ob es sich um einen elongierten oder federförmigen Muskeltyp handelt. Für jeden Eckpunkt  $v$  der Oberfläche ergibt sich anschließend die modellspezifische Vorzugsrichtung  $\hat{r}_v$ . Im Folgenden soll das jeweilige Vorgehen näher erläutert werden.



**Abb. 4.9:** Übersicht der einzelnen Verarbeitungsschritte zur Bestimmung der modellbezogenen Vorzugsrichtung  $\hat{r}_v$  für einen Eckpunkt  $v$ . Anhand des Modelltyps erfolgt eine Bestimmung für Gefäße, Organe und Knochen oder Muskelstrukturen. Dazu sind neben den Geometriedaten zusätzlich für Gefäße die Segmentierungsdaten und für Muskelstrukturen Informationen über den Muskeltyp notwendig.

### Gefäßstrukturen

Im Abschnitt 4.1.2 ist bereits erwähnt worden, dass bei Gefäßstrukturen der radiale Verlauf der ersten Hauptkrümmungsrichtung entspricht. Anhand der gezeigten Abbildungen ist jedoch zu erkennen, dass eine Festlegung allein auf diese Richtung nicht ausreicht. Das resultierende Vektorfeld verläuft teilweise nicht radial und geodätisch. Aus den Abb. 4.6 (b) und (c) ist aber erkennbar, dass die ermittelten beiden Hauptkrümmungsrichtungen den Oberflächenverlauf auf Gefäßen gut beschreiben. Somit muss an jedem Oberflächenpunkt entschieden werden, welche der beiden Hauptkrümmungsrichtungen der Vorzugsrichtung am ähnlichsten ist.

Ansatzpunkt für die Bestimmung der radialen Vorzugsrichtung ist der Skelettlinienverlauf innerhalb eines Blutgefäßes. In SELLE U. A. [2000] wird dazu ein Verfahren beschrieben, welches diese Skelettlinie mit Hilfe von morphologischen Operationen aus den binären Segmentierungsmasken extrahiert. Für jede Schicht aus den Volumendaten wird dabei der mittlere Voxel eines segmentierten Bereiches bestimmt (*Thinning*) und zu einer Skelettlinie verbunden. Jede Skelettlinie stellt dabei einen Gefäßast bis zur nächsten Verzweigung (Bi- und Trifurkationen) dar und besteht aus einzelnen, diskret platzierten Skelettvoxeln  $s_i \in S$ . Die Menge  $S$  stellt dabei die Anzahl aller Skelettvoxel einer Skelettlinie dar.

Die Glattheit des resultierenden Linienverlaufes ist dabei abhängig vom Schichtabstand und von der Voxelgröße. Je größer beides ist, desto unregelmäßiger und ungenauer ist der Verlauf. In Abb. 4.10 (a) ist beispielsweise die initiale Skelettlinie der Portalvene der Leber zu sehen. Daher wird der Skelettlinienverlauf für diese Arbeit anschließend

durch eine iterative Laplace-Glättung in 3D verbessert (vgl. Abschnitt 4.3.1). Analog zur Oberflächenglättung für die Krümmungsapproximation wird hier die Position eines Skelettvoxels  $s_i$  in das gewichtete Positionszentrum seiner direkten Nachbarknoten  $s_{i-1}$  und  $s_{i+1}$  wie folgt verschoben:

$$P'(s_i) = \left( \frac{1}{2}(P(s_{i-1}) + P(s_{i+1})) - P(s_i) \right) w + P(s_i) \quad (4.5)$$

Der Wichtungsfaktor und die Anzahl der Iterationsschritte haben dieselben Auswirkungen wie bei der Oberflächenglättung. Allerdings ist zum einen zu beachten, dass ein Wichtungsfaktor von 1 einen leicht invertierenden Effekt auf den unregelmäßigen Verlauf hat. Zum anderen werden die jeweiligen Endpunkte einer Skelettlinie fixiert, so dass der Skelettlinienzug nicht als ganzer schrumpft. Als Beispiel zeigt Abb. 4.10 (b) die geglättete Skelettlinie von (a), wobei Gleichung 4.5 in drei Iterationsschritten mit einem Wichtungsfaktor von 0.7 angewendet wurde.



**Abb. 4.10:** Darstellung der resultierenden Skelettlinie der Portalvene der Leber aus deren binären Segmentierungsmasken nach SELLE U. A. [2000]. Die anschließende Laplace-Glättung mit Wichtungsfaktor  $w = 0.7$  erzeugt nach drei Glättungsdurchläufen einen gleichmäßigeren Richtungsverlauf.

Aufgrund der definierten Gefäßlängsrichtung durch die Skelettlinie lässt sich nun die radiale Vorzugsrichtung  $\hat{r}_v$  an jedem Eckpunkt  $v$  bestimmen. Mit Hilfe von Abb. 4.11 soll das Vorgehen dafür erläutert werden. In der Abbildung ist eine Skelettlinie  $C$  mit den Skelettvoxeln  $s_i \in S$  innerhalb eines Gefäßausschnittes dargestellt. Für jeden Skelettvoxel wird eine Ebene  $\epsilon(s_i)$  definiert, die durch  $s_i$  und dem Vektor  $\hat{t}_i$  aufgespannt ist. Der Vektor  $\hat{t}_i$  gibt die Tangentenrichtung an dem jeweiligen Skelettvoxel an. Bestimmt wird die Richtung durch zentrale Differenzenbildung aus seinem Skelettvorgängervoxel und -nachfolgervoxel, was der ersten Ableitung  $f'(s_i)$  an  $s_i$  entspricht:

$$\begin{aligned} f'(S_i) &= \frac{1}{2} (P(s_{i+1}) - P(s_{i-1})) \\ \hat{t}_i &= f'(s_i) / \|f'(s_i)\|. \end{aligned} \quad (4.6)$$

Nun wird zu jedem Eckpunkt  $v$  der Oberfläche die Ebene gesucht, zu der dieser den geringsten Abstand hat. Die Abstandsbestimmung eines Eckpunktes  $v$  zu einer Ebene erfolgt durch:

$$d_v(s_i) = |\hat{t}_i \cdot (v - s_i)| \quad (4.7)$$

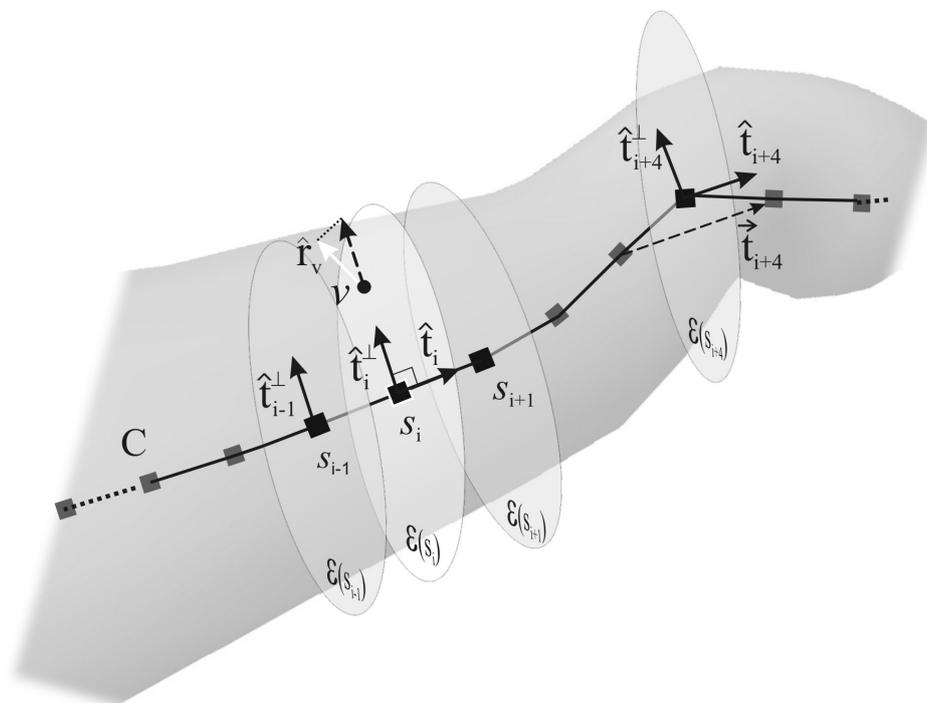
Der minimale Abstand  $d_{min}(s_i) = \min \{d_v(s_i) : s_i \in S\}$  korrespondiert somit zu der am nächsten gelegenen Ebene  $\epsilon^*(s_i)$ . Für diese wird ein Vektor  $\hat{t}_i^\perp$  definiert, der orthonormal zu  $\hat{t}_i$  steht und somit in der Ebene liegt. In AKENINE-MÖLLER UND HAINES [2002] wird ein numerisch stabiles Vorgehen präsentiert, das zu einem gegebenen Vektor  $\hat{t}$  einen solchen orthonormalen Vektor  $\hat{t}^\perp$  bestimmt:

$$\begin{aligned} \vec{t}^\perp &= \begin{cases} (0, -\hat{t}_z, \hat{t}_y) & , \text{ wenn } |\hat{t}_x| < |\hat{t}_y| \wedge |\hat{t}_x| < |\hat{t}_z| \\ (-\hat{t}_z, 0, \hat{t}_x) & , \text{ wenn } |\hat{t}_y| < |\hat{t}_x| \wedge |\hat{t}_y| < |\hat{t}_z| \\ (-\hat{t}_y, \hat{t}_x, 0) & , \text{ wenn } |\hat{t}_z| < |\hat{t}_x| \wedge |\hat{t}_z| < |\hat{t}_y| \end{cases} . \\ \hat{t}^\perp &= \vec{t}^\perp / \|\vec{t}^\perp\|. \end{aligned} \quad (4.8)$$

Dieser Vektor gibt die radiale Umlaufrichtung an dem Skelettvoxel  $s_i$  an. Um die gesuchte Vorzugsrichtung für den Eckpunkt  $v$  zu erhalten, erfolgt noch eine lineare Interpolation des Vektors  $\hat{t}^\perp$ . Dazu wird die benachbarte Ebene  $\epsilon(s_j)$  gesucht, welche mit  $\epsilon^*(s_i)$  den Eckpunkt  $v$  einschließt. Zu  $\epsilon(s_j)$  wird ebenfalls ein radialer Richtungsvektor  $\hat{t}_j^\perp$  und der minimale Abstand  $d_{min}(s_j)$  ermittelt. Aus den Abständen lässt sich ein Parameter  $u = 1 - \frac{d_{min}(s_i)}{(d_{min}(s_i) + d_{min}(s_j))}$  bestimmen, durch den sich die Vorzugsrichtung  $\hat{r}_v$  wie folgt ermitteln lässt:

$$\begin{aligned} \vec{r}_v &= u \hat{t}_i^\perp + (1 - u) \hat{t}_j^\perp \\ \hat{r}_v &= \vec{r}_v / \|\vec{r}_v\| \end{aligned} \quad (4.9)$$

Abschließend wird der Vektor noch in die Tangentialebene von  $v$  projiziert, die durch seine erste und zweite Hauptkrümmungsrichtung aufgespannt wird.



**Abb. 4.11:** Bestimmung der radialen Vorzugsrichtung  $\hat{r}_v$  für einen Eckpunkt  $v$ . Zu diesem wird die am nächsten gelegene Ebene  $\epsilon(s_i)$  gesucht, welche durch den Skelettvoxel  $s_i$  und dessen Tangentenrichtung  $\hat{t}_i$  definiert ist. Diese errechnet sich durch zentrale Differenzenbildung der benachbarten Skelettvoxel  $s_{i-1}$  und  $s_{i+1}$  (vgl. auch  $\hat{t}_{i+4}$ ). Zu  $\hat{t}_i$  wird ein orthonormaler Vektor  $\hat{t}_i^\perp$  ermittelt, der die radiale Umlaufrichtung an  $s_i$  angibt. Zwischen  $\hat{t}_i^\perp$  und  $\hat{t}_{i-1}^\perp$ , der zur einschließenden Ebene von  $v$  gehört, wird die Vorzugsrichtung  $\hat{r}_v$  linear interpoliert, abhängig vom Abstand  $v$  zu  $\epsilon(s_i)$ . Abschließend erfolgt noch eine Projektion von  $\hat{r}_v$  in die Tangentialebene von  $v$ .

Ein Aspekt, der noch berücksichtigt werden muss, ist der Spezialfall eines Gefäßbaumes  $G$ . Bei diesem treten Verzweigungen auf, wodurch nach der Methode von SELLE U. A. [2000] die Skelettlinie aus mehreren Kanten besteht, für jeden Gefäßzweig eine Kante. Letztere sollen im Folgenden als Kantenmenge  $E$  von  $G$  bezeichnet werden. Nach dem obigen Verfahren kann nun eine Ebene  $\epsilon(s_i)$  gefunden werden, wo der Skelettvoxel  $s_i$  nicht Teil der korrespondierenden Verzweigungskante ist, zu der  $v$  gehört. Der sich daraus ergebene Vektor  $\hat{r}_v$  könnte stark von der tatsächlichen, lokalen Vorzugsrichtung abweichen.

Als Kriterium für die Zugehörigkeit einer Gefäßkante  $e_i \in E$  zu einem Eckpunkt  $v$  wird festgelegt, dass es sich um die am nächsten gelegene Kante an  $v$  handelt. Bevor daher eine Abstandsbestimmung zu einer Ebene  $\epsilon(s_i)$  erfolgt, wird überprüft, ob diese zu  $v$  korrespondiert. Dafür werden zunächst alle die Kanten gefunden, die  $\epsilon(s_i)$  schneidet. Der Schnitttest erfolgt zunächst mit Hilfe der impliziten Ebenengleichung von  $\epsilon(s_i)$  und Parameterform einer Geraden  $L(t)$  (vgl. SCHNEIDER UND EBERLY [2003]):

$$\begin{aligned} \epsilon &: ax + by + cz + d = 0 \\ L(t) &= P + t\vec{s} \end{aligned} \quad (4.10)$$

Bezüglich der Ebene stellen  $[a \ b \ c]$  die Komponenten von  $\hat{t}_i$ ,  $[x \ y \ z]$  die Komponenten der Skelettvoxelposition  $s_i$  und  $d$  den minimalen Abstand der Ebene zum Ursprung dar. Des Weiteren ist die Gerade definiert durch einen Punkt  $P \in L$  und dem skalierten Richtungsvektor  $\vec{s}$  mit dem Parameter  $-\infty \leq t \leq \infty$ . Der Schnittpunkt  $Q$  zwischen einer Geraden und Ebene liegt einerseits auf der Geraden selbst und andererseits auch in der Ebene. Gesucht wird daher der Parameter  $t$ , der durch Substitution von  $L$  in  $\epsilon$  und anschließendem Umstellen, wie folgt bestimmt wird:

$$\begin{aligned} a(v_x + s_x t) + b(v_y + s_y t) + c(v_z + s_z t) + d &= 0 \\ t &= \frac{-(aP_x + bP_y + cP_z + d)}{ad_x + bd_y + cd_z} = \frac{-(\hat{n} \cdot P + d)}{\hat{n} \cdot \vec{s}} \end{aligned} \quad (4.11)$$

Da es sich bei den Kanten um Liniensegmente handelt, kann die Geradengleichung aus 4.10 zu  $L(t) = v_0 + t(v_1 - v_0)$  umgeschrieben werden, wobei  $v_0$  und  $v_1$  die Anfangs- und Endpunkte des Liniensegmentes bzw. der Gefäßkante sind. Der Schnittpunkt liegt nur dann auf dem Liniensegment, wenn  $0 \leq t \leq 1$  erfüllt ist. Somit muss das Ergebnis von Gleichung 4.11 anschließend gegen diesen Bereich getestet werden. Alle Kanten  $e_i$ , bei denen diese Bedingung erfüllt sind, werden zu einer Schnittkantenliste  $E' \in E$  zusammengefasst.

Wenn  $|E'| > 1$  ist, d. h. mehrere Kanten von  $\epsilon(s_i)$  geschnitten wurden, gilt es diejenige zu finden, die am nächsten zu  $v$  liegt. Der minimale Abstand  $d_v(e'_i)$  zwischen  $v$  und einer Kante  $e'_i$  ist dabei durch die Länge des Vektors  $\vec{q}$  bestimmt, der senkrecht zum Liniensegment steht und durch  $v - v'$  definiert ist. Der Punkt  $v'$  ist dabei die senkrechte Projektion von  $v$  auf  $e'_i$ . Formal lässt sich  $d_v(e'_i)$  wie folgt ermitteln:

$$\begin{aligned} d_v(e'_i) &= \|\vec{q}\| = \|v - v'\| = \|v - (v_0 + t\vec{s})\| \\ t &= \hat{s} \cdot (v - v_0) \end{aligned} \quad (4.12)$$

Für die durch  $d_{min} = \min \{d_v(e'_i) : e'_i \in E'\}$  gefundene Gefäßkante wird anschließend überprüft, ob  $s_i$  zu dieser Kante gehört. Trifft dies nicht zu, so wird  $\epsilon(s_i)$  verworfen und die nächste Ebene für  $v$  betrachtet. Erst wenn der Test erfolgreich ist, erfolgt die Abstandsbestimmung zu  $v$ .

## Muskelstrukturen

Bei Muskelstrukturen ist die Kontraktions- und Dehnungsrichtung zu finden, die der gesuchten Vorzugsrichtung der Schraffurlinien entspricht. Wie in den Vorbetrachtungen festgestellt wurde, hängt diese vom jeweiligen Klassifikationstyp der Lagebeziehung von Ursprung und Ansatz ab. Im Folgenden wird das Vorgehen beschrieben, wie sich bei gegebenem elongierten oder federförmigen Muskeltyp eine Approximation der Kontraktions- und Dehnungsrichtung automatisch aus den Daten extrahieren lässt. Ausgangsbasis dafür ist die geometrische Objektausdehnung des Muskels. Dazu wird eine *Oriented Bounding Box* (*OB*) des Objektes benutzt. Diese ist definiert durch drei orthonormalen Basisvektoren  $(\hat{u}, \hat{v}, \hat{w})$ , dem Objektmittelpunkt  $M$  und drei skalaren Werten  $h_u, h_v, h_w$ , die jeweils die Hälfte der Objektausdehnung in die entsprechende Richtung angeben (SCHNEIDER UND EBERLY [2003]). Die längste Achse entspricht somit der größten Objektausdehnung. Abstrakt betrachtet können die skalierten Basisvektoren als internes Skelett von Muskeln betrachtet werden, ähnlich dem obigen Skelettlinien bei Blutgefäßen. In AKENINE-MÖLLER UND HAINES [2002] wird eine Methode beschrieben, die eine *OB* mit Hilfe der Hauptachsentransformation in  $O(n \log n)$  bestimmt, wobei  $n$  für die Anzahl der Objektdreiecke steht.

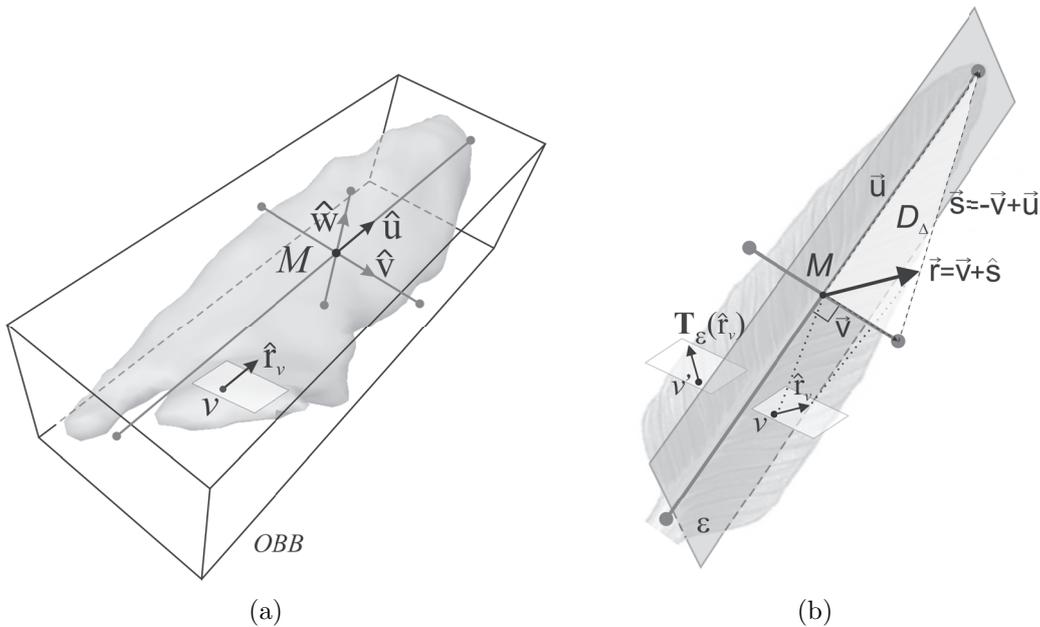
### Elongierter Muskeltyp

Für den elongierten Muskeltyp korrespondiert die Lagebeziehung von Ansatz und Ursprung mit der größten Objektausdehnung bzw. der längsten Hauptachse  $\hat{r} \in (\hat{u}, \hat{v}, \hat{w})$ . Demzufolge entspricht diese Richtung der Vorzugsrichtung für jeden Oberflächeneckpunkt. Für eine spätere Kombination mit den Krümmungsinformationen wird  $\hat{r}$  anschließend in die Tangentialebene von  $v$  projiziert. Dieses Vorgehen verdeutlicht Abb. 4.12 (a) am Beispiel einer Halsmuskeloberfläche. Zu sehen ist die *OB* des Objektes und dessen eingezeichnete Hauptachsen, die die maximalen Objektausdehnungen beschreiben. Als längste Achse wurde  $\hat{u}$  hervorgehoben und als  $\hat{r}_v$  in die angedeutete Tangentialebene von  $v$  projiziert.

### Federförmiger Muskeltyp

Charakteristisch für den federförmigen Muskeltyp ist die schräge Ausbreitung der Muskelfasern bzw. Schraffurlinien. Das begründet sich durch eine dreiecksartige Anordnung von Muskelursprung und -ansatz. Beschrieben werden kann diese Lagebeziehung ebenfalls durch die Achsen einer *OB* und einem sich dadurch aufspannenden, rechtwinkligen Dreieck. Anhand von Abb. 4.12 (b) soll dies näher erläutert werden. In der Zeichnung ist der oben erwähnte *Musculus bibennatus* angedeutet, auf dem die zwei größten Hauptachsen  $\vec{u} = h_u \hat{u}$  und  $\vec{v} = h_v \hat{v}$  abgetragen sind. Diese beschreiben die charakteristische

Objektausdehnung bezüglich der Länge und dem Verhältnis von Breite zu Tiefe. Zusätzlich definieren  $M$  und  $\hat{v}$  eine Ebene  $\epsilon$ , durch die eine spätere Orientierung der Vorzugsrichtung festgelegt wird (siehe unten). Mit Hilfe von  $\vec{u}$  und  $\vec{v}$  wird ein Vektor  $\vec{s} = -\vec{v} + \vec{u}$  erzeugt, durch den sich ein rechtwinkliges Dreieck  $D_\Delta$  ergibt. Anschließend lässt sich die Vorzugsrichtung  $\hat{r}$  durch Normalisierung von  $\vec{r} = \vec{v} + \hat{s}$  ermitteln.



**Abb. 4.12:** Approximation der Vorzugsrichtung  $\hat{r}_v$  für einen Oberflächenpunkt  $v$  bei Muskelstrukturen anhand ihrer *Oriented Bounding Box* (*OBB*). Für elongierte Muskeltypen, wie in (a), entspricht  $\hat{r}_v$  der normierten, längsten Hauptachse ( $\vec{u}$ ) der *OBB*. Bei federförmigen Strukturen, wie in (b), wird durch die beiden längsten Hauptachsen  $\vec{u}, \vec{v}$  der *OBB* ein Dreieck  $D_\Delta$  mit dessen Hypotenusenvektor  $\vec{s}$  beschrieben. Die Vorzugsrichtung für  $v$  entspricht dem normierten Vektor aus  $\vec{v} + \hat{s}$ . Eine anschließende Spiegelung  $\mathbf{T}_\epsilon$  an der Ebene  $\epsilon$ , ist notwendig, wenn  $v$  auf der negativen Seite von  $v'$  liegt. Abschließend erfolgt in (a) und (b) die Projektion von  $\hat{r}_v$  in die jeweilige Tangentialebene von  $v$ .

Wie aus Abb. 2.13 von Seite 20 erkennbar ist, kann der federförmige Muskeltyp einfach oder zweifach gefiedert sein. Je nachdem, welche Variante vorliegt, muss gegebenenfalls eine Spiegelung der Vorzugsrichtung an der Ebene  $\epsilon$  stattfinden. Die Orientierung von  $\hat{r}_v$  bei einem Eckpunkt  $v$  ist somit abhängig von dessen Lage bezüglich der Ebene bzw. Sehnenursprunges. Für jeden Eckpunkt muss entschieden werden, auf welcher Seite er von  $\epsilon$  liegt. Mit Hilfe einer vorzeichenbehafteten Distanz  $\pm d_v$  von  $v$  zu  $\epsilon$  wird dessen Lagebeziehung ermittelt (KIRK [1992]). Diese sei gegeben durch

$$\pm d_v = \hat{n} \cdot v - d_\epsilon, \quad (4.13)$$

wobei  $d_\epsilon = \hat{n} \cdot M$  der Abstand von  $\epsilon$  zum Koordinatenursprung ist. Ist  $d_v$  dabei positiv, so liegt  $v$  auf der Seite der Ebene, in die auch deren Normale zeigt. Anderenfalls liegt

$v$  auf der gegenüberliegenden Seite. Für die Orientierung der Vorzugsrichtung  $\hat{r}_v$  ergibt sich daher folgende Abhängigkeit:

$$\hat{r}_v = \begin{cases} \hat{r}_v & , d_v \geq 0 \\ \mathbf{T}_\epsilon(\hat{r}_v) & , d_v < 0 \end{cases} . \quad (4.14)$$

Die Transformation  $\mathbf{T}_\epsilon$  beschreibt dabei die Spiegelung an der Ebene  $\epsilon$  und ist definiert durch die 4x4 Matrix

$$\mathbf{T}_\epsilon = \begin{bmatrix} [\mathbf{I} - 2((\hat{r} \cdot \hat{v}_\perp)\hat{v}_\perp)] & \mathbf{0}^T \\ 2(M \cdot \hat{v})\hat{v} & 1 \end{bmatrix} . \quad (4.15)$$

Abschließend wird  $\hat{r}_v$  noch in die Tangentialebene von  $v$  projiziert. In der Abb. 4.12 (b) ist dazu ein Beispiel gegeben. Für den Eckpunkt  $v$  wird einfach die normierte Vorzugsrichtung in die Tangentialebene projiziert, da sich dieser auf der „positiven“ Seite der Ebene befindet. Die Vorzugsrichtung für den Eckpunkt  $v'$  wird vor der Projektion an der Ebene gespiegelt, da dieser auf der „negativen“ Ebenenseite liegt.

## Organe und Knochenstrukturen

Bei Organen und Knochenstrukturen wird für diese Arbeit ein Kompromiss bezüglich einer *Frame*-kohärenten Schraffierung und der korrekten Beschreibung des Oberflächenverlaufes gesucht. Angelehnt an die Untersuchungen der anatomischen Illustrationen, den existierenden Schraffierungsansätzen sowie der betrachteten Krümmungsapproximation aus Abschnitt 3.1.2 wird für Organe und Knochenstrukturen die erste Hauptkrümmungsrichtung als charakteristische Vorzugsrichtung gewählt. Da diese der stärksten Krümmungsrichtung entspricht, werden damit markante Oberflächenverläufe wie konkave oder konvexe Bereiche beschrieben.

Diese Richtungen werden für alle Eckpunkte automatisch aus den Geometriedaten durch den *BezierQuadOP*-Operator ermittelt. Eine spätere Zusammenführung mit weiteren Krümmungsinformationen ist somit nicht nötig, so dass hier bereits für Organe und Knochen das initiale Vektorfeld definiert wird. Mit Hilfe des gewählten Operators wird auch sichergestellt das durch diesen die Richtungsinformationen unter den hier zur Verfügung stehenden Approximationsmethoden am genauesten ermittelt werden.

Aufgrund rauer Oberflächenbereiche kann es jedoch zu lokalen Abweichungen in der ersten Hauptkrümmungsrichtung kommen, die den geodätischen Richtungsverlauf stören. Daher erfolgt eine zusätzliche Glättung des Vektorfeldes über eine  $k$ -Nachbarschaft, wobei „ $k$ “ den topologischen Radius um einen Eckpunkt angibt. Die Glättung geschieht dabei analog zur Gleichung 4.4 für die Interpolation fehlender Krümmungsrichtungen aus der direkten Nachbarschaft. Der Unterschied besteht zum einen darin, dass nur über die erste Hauptkrümmungsrichtung aus der Nachbarschaft gemittelt wird und zum anderen dass der topologische Radius nicht nur auf die direkten Nachbarn beschränkt ist. Die Wahl von „ $k$ “ kann vom Benutzer eingestellt werden.

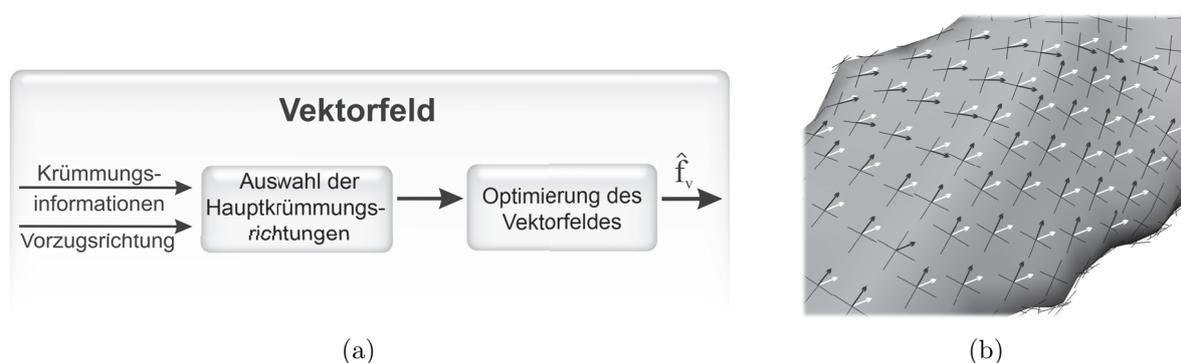
Dieses Vorgehen stellt somit einen Kompromiss dar zwischen der korrekten Beschreibung des Oberflächenverlaufes, ähnlich zur Abb. 4.3 (c) von Seite 62 und der *Frame*-Kohärenz bei einer interaktiven Visualisierung. Das einmal so erzeugte Vektorfeld ändert seinen Verlauf während einer Interaktion durch den Benutzer nicht. Dadurch ändert sich auch nicht die Parametrisierung der Oberfläche bzw. die daraus resultierende Texturierung.

### 4.3.5 Kombination von Krümmungsinformationen und Vorzugsrichtung

Der folgende Abschnitt bezieht sich nur auf Gefäß- und Muskelstrukturen, da für Organe und Knochen Vorzugsrichtung und Krümmungsinformationen gleich sind. Eine Kombination ist somit nicht nötig.

Für jeden Eckpunkt  $v$  der Oberfläche eines Gefäßes oder Muskels existieren nun die beiden Hauptkrümmungsrichtungen anhand von Abschnitt 4.3.3 und eine modellspezifische Vorzugsrichtung entsprechend dem vorhergehenden Abschnitt. Diese Informationen werden im Folgenden zu einem Richtungsvektor  $\hat{f}_v$  zusammengeführt, welcher mit allen anderen Richtungsvektoren das für die Parametrisierung notwendige Vektorfeld ergibt. Dazu sind zwei Teilaufgaben erforderlich, die in Abb. 4.13 (a) dargestellt sind. Zum einen ist die Auswahl der Hauptkrümmungsrichtung erforderlich, die der Vorzugsrichtung am ähnlichsten ist. Dies begründet sich durch eine visuelle Beurteilung der beiden Richtungen bezüglich dem zu erwartenden Verlauf. Anhand von Abb. 4.13 (b) ist erkennbar, dass jeweils eine der Hauptkrümmungsrichtungen oder deren entgegengesetzten Richtung der Vorzugsrichtung am nächsten kommt. Letztere ist durch einen weißen Pfeil gekennzeichnet. Zum anderen ist aus der Abbildung ersichtlich, dass das initiale Vektorfeld anschließend optimiert werden muss, um den geodätischen Verlauf zu verbessern.

In diesem Zusammenhang soll noch ein Gedanke diskutiert werden, der während der Bestimmung der Vorzugsrichtung bei Gefäß- und Muskelstrukturen aufgekommen ist. Da beispielsweise bei Gefäßen die ideale Vorzugsrichtung für einen Eckpunkt  $v$  auf einer Gefäßoberfläche dem radialen Vektor  $\hat{r}_v$  entspricht, wären dafür keine Krümmungsinformationen notwendig. Dieser lässt sich allein aus dem Skelettverlauf des Gefäßes ermitteln. Werden für alle Oberflächenpunkte allerdings ausschließlich deren Vorzugsrichtung verwendet, so können wichtige Details im Oberflächenverlauf des Vektorfeldes fehlen. Dies ist beispielsweise bei der Beurteilung von krankhaften Gefäßweiterungen (*Aneurysmen*) oder -verengungen (*Stenosen*) ausschlaggebend. Hier ist eine korrekte Vermittlung des Oberflächenverlaufes durch das Vektorfeld bzw. die daraus später resultierenden Schraffurlinien wichtig. Der Skelettlinienverlauf spiegelt diese Oberflächenunebenheiten nur unzureichend bis gar nicht wider. Dafür sind wiederum Krümmungsinformationen notwendig. Somit ist es sinnvoll, beide Informationen zu ermitteln und in geeigneter Weise zu kombinieren. Damit wird für den Betrachter, die in diesem Kapitel erwähnte korrekte Beschreibung des Oberflächenverlaufes sichergestellt .



**Abb. 4.13:** Kombination von Hauptkrümmungsrichtung und Vorzugsrichtung zur Generierung des für die Parametrisierung notwendigen Vektorfeldes. (a) Initialisierung des Vektorfeldes durch Auswahl der Hauptkrümmungsrichtung, die der Vorzugsrichtung am ähnlichsten ist. Veranschaulicht ist dies in (b) durch die weiß dargestellte Vorzugsrichtung  $\hat{r}_v$  an jedem Eckpunkt und Hervorhebung der ähnlichsten Hauptkrümmungsrichtung. Anschließend erfolgt eine Optimierung des initialen Vektorfeldes, welches am Ende für jeden Eckpunkt einen Richtungsvektor  $\hat{f}_v$  besitzt.

### Auswahl der Krümmungsrichtung

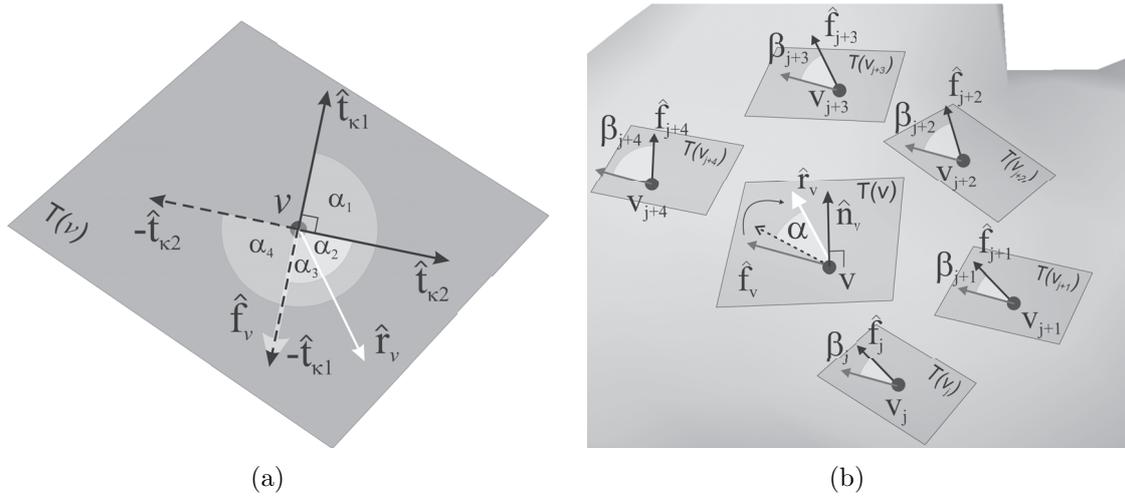
Die Abb. 4.14 (a) zeigt zunächst die Tangentialebene  $T(v)$  von einem Eckpunkt  $v$ , in der dessen Hauptkrümmungsrichtungen  $\hat{t}_{\kappa 1}$  und  $\hat{t}_{\kappa 2}$  sowie Vorzugsrichtung  $\hat{r}_v$  eingezeichnet sind. Des Weiteren sind die entgegengesetzten Richtungen  $-\hat{t}_{\kappa 1}$ ,  $-\hat{t}_{\kappa 2}$  der beiden Hauptkrümmungsrichtungen angedeutet. Nun wird für jede der vier möglichen Richtungen deren Winkel  $\alpha_i$  mit  $i : dir_i \in \{1 = \hat{t}_{\kappa 1}, 2 = \hat{t}_{\kappa 2}, 3 = -\hat{t}_{\kappa 1}, 4 = -\hat{t}_{\kappa 2}\}$  zu der Vorzugsrichtung bestimmt. Berechnet wird dieser über das Skalarprodukt und anschließender Umwandlung des Bogenmaßes in Grad:

$$\begin{aligned} \cos \alpha_i &= \hat{r}_v \cdot \vec{dir}_i = rad_i \\ \alpha_i &= \frac{\arccos(rad_i) 180.0}{\pi} \end{aligned} \quad (4.16)$$

Aufgrund der normalisierten Vektoren ist eine Division durch den Betrag der Vektoren nicht notwendig. Anschließend wird der Vektor als initiale Richtung  $\hat{f}_v$  für das Vektorfeld ausgewählt, der den geringsten Winkel zu  $\hat{r}_v$  besitzt. In der gezeigten Abbildung entspricht dies der entgegengesetzten ersten Hauptkrümmungsrichtung  $-\hat{t}_{\kappa 1}$ .

### Optimierung des Vektorfeldes

Das initiale Richtungsfeld weist an manchen Stellen noch keinen geodätischen Verlauf auf, wie dies auch anhand von Abb. 4.13 (b) erkennbar ist. Einzelne Vektorrichtungen



**Abb. 4.14:** (a) Bestimmung des initialen Richtungsvektors  $\hat{f}_v$  durch Auswahl der ähnlichsten Hauptkrümmungsrichtung zur Vorzugsrichtung  $\hat{r}_v$ . Ähnlichkeitsmaß ist definiert durch den kleinsten Winkel von  $\alpha_1, \alpha_2, \alpha_3$  oder  $\alpha_4$ , welche zwischen  $\hat{r}_v$  und den beiden Hauptkrümmungsrichtungen sowie ihrer entgegengesetzten Richtungen ermittelt werden. Im Beispiel entspricht dies der Richtung  $-\hat{t}_{\kappa 1}$ . (b) Optimierung des initialen Vektorfeldes durch Vergleich der mittleren Winkelabweichung von  $\hat{f}_v$  zu seinen direkten Nachbarrichtungsvektoren nach Rotation um  $\hat{n}_v$  in Richtung  $\hat{r}_v$ .

entsprechen nur sehr grob der eigentlichen Vorzugsrichtung oder besitzen teilweise eine orthogonalen Orientierung zu benachbarten Vektoren. Somit ist eine Optimierung notwendig, die einerseits die Vektoren mehr an der Vorzugsrichtung ausrichtet und andererseits die starken Abweichungen in der lokalen Nachbarschaft glättet. Letzteres soll allerdings markante Oberflächenverläufe nicht völlig ausgleichen.

Die Optimierung wird als ein Energieminimierungsproblem  $E$  betrachtet. Dieses besteht einerseits aus der mittleren Winkelabweichung zwischen dem Richtungsvektor  $\hat{f}_v$  mit seinen direkt benachbarten Richtungsvektoren  $\hat{f}_j \in nhd(v)$  und andererseits zwischen der Winkeldifferenz von  $\hat{f}_v$  und der Vorzugsrichtung  $\hat{r}_v$ :

$$\min E = \frac{1}{n} \sum_{j=0}^n \angle(\hat{f}_v, \hat{f}_j) + \angle(\hat{f}_v, \hat{r}_v) \quad (4.17)$$

Die Winkelbestimmungen werden analog zu Gleichung 4.16 durchgeführt. In Abb. 4.14 (b) wird der Optimierungsschritt näher erläutert werden. Dargestellt ist ein vergrößerter Oberflächenbereich um einen Eckpunkt  $v$  und dessen direkte Nachbareckpunkte  $v_j \in nhd(v)$ . Zusätzlich ist zu jedem Eckpunkt seine Tangentialebene  $T(v)$  sowie sein initialer Richtungsvektor  $\hat{f}_v$  angegeben. Für die mittlere Winkelabweichung wird zunächst der Vektor  $\hat{f}_v$  in die jeweilige Tangentialebene seiner Nachbareckpunkte projiziert und die Winkel  $\beta_j$  zu  $\hat{f}_j$  bestimmt. Danach wird der Vektor  $\hat{f}_v$  in seiner Tangentialebene um  $1^\circ$

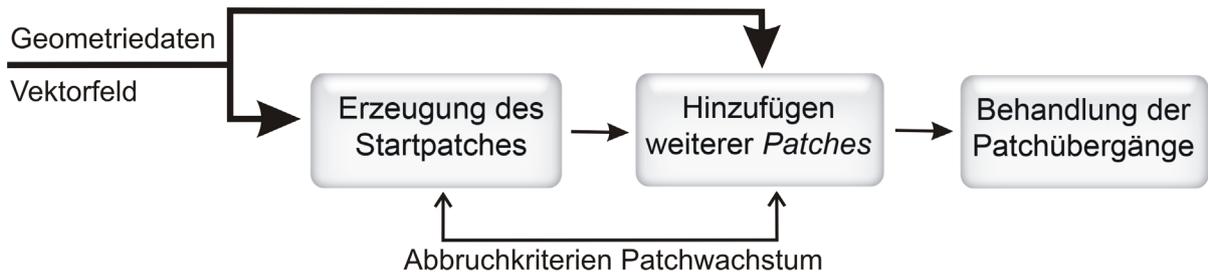
in Richtung  $\hat{r}_v$  um  $\hat{n}_{T(v)}$  rotiert und anschließend anhand des neu ausgerichteten Vektors die neue mittlere Winkelabweichung zu seinen Nachbarn bestimmt. Die Rotation ist nur dann gültig und wird weiter fortgeführt, wenn die neue Winkelabweichung sowie der Winkel  $\alpha$ , zwischen  $\hat{f}_v$  und  $\hat{r}_v$ , kleiner geworden sind. Ist dies nicht der Fall, so wird die Rotation zurückgenommen und die Optimierung für den Eckpunkt abgebrochen. Das beschriebene Vorgehen wird auf jeden Eckpunkt angewendet und anschließend rekursiv so lange wiederholt bis keine Optimierung mehr eintritt oder eine bestimmte Rekursionstiefe erreicht ist.

Für die anschließende Parametrisierung ist ein Richtungsvektor pro Dreieck notwendig. Daher werden die drei vorhandenen Vektoren an den Eckpunkten eines Dreiecks gemittelt und diesem zugewiesen.

## 4.4 Parametrisierung der Oberfläche

Dieser Abschnitt beschreibt die Oberflächenparametrisierung der anatomischen 3D-Modelle. Das Ziel dabei ist die Erzeugung von Parameterkoordinaten  $(u, v)$  für alle Oberflächeneckpunkte, so dass später eine möglichst verzerrungsfreie (isomorphe) Abbildung der Schraffierungstextur auf dem Objekt erfolgt. Aufgrund der bestehenden Charakteristik der Oberflächenmodelle, wie starke Unregelmäßigkeit oder Vorkommen von Löchern in den Dreiecksnetzen sowie der Notwendigkeit die Texturen lokal auszurichten, wird im Folgenden der direkte Parametrisierungsansatz von PRAUN U. A. [2000] aufgegriffen. Bei diesem erfolgt eine Partitionierung aller Oberflächendreiecke in so genannte *Lapped Patches*, welche einzeln und unabhängig voneinander parametrisiert sowie texturiert werden (vgl. Abschnitt 2.4.2). Der Nachteil dieser Art von Parametrisierung liegt im Entstehen von Texturnähten an den Patchübergängen (vgl. Abschnitt 4.4.4), welche kompensiert werden müssen. PRAUN U. A. [2000] löst dies durch Texturüberblendung der einzelnen *Patches*.

Die folgende Parametrisierung erzeugt ebenfalls überlappende *Patches*, modifiziert aber deren Übergänge. Dabei werden Texturnähte weitestgehend vermieden, so dass eine Texturüberblendung nicht mehr nötig ist. Die dafür erforderlichen Schritte sind in der Abb. 4.15 zusammengefasst. Das gesamte Patchwachstum basiert zunächst auf den Geometriedaten der Oberfläche und dem darauf erzeugten Vektorfeld. Zu Beginn wird ein initiales Startpatch erzeugt und deren Dreiecke in den Parameterraum überführt. Um das Startpatch werden nun weitere *Patches* hinzugefügt und ebenfalls parametrisiert, bis die gesamte Oberfläche vollständig überdeckt ist. Die Ausbreitung jedes Patchwachstums wird dabei durch verschiedene Abbruchkriterien eingeschränkt. Am Schluß erfolgt die Behandlung der Patchübergänge zur Kompensation der Texturnähte. Jeder dieser Schritte wird nun im Einzelnen genauer erläutert.



**Abb. 4.15:** *Workflow* für die Oberflächenparametrisierung durch Generierung von *Patches*. Anhand der Geometriedaten und des Vektorfeldes erfolgt zunächst die Erzeugung und Ausrichtung eines Startpatch, um welches sich anschließend weitere *Patches* anfügen. Das Patchwachstum wird dabei durch verschiedene Abbruchkriterien kontrolliert. Abschließend erfolgt die Behandlung von Texturnähten an sich überlappenden Patchübergängen.

#### 4.4.1 Erzeugung des Startpatch und Überführung in den Parameterraum

Die Erzeugung und Parametrisierung des Startpatch gliedert sich in drei Schritte. Zunächst muss das Startpatch initialisiert und in den Parameterraum überführt werden, anschließend beginnt das Wachstum und endet mit der Bestimmung der Parameterkoordinaten. Zur Veranschaulichung der einzelnen Schritte dient die Abb. 4.16.

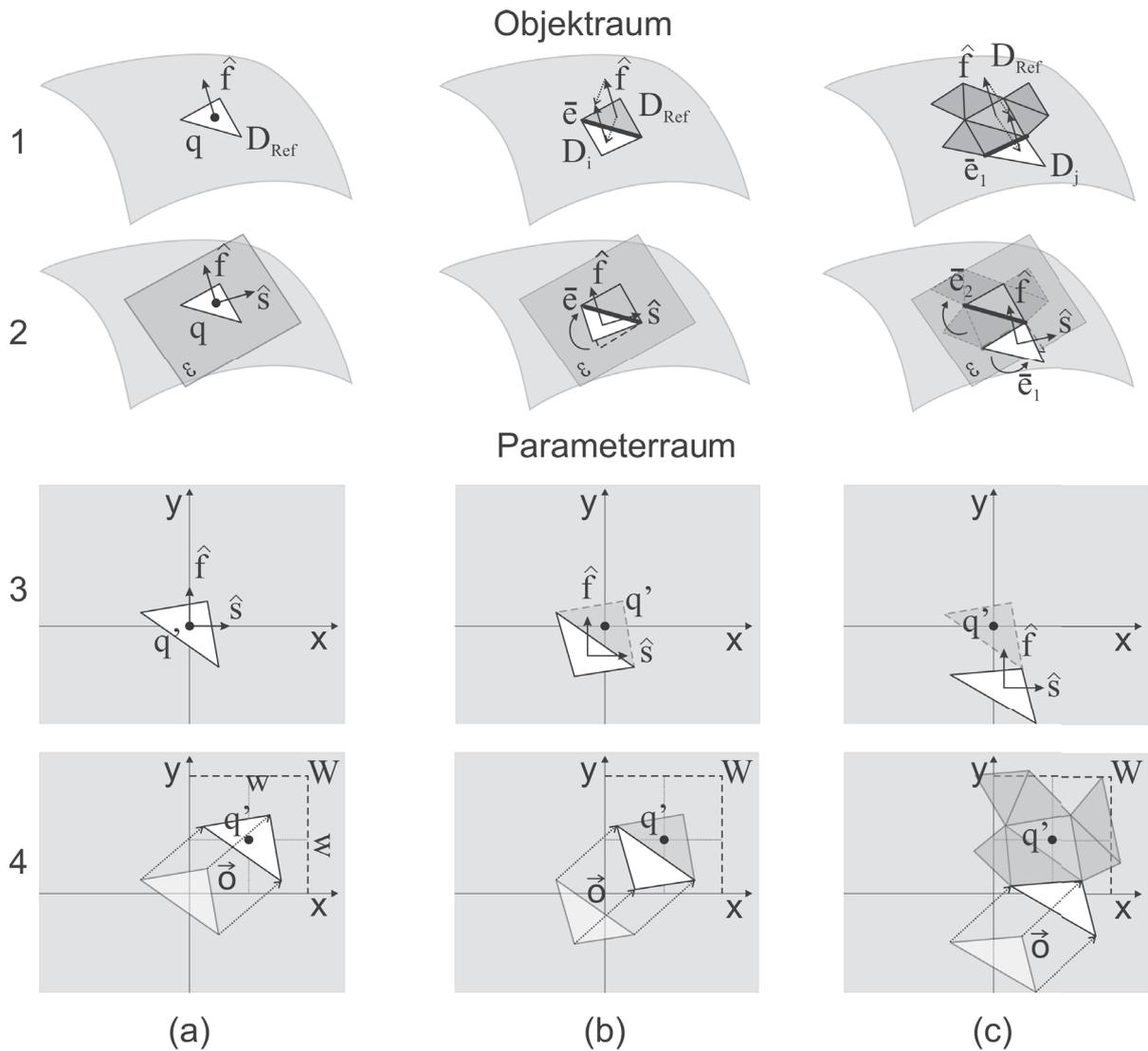
##### Initialisierung des Startpatch

Die Initialisierung des ersten *Patch* beginnt mit der Wahl eines beliebigen Dreieckes der Oberfläche. Ein solches Dreieck wird hier und im Weiteren für die Erzeugung neuer *Patches* als Referenzdreieck  $D_{Ref}$  bezeichnet. Für das Referenzdreieck wird eine Ebene  $\epsilon$  durch seinen Mittelpunkt  $P$  und seiner Normalen  $\hat{n}$  aufgespannt. Aus dem Kreuzprodukt der Normalen mit dem Richtungsvektor  $\hat{f}$  von  $D_{Ref}$ , lässt sich der zu beiden orthogonal stehende Vektor  $\hat{s}$  erzeugen (vgl. Abb. 4.16 (a-1, -2)). Diese drei Vektoren bilden eine lokale Basis, welches zusammen mit dem Referenzdreieck in den Parameterraum überführt wird. Der Parameterraum entspricht dabei der  $x$ - $y$ -Ebene im Weltkoordinatensystem. Die Abbildung  $\mathbf{X}$  in die Ebene beschreibt zuerst eine Translation  $\mathbf{T}(-P)$  des Referenzdreieckes in den Koordinatenursprung anhand seines Mittelpunktes:

$$\mathbf{T}(-P) = \begin{bmatrix} 1 & 0 & 0 & -v_x \\ 0 & 1 & 0 & -v_y \\ 0 & 0 & 1 & -v_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.18)$$

Anschließend erfolgt eine Rotation  $\mathbf{R}$  der Basisvektoren auf die Standardbasisvektoren ( $x$ -,  $y$ - und  $z$ - Achse) in der Weise, dass  $x \parallel \hat{s}$ ,  $y \parallel \hat{f}$  und  $z \parallel \hat{n}$  gilt (vgl. Abb. 4.16 (a-3)):

$$\mathbf{R} = \begin{bmatrix} \hat{s}_x & \hat{s}_y & \hat{s}_z & 0 \\ \hat{f}_x & \hat{f}_y & \hat{f}_z & 0 \\ \hat{n}_x & \hat{n}_y & \hat{n}_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.19)$$



**Abb. 4.16:** Erzeugung des Startpatches und Überführung in den Parameterraum. (a-1) Ein beliebiges Dreieck der Oberfläche dient als initiales Referenzdreieck  $D_{Ref}$  für das Startpatch. (a-2) Erzeugung einer lokalen Basis bestehend aus dem Richtungsvektor  $\hat{f}$ , Dreiecksnormale und zu beiden orthogonal stehendem Vektor  $\hat{s}$ .  $D_{Ref}$  spannt eine Referenzebene  $\epsilon$  auf. (a-3) Zunächst wird  $D_{Ref}$  anhand seines Mittelpunktes  $q$  in den Ursprung des Weltkoordinatensystems verschoben. Anschließend erfolgt eine Basistransformation in die  $x-y$ -Ebene als Parameterraum, so dass  $x \parallel \hat{s}, y \parallel \hat{f}$  und  $z \parallel \hat{n}$  gilt. (a-4) Abschließend wird  $D_{Ref}$  um Vektor  $\vec{o} = w/2$  in das Projektionsfenster  $W$  verschoben. (b-1) Patchwachstum durch Hinzunahme eines inzidenten, noch nicht parametrisierten Dreiecks  $D_i$  an der Kante  $\bar{e}$ . (b-2) Übertragung der Basis aus (a-2) auf  $D_i$  und anschließende Rotation um  $\bar{e}$  nach  $\epsilon$ . (b-3), (b-4) Transformation von  $D_i$  in den Parameterraum relativ zu  $D_{Ref}$ . Der Verschiebungsvektor  $\vec{o}$  entspricht dabei dem Abstand des bereits transformierten Mittelpunktes  $q'$  von  $D_{Ref}$ . (c-1) Ein weiteres Dreieck  $D_j$  wird an Kante  $\bar{e}_1$  nach einem fortgeschrittenen Patchwachstum hinzugefügt. (c-2) Die Rotation in  $\epsilon$  geschieht zuerst an  $\bar{e}_1$  und anschließend an  $\bar{e}_2$ . (c-3), (c-4) Die Transformation in den Parameterraum erfolgt wie beim Dreieck  $D_i$  relativ zum Referenzdreieck. In (c-4) sind zusätzlich die bisherigen transformierten Patchdreiecke eingezeichnet.

Nun befindet sich das Referenzdreieck vollständig in der  $x$ - $y$ -Ebene, wodurch sich alle weiteren Betrachtungen und Transformationen auf 2D beziehen. Die Basistransformation bewirkt dabei die Integration der Schraffurrichtung in die Parametrisierung.

In der Ebene wird ein Projektionsfenster  $W$  definiert, welches korrespondierend zu einer vollständigen Texturabbildung ist. Dieser Bereich besitzt die Größenausdehnung  $w$  und hat Einfluss auf das weitere Patchwachstum (vgl. Abbruchkriterium 4 in Abschnitt 4.4.2 auf Seite 91). Das ausgerichtete Referenzdreieck wird durch eine weitere Translation  $\mathbf{T}(\vec{\sigma})$  vom Koordinatenursprung in die Mitte von  $W$  verschoben (vgl. Abb. 4.16 (a-4)):

$$\mathbf{T}(\vec{\sigma}) = \begin{bmatrix} 1 & 0 & 0 & \vec{\sigma}_x \\ 0 & 1 & 0 & \vec{\sigma}_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.20)$$

Der Vektor  $\vec{\sigma}$  setzt sich aus dem bisher transformierten Mittelpunkt  $P'$  und der Größenausdehnung von  $W$  mit  $\vec{\sigma} = P' + \frac{w}{2}$  zusammen. Aus den einzelnen Transformationsschritten ergibt sich die endgültige Abbildung  $\mathbf{X}$  mit:

$$\mathbf{X} = \mathbf{T}(\vec{\sigma}) \mathbf{R} \mathbf{T}(-P). \quad (4.21)$$

### Patchwachstum

Für das weitere Patchwachstum werden die direkten und indirekten Dreiecke an  $D_{Ref}$  in die Ebene  $\epsilon$  rotiert. Anders als bei PRAUN U. A. [2000], wo diese projiziert werden, entstehen hier aufgrund der affinen Transformation keine Verzerrungen in der Geometrie und damit auch nicht in der Parametrisierung bzw. späteren Texturierung. Als hinzu zufügende Dreiecke werden stets solche betrachtet, die an den Außenkanten des aktuellen *Patch* liegen aber selbst noch nicht zu diesem gehören. Der Wachstumsprozess verläuft so lange bis die Außenkantenliste  $L_{Kontur}$  des *Patch* leer ist. Im Abschnitt 4.4.2 wird beschrieben, wie  $L_{Kontur}$  aktualisiert wird.

Beispielhaft wird nun in Abb. 4.16 (b) ein Dreieck  $D_i$  an  $D_{Ref}$  betrachtet, das in den Parameterraum überführt wird. Rotiert wird dieses über die Außenkante  $\bar{e}$  und erhält die Basis von  $D_{Ref}$  (vgl. (b-1), (b-2)). Die Rotation wird durch die Transformation  $\mathbf{M}$  beschrieben. Diese setzt sich aus zwei Translationen und drei Rotationen zusammen. Zunächst erfolgt eine Verschiebung der Rotationskante in den Ursprung mit  $\mathbf{T}(-\bar{e})$  über einen der Kantenendpunkte:

$$\mathbf{T}(-\bar{e}) = \begin{bmatrix} 1 & 0 & 0 & -\bar{e}_x \\ 0 & 1 & 0 & -\bar{e}_y \\ 0 & 0 & 1 & -\bar{e}_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.22)$$

Darauf folgt eine Basistransformation, wie sie in Gleichung 4.19 beschrieben ist. Da zu der Kante  $\bar{e}$  noch keine Basis  $B_{\bar{e}}$  existiert, muss diese erst erzeugt werden, d. h. die Bestimmung von drei orthonormalen Vektoren. Der erste Vektor ist  $\hat{e}$  selbst und ein zweiter Vektor  $\hat{s}$  errechnet sich aus Gleichung 4.8. Der dritte Basisvektor  $\hat{t}$  ergibt sich aus dem Kreuzprodukt von  $\hat{e}$  und  $\hat{s}$ . Anschließend erfolgt die Rotation  $\mathbf{R}(B_{\bar{e}})$  der Basis auf die Standardbasis, so dass  $\hat{e} \parallel \hat{x}$  gilt:

$$\mathbf{R}(B_{\bar{e}}) = \begin{bmatrix} \hat{e} & 0 \\ \hat{s} & 0 \\ \hat{r} & 0 \\ \hat{0} & 1 \end{bmatrix}. \quad (4.23)$$

Nun wird die eigentliche Rotation des Dreiecks ausgeführt. Dies entspricht einer Rotation  $\mathbf{R}_x(\phi)$  um die  $x$ -Achse mit einem Winkel  $\phi$  (in Bogenmaß), der dem Winkel zwischen den Normalen von  $D_i$  und  $D_{Ref}$  entspricht:

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.24)$$

Abschließend erfolgt die Rückrotation der Basis und Rückverschiebung der Rotationskante. Aufgrund der orthogonalen Matrix  $\mathbf{R}(B_{\bar{e}})$  entspricht deren Inverse, ihrer Transponierten. Zusammenfassend lässt sich somit die Transformation  $\mathbf{M}$  wie folgt beschreiben:

$$\mathbf{M} = \mathbf{T}(\bar{e}) \mathbf{R}^T(B_{\bar{e}}) \mathbf{R}_x(\phi) \mathbf{R}(B_{\bar{e}}) \mathbf{T}(-\bar{e}). \quad (4.25)$$

Kante und Rotationswinkel werden für eine spätere Verwendung zu  $D_i$ , wie auch bei allen weiteren Patchdreiecken, abgespeichert. Die anschließende Transformierung von  $D_i$  in den Parameterraum geschieht anhand der Abbildung  $\mathbf{X}$ , die für das Referenzdreieck aufgestellt wurde. Somit erfolgt für  $D_i$  die Verschiebung in den Koordinatenursprung, die Ausrichtung in die  $x$ - $y$ -Ebene und anschließender Verschiebung in das Projektionsfenster  $W$  relativ zu  $D_{Ref}$ . In Abb. 4.16 (b-3) und (b-4) sind diese Schritte noch einmal veranschaulicht, wobei zusätzlich die Lage des transformierten Referenzdreiecks mit eingezeichnet ist.

Betrachtet wird noch ein weiteres Dreieck  $D_j$ , das direkt mit  $D_i$  über eine Außerkante verbunden ist und ebenfalls in die Ebene  $\epsilon$  rotiert werden soll (vgl. Abb. 4.16 (c-1)). Das Dreieck erhält ebenfalls wieder die Basis vom Referenzdreieck. Über die gemeinsame Kante  $\bar{e}_1$  zum Dreieck  $D_i$ , wird  $D_j$  in die Ebene von  $D_i$  anhand von Gleichung 4.25 rotiert. Wie oben erwähnt wurde, sind zu  $D_i$  der eigene Rotationswinkel und die Rotationskante zum vorhergehenden Dreieck, in dem Fall zum Referenzdreieck, abgespeichert. Somit erfolgt mit diesen Informationen eine weitere Rotation um  $\bar{e}_2$  in die Ebene  $\epsilon$  von  $D_{Ref}$  (vgl. Abb. 4.16 (c-2)). Dieses Vorgehen der fortschreitenden Rotationen (Faltung) wird auf alle

dem *Patch* hinzu zu fügenden Dreiecke angewendet. Die anschließende Transformation in den Parameterraum erfolgt in der gleichen Weise wie bei  $D_i$ , d. h. relativ zu  $D_{Ref}$  (vgl. Abb. 4.16 (c-3,-4)). In der Abb. 4.16 (c-4) sind zusätzlich weitere Dreiecke abgebildet, um die relative Abbildung in den Parameterraum bezüglich des Referenzdreieckes zu verdeutlichen.

Für die praktische Umsetzung wird für die Rotation eines Dreieckes in die Ebene  $\epsilon$ , mit anschließender Transformation in den Parameterraum, nur der Eckpunkt betrachtet, der nicht Teil der ersten Rotationskante ist. Die Position der beiden anderen Eckpunkte werden einfach von dem vorhergehenden, bereits transformierten Dreieck übernommen.

### Bestimmung der Parameterkoordinaten

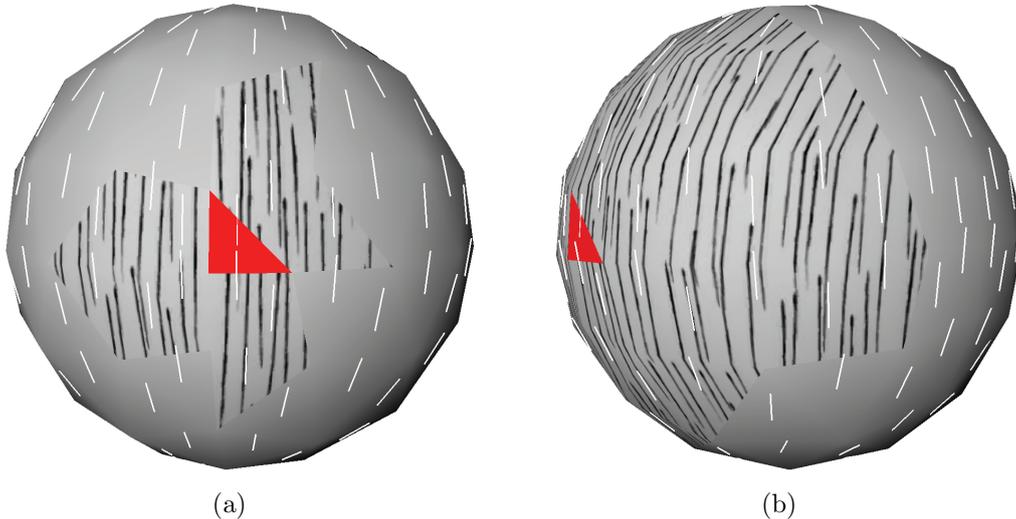
Die Ermittlung der Parameterkoordinaten  $(u, v)$  für die Eckpunkte der transformierten Dreiecke gestaltet sich recht einfach. Nach Anwendung von  $\mathbf{M}$  und  $\mathbf{X}$  existiert für einen Eckpunkt  $v$  ein entsprechendes Koordinatenpaar  $(x', y', 0)$  in der  $x$ - $y$ -Ebene. Da es sich bei der Ebene um den Parameterraum handelt, entsprechen die Parameterkoordinaten von  $v$  somit seiner Ebenenposition, d. h.  $(u = x', v = y')$ . Die Werte werden anschließend noch durch eine Division mit der Projektionsfenstergröße  $w$  und einem Skalierungsfaktor  $s$  auf ein Vielfaches der Texturgröße skaliert:  $(u, v) = (u/ws, v/ws)$ . Eine Umrechnung in ganzzahlige Texturkoordinaten erfolgt später auf der Graphikkarte (vgl. Abschnitt 2.4.1). Eine vorauslaufende Modulooperation auf den  $(u, v)$ -Koordinaten bewirkt dabei die Überführung auf das Intervall  $[0, 1]$ .

#### 4.4.2 Abbruchkriterien für ein Patchwachstum

Das im vorigen Abschnitt beschriebene Verfahren lässt ein *Patch* um ein initiales Referenzdreieck  $D_{Ref}$  wachsen. Erwähnt wurde dabei, dass dieser Wachstumsprozess so lange erfolgt, bis die Aussenkantenliste  $L_{Kontur}$  des *Patch* leer ist. Die Bestimmung von  $L_{Kontur}$  erfolgt anhand der Untersuchung aller Kanten der bisherigen Patchdreiecke. Jede Kante, die dabei nicht als Abbruchkante markiert ist, wird der Liste hinzugefügt. Die Aktualisierung von  $L_{Kontur}$  erfolgt nach jedem dem *Patch* hinzugefügten Dreieck. Im Folgenden sollen verschiedene Kriterien vorgestellt werden, die einen Abbruch des Patchwachstums über eine solche Kante hinweg erzwingen, diese also als Abbruchkante markieren. Berücksichtigt wird dabei hauptsächlich das Vermeiden von starken Verzerrungen, trotz affiner Transformationen, und der sich variierende Richtungsverlauf des Vektorfeldes in der Nachbarschaft von  $D_{Ref}$ .

### Kriterium 1: Diskontinuitäten im Vektorfeld

Die Ausrichtung der Dreiecke in einem Startpatch geschieht nur anhand des Richtungsvektors  $\hat{f}$  von  $D_{Ref}$ . Die Richtungsvektoren hinzukommender Dreiecke werden somit nicht berücksichtigt. Dies wirkt sich allerdings nachteilig für den korrekten Schraffurverlauf auf diesen Dreiecken aus. In Abb. 4.17 ist dazu ein Beispiel gegeben. Zu sehen ist in (a) eine Kugel, die Richtungsvektoren für jedes Dreieck und rot das Referenzdreieck  $D_{Ref}$  für das Startpatch. Des Weiteren sind bereits die texturierten Patchdreiecke um  $D_{Ref}$  dargestellt, deren Richtungsvektor zum Referenzvektor kleiner als  $4^\circ$  abweicht. Eine Erhöhung dieses Schwellwertes auf  $25^\circ$  bewirkt ein weiteres Patchwachstum, wobei die Schraffurlinien der äußeren Dreiecke zunehmend von ihrer lokalen Richtung abweichen. Dieser Effekt ist in (b) dargestellt, wo für eine bessere Veranschaulichung die Ansicht auf die Kugel rotiert wurde. Der Richtungsverlauf auf den texturierten Dreiecken spiegelt nur den am Referenzdreieck wieder. Somit ist ein Winkelschwellwert  $\alpha_{divergence}$  zu setzen,

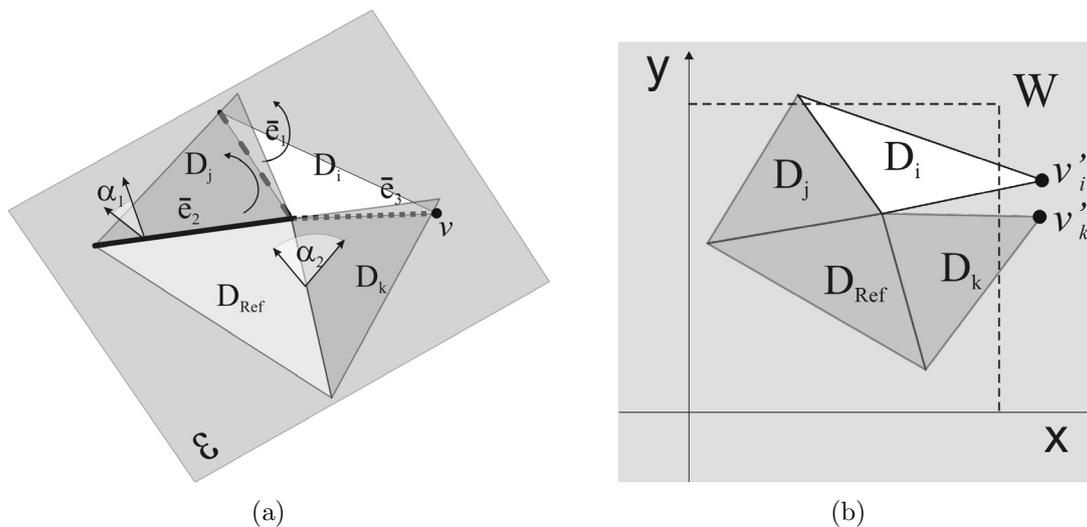


**Abb. 4.17:** Beschränkung des Patchwachstums bei starker Abweichung des Richtungsvektors neu hinzuzufügender Dreiecke bezüglich dem Referenzdreieck  $D_{Ref}$ . (a) Darstellung des texturierten Startpatch auf einer Kugeloberfläche mit Richtungsvektoren für jedes Dreieck. Der Schwellwert für die Richtungsänderung im Vektorfeld bezüglich  $D_{Ref}$  (rot) beträgt  $4^\circ$ . (b) Eine Anhebung des Schwellwertes auf  $25^\circ$  resultiert in einem breiteren Patchwachstum, wo jedoch die Schraffuren auf den äußeren Dreiecken nicht mehr dem lokalen Richtungsverlauf entsprechen (Ansicht gedreht).

der bezüglich des lokalen Richtungsverlaufes in der Nachbarschaft das Patchwachstum über eine Kante einschränkt. Für die korrekte Bestimmung der Winkeldifferenz wird für ein neu hinzu zu fügendes Dreieck  $D$  dessen Richtungsvektor in die Tangentialebene von  $D_{Ref}$  projiziert. Übersteigt der berechnete Winkel den Schwellwert, so wird das Dreieck verworfen und die Kante als Außenkante markiert.

## Kriterium 2: Kein Aufschneiden der Geometrie

Während des Patchwachstums kann es passieren, dass die Dreiecksgeometrie „aufgeschnitten“ werden muss. Dazu wird folgende Situation betrachtet, die in Abb. 4.18 zum besseren Verständnis veranschaulicht ist. Ein Dreieck  $D_i$  soll in (a) zuerst um die Kante  $\bar{e}_1$  und dann um  $\bar{e}_2$  in die Ebene  $\epsilon$  des Referenzdreieckes rotiert werden. Tritt nun der Fall ein, dass zwei Dreiecke  $D_j$  und  $D_k$ , die inzident zu beiden Kanten und bereits in die Ebene rotiert sind, dann muss die Geometrie an der Kante  $\bar{e}_3$  „aufgeschnitten“ werden. Analog zum Auffalten der Flächen einer Pyramide. Durch dieses Aufbrechen an  $\bar{e}_3$  und die anschließende Transformation in den Parameterraum würde es zu zwei unterschiedlichen Positionen der Eckpunkte  $v_i$  und  $v_k$  führen, wie in (b) zu sehen ist. Das Ergebnis wäre ein Diskontinuität im späteren Texturverlauf zwischen den Dreiecken  $D_i$  und  $D_k$ , da die beiden Eckpunkte zwei verschiedene Parameterkoordinaten erhalten. Die Diskontinuität innerhalb des *Patch* würde steigen, je größer die Normalenwinkel  $\alpha_1$  und  $\alpha_2$  zwischen den Dreiecken  $\angle(D_j, D_{Ref})$  bzw.  $\angle(D_k, D_{Ref})$  sind. Um das Aufschneiden zu verhindern,



**Abb. 4.18:** (a) Hinzufügen eines Dreiecks  $D_i$  zu einem *Patch* durch Rotation an den Kanten  $\bar{e}_1$  und  $\bar{e}_2$  in der Referenzebene  $\epsilon$ . Ist dabei neben dem Dreieck  $D_j$  noch ein zweites, inzidentes Dreieck  $D_k$  bereits in  $\epsilon$ , so muss die Geometrie beim Auffaltungsprozess an der Kante  $\bar{e}_3$  „aufgeschnitten“ werden. (b) Dies führt im Parameterraum jedoch zu zwei unterschiedlichen Parameterkoordinaten der Eckpunkte  $v_i$  und  $v_k$ , was zu späteren Texturdiskontinuitäten innerhalb des *Patch* führt.

wird entweder das Dreieck  $D_i$  nicht dem *Patch* hinzugefügt oder es erhält die Parameterpositionen für seine Eckpunkte entsprechend der transformierten drei Nachbardreiecke. Letzteres verursacht jedoch eine Texturverzerrung für dieses Dreieck, die stärker ist je größer die beiden Winkel  $\alpha_1$  und  $\alpha_2$  sind. Aus diesem Grund entscheidet ein Schwellwert

$\alpha_{rotation}$  bezüglich dieser Winkel, ob das Dreieck hinzugefügt wird oder nicht. In beiden Fällen werden die Kanten  $\bar{e}_1$  und  $\bar{e}_3$  anschließend als Abbruchkanten markiert.

### Kriterium 3: Überdeckung des Projektionsfensters

Das Projektionsfenster  $W$  stellt die Breite und Höhe der Textur im Parameterraum dar. Überdecken die Dreiecke im *Patch* diesen Bereich, so wird die Textur komplett abgebildet. Ein weiteres Patchwachstum über diesen Fensterbereich ist nicht nötig, da ein benachbartes *Patch* den kontinuierlichen Fortlauf der Textur sicherstellt. Das Fenster verhindert zudem ein „entartetes“ Patchwachstum in eine bestimmte Richtung, wenn das Hinzufügen von Dreiecken über eine Kante durch die beiden ersten Kriterien nicht abgebrochen wird. Dieser Effekt ist nicht erwünscht, da die Parametrisierung des *Patch* sich nur anhand der lokalen Dreiecke um das Referenzdreieck orientieren soll. Wie aus Abschnitt 4.4.4 noch ersichtlich werden wird, fließt bei der Vermeidung von Texturnähten der Überlappungsbereich angrenzender *Patches* mit ein. Stammen die Dreiecke dabei aus einem weiter entfernten *Patch*, aufgrund des entarteten Wachstums, so entstehen zusätzlich Textur- bzw. Richtungsverzerrungen an den Patchübergängen.

Der Wachstumsabbruch an einer Kante wird durch ein *2D-Clipping* der transformierten Dreiecksseiten gegen das Projektionsfenster  $W$  entschieden. Das *Clipping* erfolgt durch den *Cohen-Sutherland Line-Clipping* Algorithmus, der die Kanten von  $W$  in Halbräume unterteilt und die Lage einer Clippelinie binär zu diesen Halbräumen codiert (vgl. FOLEY U. A. [1990]). Auf Basis dieser Codierung ist eine effiziente Bestimmung möglich, ob ein Liniensegment komplett innerhalb bzw. außerhalb von  $W$  liegt oder das Segment die Kanten von  $W$  schneidet. Nur wenn der Test ergibt, dass eine transformierte Kante komplett außerhalb von  $W$  liegt, wird diese als Abbruchkante markiert.

### Kriterium 4: Artificielle Bereiche und schon besuchte Dreiecke

Im Abschnitt 4.1.1 wurde auf Löcher als artifiziiellen Bereiche der Oberflächengeometrie hingewiesen. Eine Kante wird daher ebenfalls als Abbruchkante markiert, wenn diese kein weiteres inzidentes Dreieck besitzt. Eine solche Markierung erhalten ebenso die Kanten, an welchen bereits eine erfolgreiche Rotation stattgefunden hat, diese also schon „besucht“ wurden. Damit wird verhindert, dass ein schon transformiertes Dreieck in ein und demselben *Patch* erneut parametrisiert wird. Zusätzlich wird ein parametrisiertes Dreieck, wie auch das Referenzdreieck, als „besucht“ gekennzeichnet. Dies ist notwendig für das im nächsten Abschnitt erläuterte Hinzufügen weiterer *Patches*.

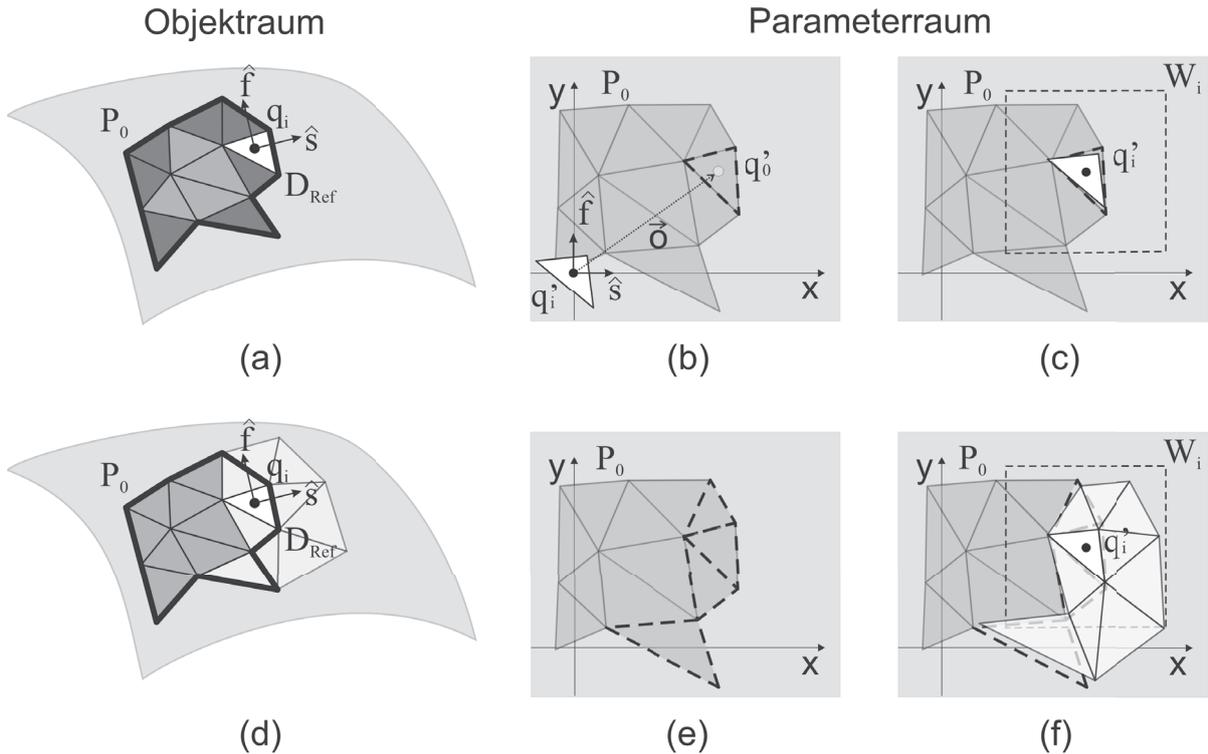
### 4.4.3 Hinzufügen weiterer Patches

Die Erzeugung weiterer *Patches* erfolgt relativ zu den bisher generierten und endet, wenn alle Oberflächendreiecke mindestens einem *Patch* zugeordnet sind. Letzteres wird über eine globale Außenkantenliste  $\bar{L}_{Kontur}$  entschieden, die über alle bisherigen *Patches* erzeugt wird. Ist diese leer, dann sind alle Dreiecke mindestens durch ein *Patch* parametrisiert. Zusätzlich ist zu jeder Kante  $e \in \bar{L}_{Kontur}$  das inzidente Dreieck an  $e$  in einer weiteren Liste  $\bar{L}'_{Kontur}$  abgespeichert, das bereits mindestens einem *Patch* angehört. Im Folgenden soll anhand dieser beiden Listen und der Abb. 4.19 die Erzeugung eines neuen *Patch* erläutert werden. Das Vorgehen ist dabei sehr ähnlich dem des Startpatch und unterscheidet sich hauptsächlich nur anhand der Wahl des Referenzdreieckes. Es wird zunächst der allgemeine Ansatz betrachtet und anschließend ein Spezialfall.

#### Allgemeines Vorgehen

Ausgangsbasis eines neuen *Patch*  $P_i$  ist wie beim Startpatch, die Wahl eines Referenzdreieckes. Für dieses wird im Gegensatz zum Startpatch kein beliebiges Dreieck der Oberfläche gewählt, sondern aus der Liste  $\bar{L}'_{Kontur}$ . Der Grund dafür liegt in einer nahen Parametrisierung von  $P_i$  bezüglich bestehender *Patches*, um einen späteren kontinuierlichen Übergang zwischen diesen zu erzeugen. Der Abschnitt 4.4.4 wird darauf noch näher eingehen. In Abb. 4.19 (a) ist dazu das Startpatch  $P_0$  auf einer Oberfläche dargestellt.

Markiert sind alle äußeren Dreiecke und Kanten von  $P_0$ , als aktuelle Liste  $\bar{L}'_{Kontur}$  sowie  $\bar{L}_{Kontur}$ . Weiterhin ist aus  $\bar{L}'_{Kontur}$  das Dreieck  $D_{Ref}$  als Referenzdreieck von  $P_i$  hervorgehoben. Für  $D_{Ref}$  wird ebenfalls eine lokale Basis aus seinem Richtungsvektor  $\hat{f}$  und seiner Normalen gebildet. Anhand seines Mittelpunktes  $q$  wird das Referenzdreieck über die Transformationen von Gleichungen 4.18 und 4.19 in den Ursprung des Parameterraumes überführt. Dieser Schritt ist in (b) dargestellt. Zusätzlich sind in dieser Abbildung die transformierten Dreiecke von  $P_0$  zu sehen sowie die Hervorhebung der alten Position des Referenzdreieckes darin. Da es sich bei  $D_{Ref}$  bereits um ein parametrisiertes Dreieck handelt, muss die neue Parametrisierung ihrer bisherigen möglichst nah sein. Um dies zu erreichen entspricht der Verschiebungsvektor  $\vec{o}$  aus Gleichung 4.20 dem Mittelpunkt  $q'_0$  aus der letzten Parametrisierung von  $D_{Ref}$ , d. h. aus dem *Patch*  $P_0$ . Bestimmt wird dieser über die Mittelung der alten transformierten Eckpunktpositionen von  $D_{Ref}$ . In (c) ist die resultierende Translation zu sehen. Zu erkennen ist ein leichter Versatz des Referenzdreieckes zu seiner alten Position. Die leichten Abweichungen resultieren zum einen aus der Basistransformation mit dem neuen Richtungsvektor von  $P_i$ , der leicht abweicht von dem des Startpatches  $P_0$ . Zum anderen kann das Dreieck  $D_{Ref}$  in der vorhergehenden Parametrisierung nicht durch eine affine Transformation in den Parameterraum überführt worden sein. Dies ist dann der Fall, wenn sich die Eckpunkte aus den bereits parametrisierten Nachbardreiecken ergeben (vgl. Kriterium 3 in Abschnitt 4.4.2).



**Abb. 4.19:** Hinzufügen weiterer *Patches* zum Startpatch. In (a) ist die aktuelle Außenkontur (dicke Linie) des Startpatch und die daran befindlichen, parametrisierten Außendreiecke (dunkelgrau). Das Referenzdreieck  $D_{Ref}$  für ein neues *Patch* entspricht einem dieser Außendreiecke. (b) Die Translation und Ausrichtung von  $D_{Ref}$  in den Parameterraum erfolgt anhand seines Mittelpunktes und seiner lokalen Basis. Der Verschiebungsvektor  $\vec{o}$  entspricht jedoch seinem transformierten Mittelpunkt  $q'_0$  aus seiner vorherigen Parametrisierung im Startpatch  $P_0$ . Zusätzlich sind von  $P_0$  seine transformierten Dreiecke inklusive die „alte“ Position von  $D_{Ref}$  dargestellt. (c) Aufgrund des angepassten Richtungsvektors  $\vec{o}$  resultiert die abschließende Translation von  $D_{Ref}$  einer relativen Position zu seiner alten Lage. (d) Beim weiteren Patchwachstum können Dreiecke schon im Startpatch enthalten sein. (e)-(f) Transformation der neuen Dreiecke relativ zu  $D_{Ref}$  resultiert bei bereits parametrisierten Dreiecken zu leichten Abweichungen. Dieser Randbereich erfordert eine nachträgliche Behandlung.

Die dabei auftretende, leichte Verzerrung überträgt sich auch auf die oben beschriebene Bestimmung von  $q'_0$ . Zusätzlich ist in (c) das Projektionsfenster  $W_i$  des neuen *Patch* eingezeichnet. Dessen Ausmaße ergeben sich durch  $q'_i \pm \frac{w}{2}$ .

Das weitere Wachstum von  $P_i$  stellen die unteren Abbildungen von 4.19 dar. In (d) sind bereits zusätzliche Dreiecke dem *Patch* hinzugefügt worden. Das Hinzufügen von neuen Dreiecken zu  $P_i$  geschieht dabei analog zum Startpatch (vgl. Abschnitt 4.4.1) unter Berücksichtigung der Abbruchkriterien. Erkennbar ist, dass einige Dreiecke mit aufgenommen worden, die auch Teil des Startpatches waren. Möglich ist auch, dass die

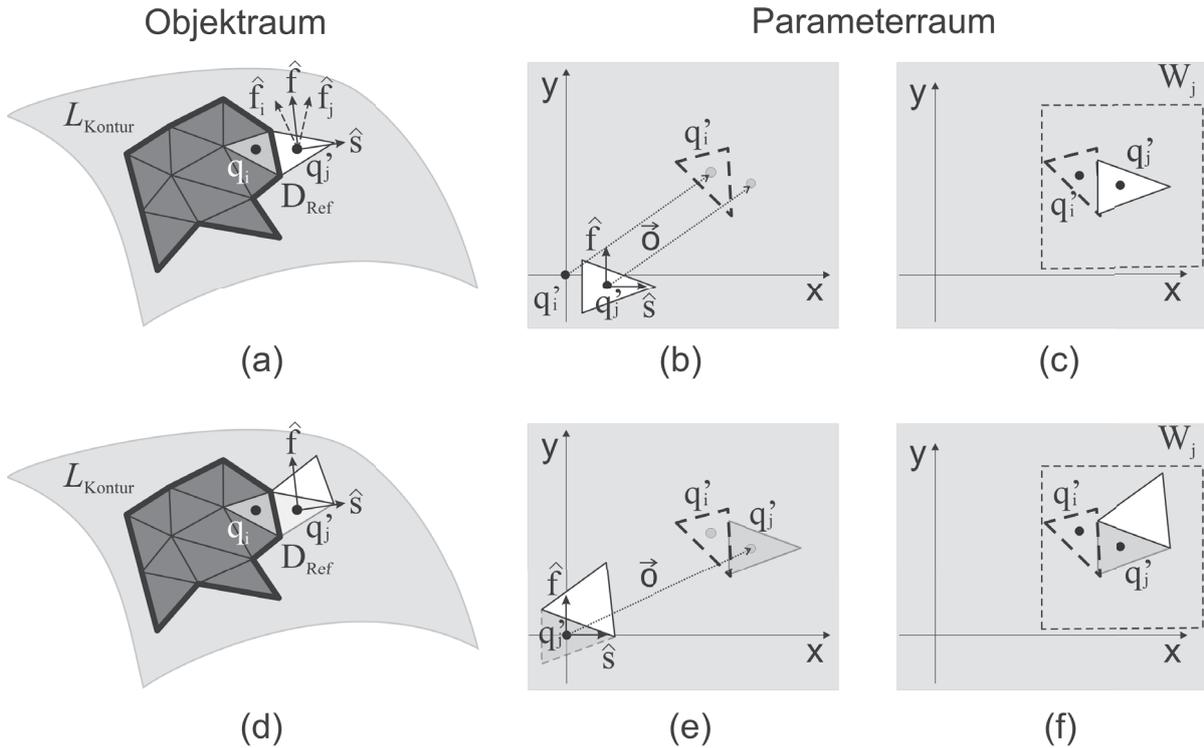
Dreiecke eines neuen *Patch* Teil von mehreren Nachbarpatches sein können. Im vorliegenden Fall des Startpatch werden in (e) noch mal die alten Positionen der überdeckten Dreiecke von  $P_0$  im Parameterraum hervorgehoben. Die Abbildung (f) zeigt die resultierende Transformation der neu hinzugefügten Dreiecke in den Parameterraum anhand der Transformation  $\mathbf{X}$  aus Gleichung 4.21. Verwendet wird dabei der nur der neue Mittelpunkt  $q_i$  des Referenzdreieckes sowie der angepasste Verschiebungsvektor  $\vec{o}$ . Zu erkennen ist in (f) ebenfalls eine leichte Abweichung der überlappenden Dreiecke bezüglich ihrer alten Position. Dies folgt entsprechend der Abweichung ihres Referenzdreieckes  $D_{Ref}$ .

Ist ein weiteres Patchwachstum abgeschlossen, so wird die Parametrisierung der Eckpunkte in dem neuen *Patch* separat abgespeichert. Dazu wird Somit können mehrere Parameterkoordinaten für ein Dreieck existieren, abhängig davon in wie vielen *Patches* es vorliegt. Dies verursacht zunächst Diskontinuitäten der Texturen an den Patchübergängen, deren Behandlung sich Abschnitt 4.4.4 widmet. Weiterhin werden die beiden Listen  $\bar{L}_{Kontur}$  und  $\bar{L}'_{Kontur}$  aktualisiert, woraus wiederum ein Referenzdreieck für den Start eines neuen *Patch* gewählt wird. Dieses Vorgehen wird so lange wiederholt, bis die Liste  $\bar{L}_{Kontur}$  leer ist und es somit keine unparametrisierten Oberflächendreiecke mehr gibt.

## Spezialfall

Beim Hinzufügen von neuen *Patches* kann es passieren, dass das Wachsen nicht erfolgreich war. Nicht erfolgreich bedeutet in dem Fall, dass sich über die ganze Konturliste  $\bar{L}'_{Kontur}$  kein Referenzdreieck finden ließ, für das ein neues Patchwachstum eine Parametrisierung neuer Dreiecke erzielte. Dies bedeutet, dass es keine weitere Patchausbreitung über die Oberfläche ergab und demzufolge die Liste  $\bar{L}'_{Kontur}$  bzw.  $\bar{L}_{Kontur}$  sich nicht verändert hat. Ursache dafür sind die oben beschriebenen Abbruchkriterien für ein Patchwachstum. Werden aufgrund dieser Kriterien keine neuen Randdreiecke an  $\bar{L}'_{Kontur}$  in das Wachstum mit aufgenommen, so würde der Parametrisierungsprozess stagnieren. Daher ist in einem solchen Fall ein anderes Vorgehen für die Wahl des Referenzdreieckes notwendig.

Im Gegensatz zum allgemeinen Vorgehen entspricht in der eingangs erwähnten Situation das Referenzdreieck einem bisher unparametrisierten Dreieck. Dieses grenzt dabei direkt an einer der Patchaußenkanten. Zur Veranschaulichung dient die Abb. 4.20. In (a) ist die aktuelle Außenkontur  $L_{Kontur}$  mit allen bisher parametrisierten Dreiecken markiert. Das neue Referenzdreieck  $D_{Ref}$  eines weiteren *Patch*  $P_j$  entspricht dem inzidenten Dreieck an einer Konturkante, das nicht Teil der bisherigen *Patches* ist. Für die Bestimmung der lokalen Basis von  $D_{Ref}$  setzt sich der Richtungsvektor  $\hat{f}$  aus dem Mittel des eigenen Vektors  $\hat{f}_j$  mit  $\hat{f}_i$ , dem Vektor seines inzidenten aber bereits parametrisierten Dreieckes  $D_i$ , zusammen. Der Grund für diese Mittelung ergibt sich aus der Annahme, dass die Richtungsänderung der unparametrisierten Dreiecke an der Außenkontur über dem eingestellten Winkelschwellwert liegt (vgl. Kriterium 1 in Abschnitt 4.4.2). Eine Bildung der Basis aus dem originalen Richtungsvektor  $\hat{f}_j$  würde bei der anschließenden



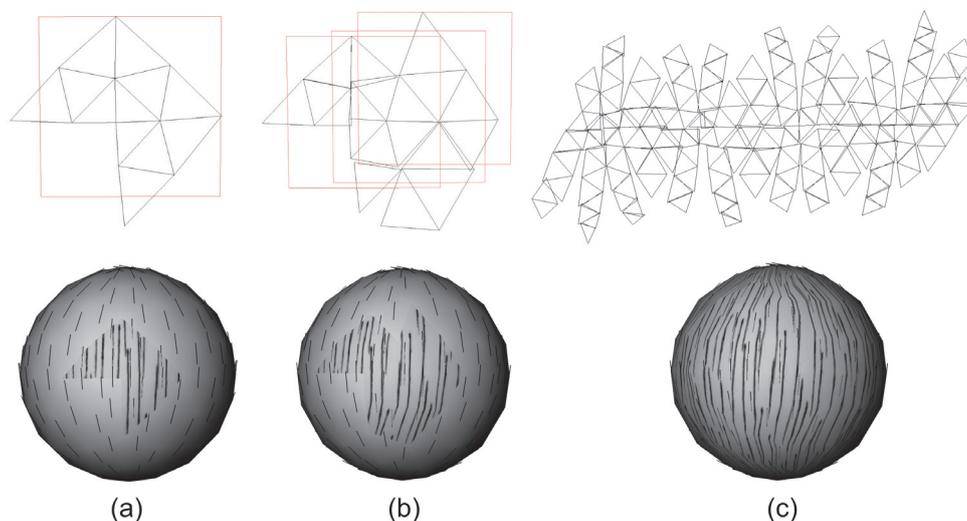
**Abb. 4.20:** Auswahl des Referenzdreieckes im Fall eines nicht erfolgreichen Patchwachstums: (a) Referenzdreieck  $D_{Ref}$  eines neuen  $Patch$   $P_j$  entspricht einem Dreieck an der Außenkontur von  $L_{Kontur}$ , das noch nicht parametrisiert ist. Richtungsvektor  $\hat{f}$  für die Basistransformation entspricht Mittelung aus eigenem Vektor  $\hat{f}_j$  und Vektor  $\hat{f}_i$  seines inzidenten und bereits parametrisierten Dreiecks  $D_i$ . (b) und (c) zeigen die Transformation von  $D_{Ref}$  relativ zu  $D_i$  in den Parameterraum. Jedes neu hinzugefügte Dreieck aus (d) wird nun entsprechend dem Referenzdreieck in den Parameterraum überführt. Dabei entspricht der Verschiebungsvektor  $\vec{o}$  in (e) dem transformierten Mittelpunkt  $q'_j$  von  $D_{Ref}$  im Projektionsfenster. (f) zeigt die relative Position der neuen Dreiecke von  $P_j$  bezüglich benachbarter, bereits parametrisierter Dreiecke.

Transformation in den Parameterraum zu stärkeren Diskontinuitäten führen. Die Abbildung (b) zeigt die Überführung von  $D_{Ref}$  in die  $x$ - $y$ -Ebene. Dabei orientiert sich die Translation in den Ursprung am Mittelpunkt  $q_i$  vom Dreieck  $D_i$ . Weiterhin ist in (b) die relative Verschiebung von  $D_{Ref}$  bezüglich dem bereits parametrisierten Dreieck  $D_i$  nach der Basistransformation zu sehen. Der Verschiebungsvektor  $\vec{o}$  entspricht dabei dem transformierten Mittelpunkt  $q'_i$  von  $D_i$  im Projektionsfenster. Das Ergebnis ist in (c) dargestellt, zusätzlich mit dem neuen Projektionsfenster  $W_j$ .

Für das weitere Patchwachstum um  $D_{Ref}$  wie es in (d) dargestellt ist, wird jedes neu hinzugefügte Dreieck wieder relativ zum Referenzdreieck transformiert. Die Abbildungen (e) und (f) zeigen die anschließende Basistransformation und Translation in das Projektionsfenster, entsprechend der Ausrichtung und Lage von  $D_{Ref}$ . Der in (b) darge-

stellte Verschiebungsvektor  $\vec{\sigma}$  ändert sich in (e) bezüglich der transformierten Lage des Mittelpunktes  $q'_j$  von  $D_{Ref}$ .

Nach dem so erweiterten Patchwachstum ändern sich auch die Außenkonturlisten und das Hinzufügen von neuen *Patches* erfolgt wieder nach dem allgemeinen Vorgehen. Zur Veranschaulichung zeigt die Abb. 4.21 den oben beschriebenen Auffaltungsprozess der Oberfläche in die Ebene am Beispiel einer Kugel. Die Abbildungen stammen dabei direkt aus dem hier umgesetzten Verfahren. Zu sehen ist in der oberen Reihe jeweils der Parameterraum, in dem die bisherige Oberfläche transformiert ist sowie in der unteren Reihe der Objektraum mit dem korrespondierenden, schraffierten Oberflächenbereich. Zusätzlich ist im Objektraum das zugrundeliegende Vektorfeld dargestellt sowie in den ersten beiden Abbildungen die jeweiligen Projektionsfenster (rot) im Parameterraum. Des Weiteren sind die einzelnen Patchübergänge bereits behandelt, wodurch die meisten Texturnähte aufgelöst sind (siehe nächster Abschnitt). In Abb. 4.21 (a) ist zunächst das erzeugte und aufgefaltete Startpatch zu sehen. Das Projektionsfenster wird nicht vollständig überdeckt, da aufgrund der Abbruchkriterien einige Dreiecke nicht mit aufgenommen worden. Die Erzeugung von drei *Patches* ist in Abb. 4.21 (b) dargestellt. Erkennbar sind die überlappenden Bereiche und die neuen, leicht veränderten Parameterpositionen von Dreiecken, die in mehreren *Patches* vorkommen. In Abb. 4.21 (c) ist die vollständig aufgefaltete und damit parametrisierte Oberfläche abgebildet. Der Schraffurlinienverlauf entspricht dabei dem zugrundeliegenden Vektorfeld.



**Abb. 4.21:** Darstellung des Auffaltungsprozesses einer Kugeloberfläche in den Parameterraum: (a) Zu Beginn wird das Startpatch erzeugt und parametrisiert. Einige Dreiecke sind aufgrund der Abbruchkriterien dabei nicht mit aufgenommen worden. (b) Darstellung von drei sich überlappenden *Patches*, was zu variierenden Parameterpositionen der betroffenen Dreiecke führt. (c) Vollständig aufgefaltete Oberfläche im Parameterraum und daraus resultierende Schraffierung der Kugel, entsprechend dem Vektorfeld.

#### 4.4.4 Behandlung der Patchübergänge

Die bisherige Parametrisierung ergibt eine Ansammlung von *Patches*, die die ganze Oberfläche überdecken. Dabei treten Überlappungen auf, so dass ein Dreieck Teil von mehreren, nah aneinanderliegenden *Patches* sein kann. Aus diesem Grund besitzen die Eckpunkte eines solchen Dreiecks mehrere Parameterkoordinaten, entsprechend der Anzahl ihrer zugehörigen *Patches*. Dies verursacht jedoch Texturdiskontinuitäten bzw. -nähte an den Patchgrenzen, da die überlappenden Dreiecke in jeden ihrer *Patches* eine leicht andere Texturausrichtung erhalten.

Zur Behandlung dieser Diskontinuitäten müssen die verschiedenen Parameterkoordinaten der Eckpunkte geeignet miteinander kombiniert werden. Im Folgenden soll dieses Vorgehen erläutert werden. Dabei zeigt es sich, dass zwei verschiedene Fälle berücksichtigt werden müssen. Die Fallunterscheidung bezieht sich zum einen auf *Patches*, die im Objekt- und Parameterraum benachbart sind und zum anderen auf *Patches*, die im Objektraum aber nicht im Parameterraum benachbart sind.

##### Fall 1: Patches im Objekt- und Parameterraum sind inzident

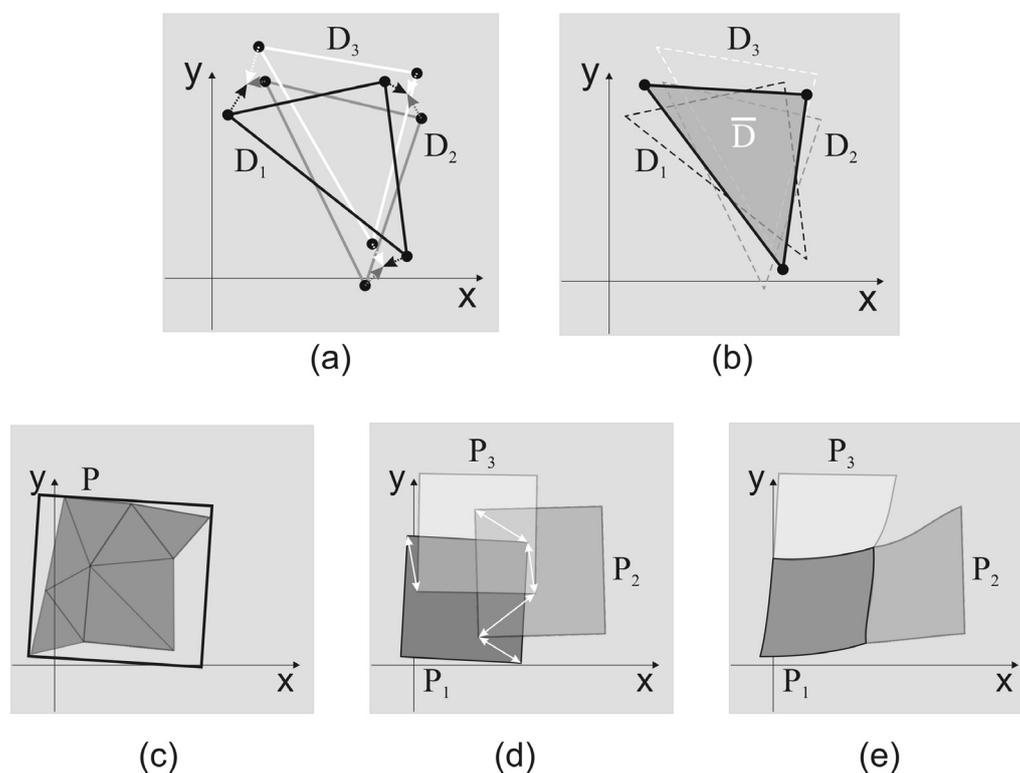
Im ersten Fall werden *Patches* betrachtet, die eine direkte Nachbarschaft im Objektraum sowie Parameterraum besitzen. Dies trifft für die meisten *Patches* zu, da der Auffaltungsprozess und die Transformation in die  $x$ - $y$ -Ebene eines neuen *Patches* stets an den Grenzen der Bestehenden erfolgt. Der Umgang mit den verschiedenen Parameterkoordinaten der Eckpunkte eines Dreiecks  $D$  erfolgt in der Weise, dass die Koordinaten pro Eckpunkt gemittelt werden. Formal erfolgt dies mit:

$$D(u_i, v_i) = \frac{1}{n} \sum_{k=1}^n (u_k^i, v_k^i), \text{ mit } i \in \{1, 2, 3\}, \quad (4.26)$$

wobei  $n$  für die Anzahl der Parameterkoordinaten des Eckpunktes  $i$  steht. Dieses Vorgehen begründet sich in der Annahme, dass die Positionen sich nicht stark voneinander unterscheiden und somit die Mittelung eine geeignete Angleichung ergibt. Aus diesem Grund wurde bei der Parametrisierung auch darauf geachtet, dass es unter korrespondierenden, überlappenden Dreiecken zu keinen großen Abweichungen bezüglich der Parameterkoordinaten kommt (vgl. Abbruchkriterien im Abschnitt 4.4.2).

Die Mittelung veranschaulicht Abb. 4.22. In (a) ist zunächst ein Dreieck  $D$  mit drei verschiedenen Parametrisierungspositionen aus drei benachbarten und sich überlappenden *Patches* dargestellt. Die Mittelung über jeden Eckpunkt entspricht dem resultierenden Dreieck  $\bar{D}$  in (b). Das Dreieck erfährt dabei eine leichte Stauchung oder Streckung. Eine Übertragung der Mittelung auf überlappende *Patches* zeigt die untere Reihe von Abb. 4.22. Dazu wird in (c) ein *Patch*  $P$  als orientierte *Bounding Box* seiner enthaltenen Dreiecke vereinfacht dargestellt. Überdecken sich nun verschiedene *Patches*, wie in (d), so

bewirkt die Mittelung ein Angleichen der Überlappungsbereiche (e). Der Abstraktionsgrad der Zeichnungen führt zu der Annahme, dass die Stauchungen und Streckungen starke Texturverzerrungen verursachen. Jedoch zeigt das Ergebniskapitel, dass die vorhandenen leichten Verzerrungen zu keinen sichtbaren Texturartefakten führen.



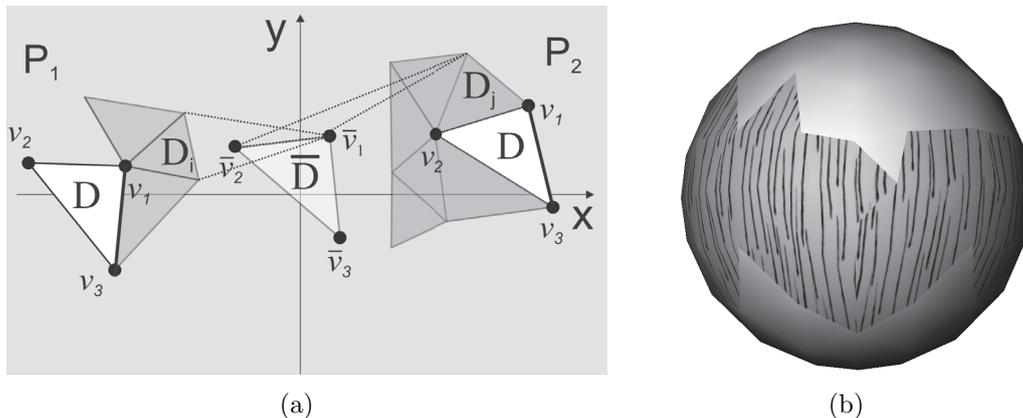
**Abb. 4.22:** Behandlung von Texturnähten im Falle von *Patches*, die im Objekt- und Parameterraum benachbart sind. (a) Die Eckpunkte eines Dreiecks  $D$ , das in drei sich überlappenden *Patches* drei verschiedene Parameterkoordinaten erhalten haben, werden in (b) gemittelt und ergeben das Dreieck  $\bar{D}$ . (c) und (d) Abstraktere Repräsentation eines *Patch* zur anschaulicheren Übertragung der unter (a) gezeigten Mittelung auf drei sich überlagernden *Patches*  $P_1, P_2, P_3$ . (e) Nach der Mittelung besitzt jeder Eckpunkt nur eine Parameterkoordinate, was zu einem Angleichen benachbarter *Patches* führt.

### Fall 2: Patches im Objekt- und Parameterraum sind nicht inzident

Der zweite Fall betrachtet *Patches*, die zwar im Objektraum, jedoch nicht im Parameterraum benachbart sind. Diese Situation tritt ein, wenn das Hinzufügen von neuen *Patches* einen Zyklus um das Objekt vollzogen hat. Vergleichbar ist dies mit dem Abwickeln eines offenen Zylinders. Das Abwickeln ergibt ein Rechteck, deren Schnittkante in

2D auseinander liegt. Die Behandlung von überlappenden Dreiecken, die in solchen *Patches* enthalten sind, erfordert ein anderes Vorgehen als im ersten Fall. Eine Anwendung der oben beschriebenen Mittelung der Parameterkoordinaten würde hier zu extremen Verzerrungen der angrenzenden Dreiecke führen. Die Abb. 4.23 (a) veranschaulicht das Problem an einem Beispiel. Dargestellt sind zwei *Patches*  $P_1$  und  $P_2$ , in beiden ist das Dreieck  $D$  enthalten. Die Mittelung der Eckpunkte ergibt das Dreieck  $\bar{D}$ , welches einen starken Versatz im Parameterraum zu den Nachbardreiecken in seinem jeweiligen *Patch* besitzt. Dies verursacht eine hohe Verzerrung der angrenzenden Dreiecke  $D_i$  und  $D_j$ .

Im Umgang mit diesem Fall wird zunächst ein weiteres Abbruchkriterium zum Patchwachstum eingeführt. Es werden sogenannte Schnittkanten definiert, die ein weiteres Hinzufügen von Dreiecken über diese Kanten verhindern. Bestimmt wird eine Schnittkante aus der aktuellen, globalen Außenkantenliste  $\bar{L}_{Kontur}$ . Die Eckpunkte jeder Außenkantenliste besitzen aus mindestens einem *Patch* bereits Parameterkoordinaten. Wird ein Dreieck einem neuen *Patch* hinzugefügt, so wird überprüft, ob eines seiner Kanten einer Kante aus  $\bar{L}_{Kontur}$  entspricht. Ist dies der Fall, so werden die neuen Parameterkoordinaten der Kanteneckpunkte gegen die bereits vorhandenen verglichen. Entspricht die Positionsdifferenz mehr als die Breite des Projektionsfensters, so wird die Kante als Schnittkante markiert. Ein Zyklus im Patchwachstum wurde gefunden und entspricht für die Abb. 4.23 (a) der Kante zwischen  $v_1$  und  $v_3$ . Das Abbrechen an dieser Stelle verursacht eine spätere Texturnaht, die in Abb. 4.23 (b) am Beispiel einer Kugel zu sehen ist. Der Grund dafür liegt in dem gezeigten Beispiel an der momentanen Existenz von je zwei verschiedenen Parameterkoordinaten der Schnittkantenpunkte. Für jedes der beiden inzidenten Dreiecke an einer Kante entspricht dies somit einem anderen Texturausschnitt.



**Abb. 4.23:** (a) Beispielkonstellation von zwei sich im Objektraum überlagernden *Patches*, die jedoch im Parameterraum nicht benachbart sind. Eine Mittelung der Parameterkoordinaten des Dreieckes  $D$ , welches in beiden *Patches* enthalten ist ( $\bar{D}$ ), verursacht eine zu starke Verzerrung der inzidenten Dreiecke  $D_i, D_j$  (punktiert). Zur Behandlung wird die Kante  $v_1, v_3$  als Schnittkante definiert, die ein weiteres Patchwachstum darüber verhindert. Diese verursacht zunächst eine in (b) dargestellte Texturnaht, die einen Zyklus im Patchwachstum kennzeichnet.

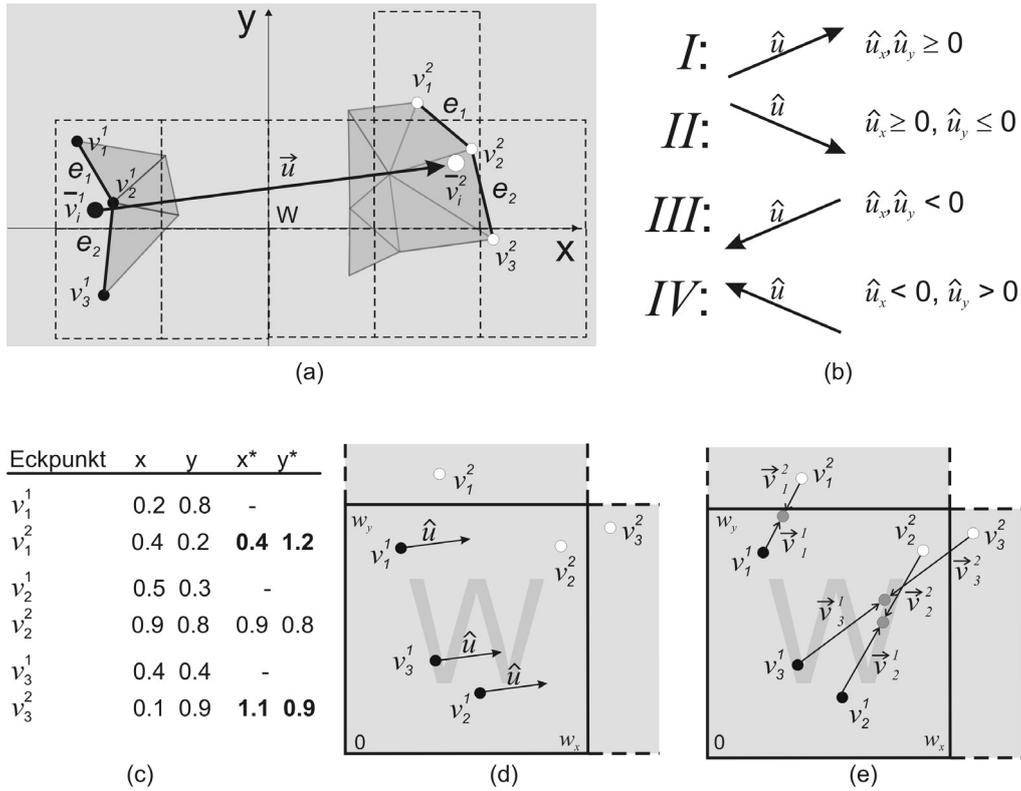
Für die Vermeidung der Schnittkante wird ein Relaxationsansatz verwendet. Im ersten Schritt werden dazu die unterschiedlichen Parameterkoordinaten der Schnittkantenpunkte relativ zum Parameterraum gleich ausgerichtet. Dabei entsteht eine „Spannung“ an den Schnittkantenpunkten, welche sich in Form von starken Verzerrungen an den inzidenten Dreiecken der Schnittkanten äußert. Der zweite Schritt beinhaltet eine Relaxation, also Entspannung, durch ein Verteilen der Verzerrungen auf die umliegenden Eckpunkte. Das Vorgehen soll im Weiteren mit Hilfe der Abbildungen 4.24 und 4.25 näher erläutert werden.

In Abb. 4.24 (a) sind zunächst beispielhaft Ausschnitte von parametrisierten Dreiecken gezeigt, wo die gemeinsamen Kanten  $e_1, e_2$  zwei Schnittkanten bilden. Deren Eckpunkte besitzen zwei Koordinatenpaare, welche jeweils mit  $v_1^1, v_1^2, v_2^1, v_2^2, v_3^1$  und  $v_3^2$  bezeichnet sind. Untereinander besitzen korrelierende Koordinatenpaare einen Abstand, der größer ist als die Breite des Projektionsfensters  $W$  und die Eckpunkte dadurch als Schnittkantenpunkte identifiziert sind. Jedes Parameterkoordinatenpaar eines Eckpunktes beschreibt dabei eine relative Position in der späteren Textur. Erkennbar ist dies durch die ange deutete Unterteilung des Parameterraumes in ein Vielfaches des Projektionsfensters (vgl. auch Bestimmung Parameterkoordinaten in Abschnitt 4.4.1).

Im Allgemeinen stimmen diese Positionen zwischen den verschiedenen Parameterkoordinaten eines Eckpunktes nicht überein. Ein kontinuierlicher Übergang in der Textur ist daher nicht möglich, wie dies auch Abb. 4.23 (b) zeigt. Damit es zu keiner Diskontinuität an einer Schnittkante kommt, müssen die verschiedenen Parameterkoordinatenpaare der Schnittkantenpunkte eine kohärente Position im Parameterraum einnehmen. Dies ist der Fall, wenn:

- jedes Koordinatenpaar auf gleicher Höhe oder Breite der Kanten des Projektionsfensters liegt. Damit beschreiben sie jeweils einen Texturanfang bzw. -ende oder
- die Koordinatenpaare besitzen die gleiche relative Position im Projektionsfenster.

Die dabei entstehenden Verschiebungen der einzelnen Koordinatenpaare in der jeweiligen Parameterraumseite können in beiden Fällen jedoch untereinander entgegengesetzt sein. Dadurch hebt sich die spätere Verschiebung ihrer benachbarten Eckpunkte für die Relaxation auf oder ist nur sehr gering. Somit bleibt die initiale, starke Verzerrung bestehen. Um diese Stagnierung zu verhindern, wird eine grobe Entspannungsrichtung definiert, in der sich die Koordinatenpaare der jeweiligen Schnittkantenseite relaxieren. Gewünscht ist dabei ein minimal gestörter, lokaler Richtungsverlauf in der Textur sowie eine geringe Anfangsverzerrung, deren Relaxation zu gleichen Anteilen auf der jeweiligen Seite der Schnittkante verläuft.



**Abb. 4.24:** Bestimmung der Verschiebungsvektoren, um korrespondierende Parameterpaare von Schnittpunkten im Parameterraum gleich auszurichten: (a) Beispielkonstellation von zwei *Patches*, die zwei Schnittkante  $e_1, e_2$  bilden. Diese sind gekennzeichnet, dass ihre Eckpunkte  $v_1, v_2, v_3$  zwei im Parameterraum nicht benachbarte Koordinatenpaare besitzen. Der Parameterraum ist mit dem Vielfachen des Projektionsfensters  $W$  unterteilt, um die relative Position jeder Koordinate in  $W$  zu verdeutlichen. Zusätzlich definiert der Vektor  $\vec{u}$  (bzw.  $-\vec{u}$ ) die Vorzugsrichtung der jeweiligen Koordinatenpaare jeder Schnittkantenseite. Die vier möglichen Richtungen von  $\vec{u}$  in 2D sind dazu in (b) dargestellt, abhängig von dessen Komponenten. (c) und (d) Relative Ausrichtung der zweiten Koordinatenpaare an ihre ersten Referenzkoordinatenpaare an ein gemeinsames Projektionsfenster  $W$ . Dabei erhalten  $v_1^2$  und  $v_3^2$  temporär neue relative Positionen entsprechend des Vektors  $\vec{u}$ . (e) Durch anschließende Mittelung korrespondierender Parameterpaare ergeben sich die Richtungsvektoren für die Koordinatenpaare an der jeweiligen Schnittkantenseite.

Zur Definierung der Entspannungsrichtung werden zunächst die Schwerpunkte  $\bar{x}_i^1, \bar{x}_i^2$  der Koordinatenpaare im korrespondierenden Parameterbereich gebildet (vgl. Abb. 4.24 (a)). Der sich daraus ergebene Verschiebungsvektor  $\vec{u} = \bar{x}_i^2 - \bar{x}_i^1$  entspricht dann der groben Richtung für die Relaxation der Koordinatenpaare  $v_1^1, v_2^1, v_3^1$  bzw.  $-\vec{u}$  für  $v_1^2, v_2^2, v_3^2$ . Nun werden

1. Alle ersten Koordinatenpaare eines Schnittpunktes als Referenzpaare be-

trachtet,

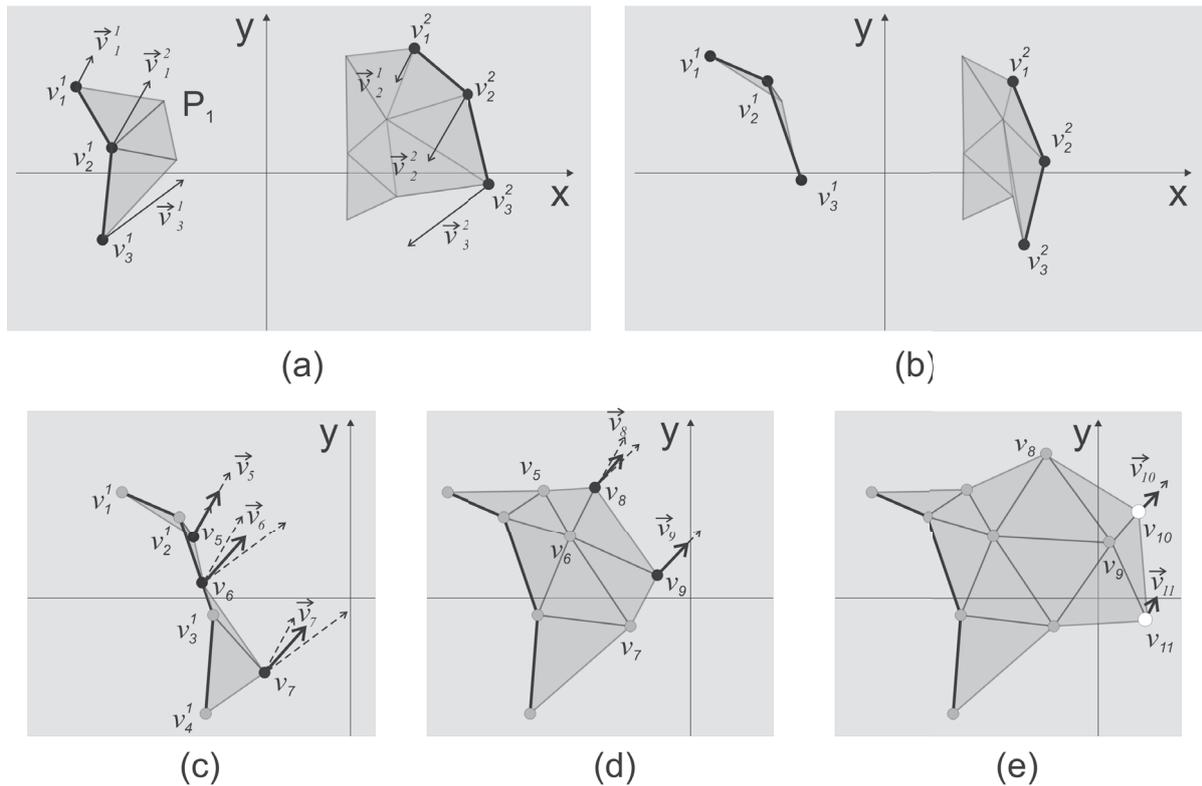
2. diese in ein gemeinsames Projektionsfenster  $W$  mit der Dimension  $(w_x, w_y)$  übertragen und
3. ihre korrelierenden Koordinatenpaare relativ zu ihnen an  $W$  positioniert.

Die Platzierung ist dabei abhängig von den Komponenten der normierten Richtung  $\hat{u}$ , wobei darin vier Fälle unterschieden werden, die in Abb. 4.24 (b) gezeigt sind. Die vier Fälle beschreiben alle möglichen Verschiebungsrichtungen eines Punktes in der Ebene. Je nachdem, welcher Fall vorliegt, ergibt sich die Position  $(x^*, y^*)$  für ein korrespondierendes Koordinatenpaar  $(x_i^2, y_i^2)$  gegenüber seinem Referenzpaar  $(x_i^1, y_i^1)$  zum Projektionsfenster  $W$  wie folgt:

$$\begin{aligned}
 I : x^* &= x_i^2 + w_x, \text{ wenn } x_i^2 < x_i^1, \text{ sonst } & x^* &= x_i^2 \\
 y^* &= y_i^2 + w_y, \text{ wenn } y_i^2 < y_i^1, \text{ sonst } & y^* &= y_i^2 \\
 II : x^* &= x_i^2 + w_x, \text{ wenn } x_i^2 < x_i^1, \text{ sonst } & x^* &= x_i^2 \\
 y^* &= y_i^2 - w_y, \text{ wenn } y_i^2 > y_i^1, \text{ sonst } & y^* &= y_i^2 \\
 III : x^* &= x_i^2 - w_x, \text{ wenn } x_i^2 > x_i^1, \text{ sonst } & x^* &= x_i^2 \\
 y^* &= y_i^2 - w_y, \text{ wenn } y_i^2 > y_i^1, \text{ sonst } & y^* &= y_i^2 \\
 IV : x^* &= x_i^2 - w_x, \text{ wenn } x_i^2 > x_i^1, \text{ sonst } & x^* &= x_i^2 \\
 y^* &= y_i^2 + w_y, \text{ wenn } y_i^2 < y_i^1, \text{ sonst } & y^* &= y_i^2
 \end{aligned} \tag{4.27}$$

Diese relative Ausrichtung von  $(x_i^2, y_i^2)$  zu  $(x_i^1, y_i^1)$  ist notwendig, damit für beide Koordinatenpaare eine „aufeinander zugehende“ Verschiebungsrichtung definiert wird. Daraus resultiert einerseits eine Gesamtbewegungsrichtung aller korrespondierenden Koordinatenpaare der Schnittkantenpunkte aufeinander zu. Andererseits wird dadurch eine gegensätzliche Verschiebung von Koordinatenpaaren in der jeweiligen Parameterseite verhindert.

In Abb. 4.24 (c) ist dies exemplarisch für die im Beispiel verwendeten Koordinatenpaare und der Dimension von  $(1.0, 1.0)$  für  $W$  gezeigt. Der Richtungsvektor für die Beispielkonstellation der Schnittkante entspricht dem Fall  $I$ . Dadurch erhalten  $v_1^2$  und  $v_3^2$  entsprechend der Gleichung 4.27 eine neue Position außerhalb von  $W$ . Dies ist in (d) zu sehen. Nun erfolgt in (e) eine Mittelung zwischen den Koordinatenpaaren eines Eckpunktes, wodurch sich für beide Positionen ein entsprechender Verschiebungsvektor ergibt. Bezeichnet ist diese jeweils mit  $\vec{v}_1^1, \vec{v}_1^2, \vec{v}_2^1, \vec{v}_2^2, \vec{v}_3^1$  sowie  $\vec{v}_3^2$ . Dieses Vorgehen ergibt ein gleichstarkes Verschieben korrespondierender Parameterkoordinaten und verhindert dabei entgegengesetzte Bewegungen innerhalb der Koordinatenpaare eines Parameterbereiches. Aufgrund der Mittelung wird auch eine geringe Anfangsverzerrung initialisiert. Der Betrag von einem der beiden Verschiebungsvektoren an einem Schnittkantenpunkt  $v_i$  gibt dabei das Maß für dessen Verzerrung an und wird als Spannungsstärke  $E(v_i) = \|\vec{v}_i\|$  bezeichnet.



**Abb. 4.25:** Veranschaulichung des Relaxationsprozesses. (a) Übertragung der erzeugten Verschiebungsvektoren aus Abb. 4.24 (e) an die jeweiligen Parameterkoordinatenpaare der Schnittkantenpunkte. (b) Neue Position der Koordinatenpaare nach der Verschiebung und daraus resultierende „Spannung“ durch die Beträge der Vektoren. (c)-(d) Abgeschwächte Weitergabe der „Spannung“ am Beispiel von  $v_1^1, v_2^1, v_3^1$  an ihre direkten und indirekten Nachbarpunkte. Deren Verschiebungsvektoren ergeben sich dabei durch Mittelung ihrer direkten, bereits relaxierten Nachbarpunkte (hellgrau). Abgeschwächt werden diese durch einen prozentualen Schwächungsfaktor  $1 - \frac{1}{7}$ .

Die resultierende Gesamtspannung aus allen Schnittkantenpunkten wird nun durch eine iterative Relaxation über die topologische Nachbarschaft verteilt. Die Ausbalancierung erfolgt jeweils nur über die Eckpunkte, welche zum benachbarten Bereich des Paramerraumes gehören. Dies bedeutet somit keine Relaxation über Schnittkanten hinaus, sondern jeweils entgegengesetzt von dieser. Veranschaulicht wird das Vorgehen anhand der Abb. 4.25, wo zunächst in (a) die Dreiecke und Schnittkanten aus Abb. 4.24 noch einmal dargestellt sind. In Richtung der eingezeichneten Verschiebungsvektoren sind die Schnittkantenpunkte in (b) verschoben worden. Die sich ergebene Verzerrung der inzidenten Dreiecke ist dabei deutlich erkennbar. Die Relaxation der Verzerrung wird im Folgenden mit Hilfe der unteren Reihe aus Abb. 4.25 am Beispiel der Parameterkoordinaten  $v_1^1, v_2^1, v_3^1$  erläutert. Es sei darauf hingewiesen, dass für ein besseres Verständnis nur ein Ausschnitt des Relaxationsbereiches dargestellt ist.

Für das Beispiel wurden in (c) die Schnittkanten um eine weitere Kante und dem Eckpunkt  $v_4^1$  mit zugehörigen Dreiecken erweitert. Die Positionen der bereits verschobenen Schnittkantenpunkte werden als „fixiert“ (grau) markiert. Anschließend erfolgt eine topologische Suche nach den direkten Nachbarn, die noch nicht relaxiert, d. h. nicht „fixiert“, sind. Im Beispiel entspricht dies den Eckpunkten  $v_5, v_6, v_7$ . Auf diese wird nun die Spannung an der Schnittkante in abgeschwächter Form weitergereicht. Die Abschwächung erfolgt dabei um einen konstanten Prozentfaktor  $\tau$ . Für die Relaxation wird für die gefundenen Nachbarnpunkte zunächst ihre neue Spannungsstärke bestimmt, die dem um  $1 - \frac{1}{\tau}$  abgeschwächten Betrag ihres Verschiebungsvektors entspricht. Der Vektor setzt sich dabei aus der Mittelung der entsprechenden Verschiebungsvektoren der direkt verbundenen, bereits relaxierten Eckpunkte  $v_j$  zusammen (vgl. (c)):

$$E(v_i) = \left(1 - \frac{1}{\tau}\right) \left\| \frac{1}{n} \sum_{j=1}^n \vec{v}_j \right\|, \text{ mit } \vec{v}_j \in L(V_{fix}). \quad (4.28)$$

Die Liste  $L(V_{fix})$  enthält dabei alle  $n$  Eckpunkte der direkten Nachbarschaft von  $v_i$ , die bereits relaxiert wurden. Anschließend werden die Eckpunkte entsprechend des abgeschwächten Vektors verschoben und als „fixiert“ markiert.

Für den nächsten Relaxationsschritt werden in (d) wieder die direkten, nicht relaxierten Nachbarn gesucht. Für das Beispiel ergeben sich hierfür  $v_8$  und  $v_9$ . Die Bestimmung des Relaxationsvektors, der Spannung  $E(v_i)$  und die Verschiebung erfolgt in der gleichen Weise wie im ersten Relaxationsschritt. Dieser Prozess für einen Eckpunkt setzt sich so lange fort, bis dessen  $E(v_i)$  einen Schwellwert unterschreitet und damit eine Entspannung in der topologischen Nachbarschaft erreicht wurde. Dieser Zustand ist in (e) dargestellt, wo die Spannungen an den Eckpunkten  $v_{10}$  und  $v_{11}$  unter dem gesetzten Schwellwert liegen. Die Eckpunkte werden zwar um den Restbetrag der Spannung verschoben, jedoch bricht nach ihnen der weitere Relaxationsprozess ab. Analog erfolgt die Relaxation für die Parameterkoordinaten  $v_1^2, v_2^2$  und  $v_3^2$  über deren topologische Nachbarn.

## 4.5 Texturierung

Die Abbildung der Textur auf die Oberfläche erfolgt in dieser Arbeit nach dem 2D-*Texture Mapping* Verfahren, wie es im Abschnitt 2.4.1 beschrieben ist. Die erzeugten Parameterkoordinaten jedes Eckpunktes werden direkt von der Graphikkarte auf das Intervall  $[0, 1]$  skaliert und in entsprechende Texturkoordinaten konvertiert.

Für eine Standardtexturierung, wie sie zunächst in den Abbildungen dieses Kapitels gezeigt wurden, ist eine 2D-Schraffierungstextur zu erzeugen. Aufgrund des kontinuierlichen Parameterraumes und dessen Unterteilung in ein Vielfaches des Projektionsfensters  $W$ , muss die Textur ein kachelfähiges Muster aufweisen. Durch herkömmliche Bildverarbeitungs- und Zeichenprogramme lassen sich diese manuell generieren. Dieses Vorgehen lässt sich auch durch eine automatische, prozedurale Generierung erweitern, wo

Varianz des Linienverlaufes und Liniendichte vom Benutzer eingestellt werden können. Die Entwicklung einer solchen Methode soll aber nicht Inhalt dieser Arbeit sein.

#### 4.5.1 Integration der Beleuchtung

Neben der einfachen Texturierung der Strukturen zur Charakterisierung des Oberflächenverlaufes ist eine wichtige Anforderung des Verfahrens die Integration der lokalen Beleuchtung. Dafür wird das *TAM*-Konzept von PRAUN U. A. [2001] verwendet (vgl. Abschnitt 2.3.4). Für deren Einsatz sind verschiedene Helligkeits- und Auflösungstexturen notwendig, die in Form mehrerer *MipMaps* zusammengestellt werden. Der Aufbau dieser Texturen muss für eine *Frame*-kohärente Darstellung dem Aufbau aus Abb. 2.21 (Seite 29) entsprechen. Eine automatische Erzeugung dieser 2D-Texturen wird dazu ebenfalls in PRAUN U. A. [2001] präsentiert.

Zur Überblendung der Helligkeitstexturen an einem Eckpunkt (vgl. Abb. 2.22 auf Seite 30) ist noch die Ermittlung seiner lokalen Intensität erforderlich. Bestimmt wird diese über den diffusen Beleuchtungsanteil  $I_{diff}$ :

$$I_{diff} = \hat{n} \cdot \hat{l}, \quad (4.29)$$

zwischen Eckpunktnormale  $\hat{n}$  und Lichtquelle  $\hat{l}$ . Je nach Anzahl der Helligkeitstexturen werden entsprechende Schwellwerte definiert, die zwischen den Texturen auswählen.

Die Auswahl und Überblendung der *MipMap*-Stufen wird von moderner Graphikhardware automatisch realisiert. Berechnet wird hierbei die Größe der projizierten Fläche des Objektes auf den Bildschirm. Je nachdem, wie viele *Texel* ein *Pixel* dabei überdeckt, wird die entsprechende Stufe der *MipMap*-Textur ausgewählt (*Level of Detail*). Besitzen zwei benachbarte Oberflächenpunkte unterschiedliche Auflösungsstufen, so wird zwischen diesen linear interpoliert (vgl. Abb. 2.21).

Die Bestimmung der lokalen Beleuchtung sowie Auswahl und Interpolation der Texturen erfolgt wie bei PRAUN U. A. [2001] mittels *Vertex* und *Fragment Shader*. Damit wird eine echtzeitfähige, texturierte Darstellung der Oberfläche sichergestellt.

## 4.6 Zusammenfassung

Es wurde ein texturbasiertes Verfahren entwickelt, welches Schraffierung auf patientenspezifischen, anatomischen Strukturen erzeugt. Die Schraffierung basiert zum einen auf den Anforderungen an eine interaktive Visualisierung und zum anderen auf einen korrekten Linienverlauf, entsprechend den Beobachtungen aus anatomischen Illustrationen. Der Linienverlauf orientiert sich dabei an einem krümmungs- und modellbasierten Vektorfeld. Für jeden Oberflächenpunkt werden dazu Krümmungsinformationen mit Informationen

über eine gegebenenfalls vorhandene, modellspezifische Vorzugsrichtung kombiniert. Letztere existieren für Gefäß- und Muskelstrukturen.

Die erforderliche Parametrisierung der Oberfläche geschieht anhand von sich überlappenden *Patches* nach der Methode von PRAUN U. A. [2000]. Die Patchdreiecke werden dazu im Parameterraum anhand des zugrunde liegende Vektorfeldes ausgerichtet, damit die spätere Texturierung dem vorgegebenen Linienverlauf entspricht. Vorteil dieser Art von Parametrisierung ist ein minimierte Texturverzerrung und lokale Beeinflussung der Texturausrichtung. Nachteilig sind die entstehenden Texturnähte an den Patchübergängen. Daher wurde der Ansatz erweitert und eine entsprechende Behandlung dieser Übergänge entwickelt, so dass keine Texturnähte mehr auftreten. Die Behandlung beinhaltet zum einen eine Mittelung von Parameterkoordinaten für *Patches*, die im Objekt- und Parameterraum benachbart sind. Zum anderen ist ein Relaxationsprozess für Dreiecke entwickelt worden, welche miteinander Schnittkanten bilden. Diese beschreiben einen Zyklus im Patchwachstum und sind charakterisiert durch mehr als eine Parameterkoordinate pro Eckpunkt, welche zu weit im Parameterraum auseinander liegen. Texturverzerrungen sind zusätzlich minimiert worden, indem die Patchdreiecke durch affine Transformationen, anstatt von Projektion, in den Parameterraum überführt wurden.

Die Generierung des Vektorfeldes und die Oberflächenparametrisierung erfolgen in einem einmaligen Vorverarbeitungsschritt für jede Oberfläche. Das Ergebnis kann somit für jede erneute Texturierung wieder verwendet werden. Für die beleuchtungsabhängige Schraffierung wird auf das *TAM*-Konzept zurückgegriffen, um eine *Frame*-Kohärenz in Helligkeit und Skalierbarkeit innerhalb einer interaktiven Visualisierung sicher zu stellen.



## 5 Implementierung

Dieses Kapitel beschreibt die Umsetzung des entwickelten Verfahrens zur Schraffierung triangulierter, anatomischer Oberflächenmodelle. Zunächst erfolgt eine kurze Vorstellung der verwendeten Entwicklungswerkzeuge und Bibliotheken. Hierzu gehört insbesondere die *Wing Edge Mesh* (*WEM*)-Datenstruktur zur effizienten Verwaltung der Dreiecksgeometrie sowie die Bibliothek von SCHULZ [2005] zur Krümmungsapproximation. Anschließend wird auf die Umsetzung der Vektorfelderzeugung und Oberflächenparametrisierung sowie den dazu entwickelten Datenstrukturen eingegangen. Abschließend erfolgt die Implementierung der Texturabbildung.

### 5.1 Entwicklungswerkzeuge und verwendete Bibliotheken

Das umgesetzte Verfahren wurde auf einem INTEL PENTIUM 4 Prozessor mit 3.0 GHz, 2 GB Arbeitsspeicher und einer ATI Radion 9600 Graphikkarte entwickelt. Als Entwicklungsumgebung diente die Bildverarbeitung- und Visualisierungsplattform MEVISLAB. MEVISLAB integriert eine Vielzahl von Bibliotheken, die bei Bedarf dynamisch zur Laufzeit eingebunden werden können. Die für diese Arbeit verwendeten Bibliotheken und MEVISLAB sollen im Folgenden kurz vorgestellt werden.

#### 5.1.1 MeVisLab

MEVISLAB (HAHN U. A. [2003]) ist eine Bildverarbeitung- und Visualisierungsplattform des Bremer Forschungszentrums MEVIS RESEARCH<sup>1</sup>. Es bietet eine *Prototyping*-Umgebung zur Erstellung komplexer Anwendungen für die medizinische Bildverarbeitung und Visualisierung. Mit Hilfe von einzelnen Modulen, die eine große Anzahl von Bildverarbeitungsalgorithmen und Visualisierungswerkzeugen zur Verfügung stellen, lassen sich Netzwerke zur Lösung verschiedener medizinischer Problemstellungen zusammenstellen. Komplexe Netzwerke lassen sich zu Makromodulen zusammenfassen, welche wiederum als Module in andere Netzwerke integriert werden können. Ein wichtiges Konzept ist dabei die Erzeugung von einzelnen Feldverbindungen zwischen den Modulen, worüber der Datenfluss verläuft. Des Weiteren besteht die Möglichkeit, neue Module mit eigenen Algorithmen bzw. Bildverarbeitungs- und Visualisierungstechniken zu implementieren und in MEVISLAB zu integrieren. Die Programmierung erfolgt dazu in C++ und

---

<sup>1</sup>[www.mevislab.de](http://www.mevislab.de) bzw. [www.mevis.de](http://www.mevis.de)

es können dabei auf MEVISLAB interne Datenstrukturen und bereits vorhandene Algorithmen zurückgegriffen werden.

### 5.1.2 Verwendete Bibliotheken

In diesem Abschnitt sollen die verwendeten Bibliotheken OPEN INVENTOR, MLWEM, MLWEMSURFACECURVATURE, MLGRAPH sowie SOSHADER kurz vorgestellt werden.

#### Open Inventor

OPEN INVENTOR (WERNECKE [1993]) ist eine frei erhältliche, objektorientierte 3D-Graphikbibliothek von Silicon Graphics (SGI), basierend auf OpenGL. Der Fokus liegt dabei auf der Erstellung komplexer, echtzeitfähiger 3D-Szenen, die vom Benutzer interaktiv exploriert und manipuliert werden können. Dazu bietet OPEN INVENTOR eine große Anzahl definierter Objekte in Form von einzelnen Knoten, die zu einem Szenengraphen zusammengesetzt werden. Des Weiteren bietet OPEN INVENTOR eine Vielzahl von Klassen an, beispielsweise für Vektoren und Matrizen, mit entsprechenden Operationen wie Matrizenmultiplikationen. Zusätzlich lassen sich neue Funktionalitäten und Objekte erzeugen, die den bestehenden Umfang erweitern. Die Programmierung erfolgt dabei ebenfalls in C++. Die Graphikbibliothek ist in MEVISLAB integriert, wodurch bestehende Module mit den Knoten aus OPEN INVENTOR kombiniert sowie für die eigene Modulentwicklung genutzt werden können.

Für diese Arbeit wurde speziell ein `SoTriangleStripSet`-Knoten verwendet, welchem neben Farbe und Position für jeden Dreieckseckpunkt auch Parameterkoordinaten für die Texturierung übergeben werden können (vgl. Abschnitt 5.4.1).

#### MLWEM

Die Umsetzung des entwickelten Verfahrens aus Kapitel 4 setzt einen direkten Zugriff auf die einzelnen Dreiecksprimitive Punkt, Kante und Fläche sowie deren inzidente Nachbarn voraus. Zur effizienten Verwaltung dieser Geometriedaten wurde die *WEM*-Datenstruktur (BAUMGART [1972]) verwendet. Diese enthält neben allen Dreiecksprimitiven einer Oberfläche auch deren direkte, topologische Nachbarn. Anhand dieser Zusammenhangsinformationen ist ein schneller Zugriff auf einzelne Primitive und deren *k*-Nachbarschaft möglich. Da die verwendeten anatomischen Oberflächen in einem OPEN INVENTOR spezifischen Format vorliegen, müssen diese erst in die *WEM*-Struktur überführt werden. Dies erfordert eine zusätzliche Vorverarbeitungszeit, die abhängig ist von der Anzahl der Oberflächendreiecke. Eine in der Oberfläche enthaltene Nicht-2-Mannigfaltigkeit (vgl. Abschnitt 3.2 auf Seite 47) wird dabei von der *WEM*-Struktur

übernommen und muss bei der Traversierung über die Dreiecksprimitive berücksichtigt werden.

Die *WEM*-Datenstruktur ist als Bibliothek *MLWEM* in *MEVISLAB* integriert und bietet verschiedene Werkzeuge für die *WEM*-Generierung und deren Weiterverarbeitung. Die Weiterverarbeitung erfolgt in Form von vorhandenen *MEVISLAB*-Modulen, von denen einige in dieser Arbeit verwendet wurden. So existiert beispielsweise ein Konvertermodul, welches zu einem *OPEN INVENTOR*-Objekt eine *WEM*-Datenstruktur erzeugt. Dazu wird für jeden Eckpunkt ein *WEMNode*, jeder Kante eine *WEMEdge* und jeder Fläche ein *WEMFace* generiert. Jedes dieser Dreiecksprimitive besitzt dabei Informationen über seine direkte Nachbarschaft. Dies erlaubt eine effiziente Traversierung über eine *k*-Nachbarschaft. Für diese Arbeit wurde das Konvertermodul überarbeitet, da dieses einen quadratischen Aufwand für die Umwandlung benötigte. In der modifizierten Variante wurde die 3D-Szene in Würfel unterteilt, so dass die lokale Zuordnung einzelner Dreiecke aus der originalen *OPEN INVENTOR*-Szene zu der bereits konvertierten *WEM*-Oberfläche beschleunigt wird. Der Aufwand reduzierte sich dadurch auf  $O(n \log n)$ , wobei  $n$  für die Anzahl der zu konvertierenden Oberflächendreiecke steht.

Ferner sind folgende *WEM*-Module verwendet wurden:

- *WEMSmooth* zur Oberflächenglättung sowie
- *WEMCollapseEdges* zur Geometrievereinfachung.

Die Glättung der Positionen und Normalen aller *WEMNode* erfolgt dabei anhand der Laplace-Glättung von Abschnitt 4.3.1. Einzustellen ist dabei ebenfalls die Anzahl der Iterationsschritte und der Wichtungsfaktor  $w$ .

### **MLWEMSurfaceCurvature**

Die in Abschnitt 3.2 vorgestellten Operatoren zur Krümmungsapproximationen sind in der Bibliothek *MLWEMSURFACECURVATURE* von SCHULZ [2005] realisiert. Das dabei entwickelte *MEVISLAB*-Modul *WEMCurvature* stellt die Auswahl der Krümmungsoperatoren zur Verfügung und ermittelt auf Basis einer *WEM*-Datenstruktur die entsprechenden Krümmungsinformationen der Oberfläche. Neben dem topologischen Radius  $r$  zur Beschreibung von  $nhd(v_i)$  können diverse Parameter zur Glättung der erzeugten Hauptkrümmungsrichtungen eingestellt werden. Hierbei zählen vor allem die Glättungsparameter  $\delta_{ratio}$  und  $\delta_{reliable}$  (vgl. Gleichung 3.9 auf Seite 48) zur Bestimmung der parabolischen Bereiche.

## MLGraph

Die Bibliothek MLGRAPH dient zur Extraktion, Analyse und Darstellung von Gefäßstrukturen auf Basis von segmentierten, radiologischen 2D-Schichtbildern nach der Methode von SELLE U. A. [2000]. In dieser Arbeit wurde die Bibliothek verwendet, um zum einen die Skelettlinien aus den Blutgefäßen der zur Verfügung stehenden Datensätze zu extrahieren. Zum anderen diente sie zum Zugriff auf alle Kanten eines erzeugten Gefäßskelettes sowie deren einzelne Skelettvoxel. Das extrahierte Skelett wird in Form eines Graphen in einer XML-Datei abgespeichert. Dieser enthält für jeden neuen Gefäßzweig ein `Node`-Objekt und dazu die Anzahl angrenzender Kanten, welche als `Edge`-Objekte deklariert sind. Eine Bifurkation ist beispielsweise durch drei inzidente Kanten gekennzeichnet, die sich aus einer eingehenden und zwei ausgehenden Kanten ergeben. Durch entsprechende Zugriffsfunktionen auf jedes eindeutige `Node`- und `Edge`-Objekt erfolgt eine Traversierung durch den Graphen.

## SoShader

Im Abschnitt 2.3.4 wurde die Schraffierung mittels *TAM*-Texturen von PRAUN U. A. [2001] erläutert. Dieses Konzept basiert auf der Verwendung von *Hardware Shadern* (CARD UND MITCHELL [2002]), welche durch Module der Klassenbibliothek SOSHADER integriert werden. Unterstützt wird hierbei die *OpenGL Shading Language* (ROST [2004]). Von den Modulen wurden in der vorliegenden Arbeit hauptsächlich folgende verwendet:

- `SoVertexShader` und `SoFragmentShader` zur *Vertex*- und *Fragment Shader*-Programmierung,
- `SoShaderProgram` für die korrekte Verlinkung der beiden ersten Module sowie
- `SoSampler2D` für den Zugriff auf eine Bildtextur.

Der *Vertex Shader* manipuliert dabei die Eckpunkte einer Oberfläche, welche in Form einer OPEN INVENTOR-Szene vorliegen muss.

## 5.2 Erzeugung des Vektorfeldes

Im Folgenden wird näher auf die Erzeugung des Vektorfeldes eingegangen. Betrachtet werden soll zunächst die Realisierung der Vorzugsrichtung für die Gefäß- und Muskelstrukturen. Bei Organen und Knochenstrukturen entfällt dies, da hier das Vektorfeld aus der geglätteten ersten Hauptkrümmungsrichtung besteht. Anschließend erfolgt die Umsetzung der adaptiven Krümmungsapproximation sowie eine Übersicht der entwickelten Klassen für die Erzeugung des Vektorfeldes.

### 5.2.1 Vorzugsrichtung bei Gefäßstrukturen

Die Bestimmung der Vorzugsrichtung  $\hat{r}_v$  erfordert das Finden der am nächsten gelegenen, korrespondierenden Schnittebene  $\epsilon(s_i)$ . „Korrespondierend“ bedeutet dabei, dass der Skelettvoxel  $s_i$  auf der Gefäßkante liegt, die  $v$  am nächsten liegt (vgl. Abschnitt 4.3.4). Dazu müsste jeder Eckpunkt der Oberfläche mit allen Schnittebenen verglichen werden. Für  $n$  Eckpunkte und  $m$  Ebenen ergibt sich damit ein Aufwand von  $O(nm)$ . Dabei muss jeweils noch getestet werden, ob die Schnittebene zu dem Eckpunkt korrespondiert.

In einem Vorverarbeitungsschritt wird daher die zu überprüfende Anzahl der Eckpunkte pro Schnittebene eingeschränkt, um den Aufwand zu verringern. Die achsenorientierte *Bounding Box* der OPEN INVENTOR-Szene wird dazu in gleichmäßige Würfel unterteilt. Jeder Eckpunkt wird anschließend durch Positionsvergleich mit den Würfelgrenzen einem von diesem zugeordnet. Weiterhin werden mit Hilfe von Gleichung 4.11 zu jeder Ebene die Gefäßkanten gespeichert, die diese schneidet. Als effiziente Datenstrukturen sind dafür *maps* aus der *STL* von C++ verwendet wurden. Der *Key* entspricht dabei jeweils dem Würfel- oder Ebenenindex und der *Value* einem Vektor mit den entsprechenden *WEMNode* der *WEM*-Struktur bzw. *Edge*-Objekten der Skelettlinie.

Die Bestimmung der korrespondierenden Schnittebene zu jedem Eckpunkt erfolgt über die Iteration der erzeugten Würfel  $K_i$ . Zunächst werden für jeden Würfel die Schnittebenen bestimmt, die diesen schneiden. Diese Suche erfolgt durch eine vorzeichenbehaftete Distanzbestimmung (vgl. Gleichung 4.13) der Würfeleckenpunkte zu den Ebenen. Ist das Vorzeichen zu mindestens einer Würfecke verschieden, so schneidet die Ebene den Würfel. Im Anschluss werden für jeden Eckpunkt  $v \in K_i$  nur die geschnittenen Ebenen betrachtet. Davon werden wiederum nur diejenigen weiter untersucht, die zur korrespondierenden Gefäßkante von  $v$  gehören. Der Test dafür erfolgt anhand der Ausführungen zu Gleichung 4.12 (Seite 78). Zu den daraus übrig gebliebenen Ebenen wird anschließend diejenige Schnittebene  $\epsilon(s_i)$  bestimmt, die den geringsten Abstand zu  $v$  hat.

Für die im Entwurf beschriebene Interpolation des aus  $\epsilon(s_i)$  resultierenden Richtungsvektors  $\hat{r}_v$  wird noch die benachbarte Schnittebene  $\epsilon(s_j)$  gesucht, welche  $v$  mit  $\epsilon(s_i)$  einschließt. Um zu entscheiden, ob  $\epsilon(s_j)$  die Nachfolge- oder Vorgängerebene von  $\epsilon(s_i)$  ist, wird wiederum durch eine vorzeichenbehaftete Distanz (vgl. Gleichung 4.13) überprüft, auf welcher Seite  $v$  bezüglich  $\epsilon(s_i)$  liegt.

### 5.2.2 Vorzugsrichtung bei Muskelstrukturen

Zur Bestimmung der Kontraktions- und Dehnungsrichtung bei Muskelstrukturen wurde die *Oriented Bounding Box* (*OBB*) mit Hilfe des Modules *UMDSOMainAxis*-Knoten generiert. Die dabei ermittelten drei Hauptachsen wurden über Feldverbindungen für die eigene Umsetzung übernommen und bezüglich ihrer Länge analysiert. In dieser Arbeit wurde dazu der elongierte Muskeltyp implementiert, da nur hierfür anatomische Datensätze zur Verfügung standen. Im Abschnitt 4.3.4 auf Seite 80 wurde jedoch auch

ein Konzept vorgestellt, womit die Vorzugsrichtung des federförmigen Muskeltyps aus der *OBB* ermittelt werden kann.

### 5.2.3 Approximation fehlender Krümmungsinformation

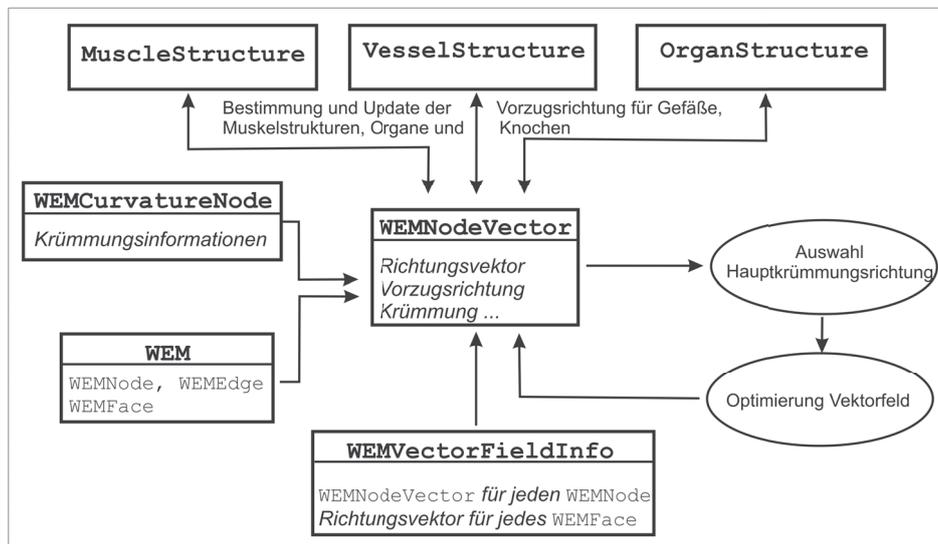
An Eckpunkten, die entweder durch eine degenerierte Umgebung gekennzeichnet sind oder deren Anzahl direkter Nachbarnpunkte für das gewählte Approximationsverfahren nicht ausreichen, können die von SCHULZ [2005] implementierten Verfahren keine Krümmungsinformationen bestimmen. Daher wurden im Abschnitt 4.3.2 zwei Möglichkeiten aufgezeigt, wie die fehlenden Informationen ermittelt werden können. Dafür ist es zunächst notwendig, die betroffenen Eckpunkte separat zu extrahieren. In dem Modul `WEMCurvature` werden bereits die degenerierten Eckpunkte gefunden, jedoch nicht weiter behandelt. Für diese Arbeit werden diese in einer `vector`-Struktur der *STL* von C++ abgelegt. Ebenso werden auch die Eckpunkte separat abgespeichert, deren direkte Nachbarschaft für das Approximationsverfahren nicht ausreicht. Dieser Test erfolgt nachdem alle degenerierten Eckpunkte bereits extrahiert wurden. Damit wird sichergestellt, dass die Anfrage der inzidenten `WEMNode`-Objekte der *WEM*-Struktur sich nur auf 2-mannigfaltige Eckpunkte bezieht. Des Weiteren erfolgt der Test nur bei den Operatoren *ParametricOP* und *BiQuadBezOP*, da diese mehr als drei direkte Nachbarnknoten benötigen.

Das Modul `WEMCurvature` wurde anschließend durch die beiden Möglichkeiten erweitert die fehlenden Krümmungsinformationen zu interpolieren oder adaptiv zu bestimmen. Der Interpolationsansatz erfolgt dabei anhand der Gleichungen 4.3 und 4.4. Dabei wird darauf geachtet, dass die zur Mittelung verwendeten Nachbarnpunkte selbst gültige Krümmungsinformationen besitzen. Für den adaptiven Ansatz wird der Vektor mit den Eckpunkten, deren direkte Nachbaranzahl nicht ausreicht, dem gewählten `TaubinOP`-Operator übergeben. Die Bestimmung der Krümmungsinformationen an den degenerierten Eckpunkten erfolgt nach dem Interpolationsansatz.

### 5.2.4 Datenstrukturen

Für die Generierung des Vektorfeldes sind verschiedene Klassen implementiert worden, die einen modularen und übersichtlichen Aufbau des Vektorfeldes ermöglichen. Ihre Beziehungen untereinander sind in Form eines Klassendiagrammes in der Abb. 5.1 dargestellt. Alle Informationen über das erzeugte Vektorfeld werden durch eine Instanz der Klasse `WEMVectorFieldInfo` repräsentiert. Für jeden *WEM*-Eckpunkt wird darin ein `WEMNodeVector`-Objekt erzeugt, das u. a. Informationen über die Krümmungsinformationen, den Richtungsvektor, der Vorzugsrichtung sowie Tangentialebene beinhaltet. Die Krümmungsinformationen erhält jeder `WEMNodeVector` durch ein korrespondierendes `WEMCurvatureNode`-Objekt aus dem Krümmungsmodul von SCHULZ [2005]. Die Vorzugsrichtung erhalten alle `WEMNodeVector`-Objekte durch Instanz von einer der Klassen

MuscleStructure, VesselStructure und OrganStructure, abhängig von der zu schraffierenden Struktur. Zu letzteren gehören auch die Knochenstrukturen. Zusätzlich ist in WEMVectorFieldInfo für jedes WEMFace der gemittelte Richtungsvektor aus deren Eckpunkten enthalten.



**Abb. 5.1:** Klassendiagramm der entwickelten Datenstrukturen zur Erzeugung des krümmungs- und modellbasierten Vektorfeldes. Jeder *WEM*-Knoten wird dabei durch ein *WEMNodeVector* repräsentiert, in dem u. a. sein Richtungsvektor und seine lokale Krümmung gespeichert ist. Die Vorzugsrichtung wird in einer Instanz der Klassen *MuscleStructure*, *VesselStructure* sowie *OrganStructure* ermittelt und dem *WEMNodeVector* übergeben. Aus den Krümmungsinformationen und der Vorzugsrichtung erfolgt die initiale Erzeugung des Vektorfeldes mit anschließender Optimierung der ermittelten Richtungsvektoren. Alle *WEMNodeVector*-Objekte sind dazu in der Instanz der Klasse *WEMVectorFieldInfo* abgelegt, welche an das Parametrisierungsmodul übergeben wird.

In der Klasse `OrganStructure` wird für jeden `WEMNodeVector` dessen erste Hauptkrümmungsrichtung als Vorzugsrichtung definiert. Anschließend wird das resultierende Vektorfeld über eine  $k$ -Nachbarschaft geglättet. Für Muskelstrukturen ist in der Klasse `MuscleStructure` momentan nur der elongierte Muskeltyp aus Abschnitt 4.3.4 umgesetzt. Hier wird die längste der drei Hauptachsen bestimmt, welche aus dem Modul `UMDSMain-Axis` stammen, in die Tangentialebene der jeweiligen `WEMNodeVector`-Objekte projiziert und anschließend normiert. Für Gefäße erfolgt die Laplace-Glättung des Gefäßskelettes nach Gleichung 4.5 und Bestimmung des radialen Richtungsvektors  $\hat{r}_v$  für jeden Eckpunkt in der Klasse `VesselStructure`.

Die in einem `WEMNodeVector` gespeicherten Krümmungsinformationen und Vorzugsrichtungen werden anschließend in dem Modul `UMDVectorField` zusammengeführt (vgl. Abschnitt 4.3.5). Dazu wird zunächst die Hauptkrümmungsrichtung eines `WEMNodeVector`-Objektes ausgewählt, welche der darin enthaltenen Vorzugsrichtung am ähnlichsten ist. Anschließend erfolgt die iterative Optimierung des initialen Vektorfeldes anhand von Gleichung 4.17.

## 5.3 Oberflächenparametrisierung

Dieser Abschnitt erläutert die Umsetzung der Oberflächenparametrisierung durch die Verwendung von *Patches*. Hierfür erfolgt eine nähere Betrachtung wie die erzeugten Parameterkoordinaten pro Eckpunkt verwaltet werden. Aus der daraus erzeugten Informationsbasis erfolgt abschließend die korrekte Zuweisung einer Parameterkoordinate für jeden Eckpunkt eines Oberflächendreieckes. Anzumerken ist dabei, dass bezüglich der Behandlung der Patchübergänge bisher nur der Fall 1: „Patches im Objekt- und Parameterraum sind inzident“ implementiert ist (vgl. Abschnitt 4.4.4 auf Seite 100). Abschließend werden auch hier die erzeugten Datenstrukturen in Form eines Klassendiagrammes vorgestellt.

### 5.3.1 Verwaltung Parameterkoordinaten

Aufgrund der überlagerten *Patches* und der daraus mehrfachen Parametrisierung einzelner Dreiecke, existieren für die entsprechenden Eckpunkte auch verschiedene Parameterkoordinaten. Wichtig für die Behandlung der Patchübergänge ist, dabei zu unterscheiden, welcher Fall aus Abschnitt 4.4.4 betrachtet werden muss. Das Kriterium dafür ist die Lage der verschiedenen Parameterkoordinaten eines Eckpunktes im Parameterraum. Liegen diese zu weit auseinander, so gehört der Eckpunkt zu einer Schnittkante, also einer Kante, bei der das Patchwachstum einen Zyklus erreicht hat. Das Maß für die maximale Entfernung ist dabei die Größe des Projektionsfensters.

Um die eingangs erwähnte Fallunterscheidung zu treffen, werden zunächst alle Parameterkoordinaten in einer globalen Liste  $L_{Coordinates}$  abgelegt. Auf dieser erfolgt anschließend

eine Partitionierung in verschiedene *Cluster*, d. h. Zusammenfassung einzelner Parameterkoordinaten zu relativen Bereichen im Parameterraum. Die erzeugten *Cluster* werden ebenfalls für jeden Eckpunkt in einer Liste  $L_{Cluster}$  abgespeichert. Der Aufbau dieser Listen ist in Abb. 5.2 (a) und (b) beispielhaft veranschaulicht. Innerhalb eines Eckpunktes sind in (a) zunächst alle Koordinaten aufgelistet und gleich markiert, wenn ihr Abstand zueinander kleiner als die Breite des Projektionsfensters ist. Im Beispiel beträgt diese 1.0. Die so markierten Koordinaten werden in (b) zu einzelnen *Clustern* zusammengefasst und jeweils mit A, B usw. bezeichnet. Jeder *Cluster* enthält dabei die aufsummierten, ähnlichen Parameterkoordinaten  $(\bar{x}, \bar{y})$  sowie deren Anzahl („#“). Erkennbar ist, dass pro Eckpunkt mindestens ein *Cluster* erzeugt wird. Eine Analyse bezüglich der Clusteranzahl pro Eckpunkt gibt den Hinweis darauf, ob dieser zu einer Schnittkante gehört oder nicht. Dies ist der Fall, wenn die Anzahl größer eins ist. Im gezeigten Beispiel ist dies für die Eckpunkte 0, 1, 4 und 5 der Fall.

$\mathcal{L}_{Coordinates}$		$\mathcal{L}_{Cluster}$		$\mathcal{L}_{PatchFaces}$			
ID <sub>Eckpunkt</sub>	Parameterkoordinaten (x,y)	ID <sub>Eckpunkt</sub>	Cluster ID $[\bar{x}, \bar{y}, \#]$	ID <sub>PatchFace</sub>	$v_1(x,y)$	$v_2(x,y)$	$v_3(x,y)$
0	(0.2,1.0) (0.17,0.95) (-2.1,1.2)	0	A[0.37,1.95,2] B[-2.1,1.2,1]	0[0,1,2]	(0.19,0.98) (0.39,0.71) (0.86,0.88)		
1	(0.39,0.71) (-1.7,0.7)	1	A[0.39,0.71,1] B[-1.7,0.7,1]	1[1,2,3]	(0.39,0.71) (0.86,0.88) (0.54,0.31)		
2	(0.87,0.92) (0.8,0.83) (0.92,0.89)	2	A[2.59,2.64,3]	2[1,3,4]	(0.39,0.71) (0.54,0.31) (0.2,0.09)		
3	(0.52,0.3) (0.55,0.32)	3	A[1.07,0.62,2]	3[1,4,7]	(-1.7,0.7) (-1.85,0.2) (-2.4,-0.12)		
4	(0.2,0.1) (0.2,0.08) (-1.9,0.2) (-1.8,0.2)	4	A[(0.4,0.18,2] B[-3.7,0.4,2]	4[0,1,7]	(-2.1,1.2) (-1.7,0.7) (-2.4,-0.12)		
5	(0.27,-0.1) (0.31,-0.11) (-1.8,-0.16)	5	A[0.58,-0.21,2] B[-1.8,-0.16,1]	5[7,4,6]	(-2.4,-0.12) (-1.85,0.2) (-1.8,-0.16)		
6	(-2.0,-0.15) (-1.9,-0.17)	6	A[-3.9,-0.32,2]	6[4,5,6]	(-2.0,-0.15) (-1.9,-0.17) (-1.8,-0.16)		
7	(-2.4,-0.12)	7	A[-2.4,-0.12,1]				
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
n	.	n	.	m	.	.	.

**Abb. 5.2:** Verwaltung und Zuweisung von Parameterkoordinaten für jeden Eckpunkt. (a) Die Liste  $L_{Coordinates}$  speichert während der Patcherzeugung alle erzeugten Koordinaten zu den entsprechenden Eckpunkten. (b) Anschließend erfolgt eine Gruppierung dieser Koordinaten zu *Clustern* (A, B) entsprechend eines maximalen zulässigen Abstandes zueinander. Die *Cluster* werden in einer weiteren Liste  $L_{Cluster}$  verwaltet. (c) Die Eckpunkte eines Patchdreieckes erhalten die gemittelten Parameterkoordinaten aus den *Clustern*, die untereinander ähnlich sind. Am Beispiel für zwei Dreiecke in (b) und (c) jeweils durch grau und schwarz hervorgehoben.

### 5.3.2 Zuweisung der Parameterkoordinaten

Nachdem alle *Patches* erzeugt wurden, erhält jeder Eckpunkt eines Dreieckes, basierend auf den erzeugten *Clustern*, sein letztendliches Parameterkoordinatenpaar. Dazu wird über jedes Oberflächendreieck iteriert und bezüglich seiner Eckpunkte in der Liste  $L_{Cluster}$  überprüft, welche *Cluster* unter den Eckpunkten ähnlich sind. Kriterium dafür ist der Vergleich der gemittelten Positionen in den *Clustern* eines Eckpunktes zu den ebenfalls gemittelten Koordinaten aus den *Clustern* der anderen beiden Eckpunkte. Liegt

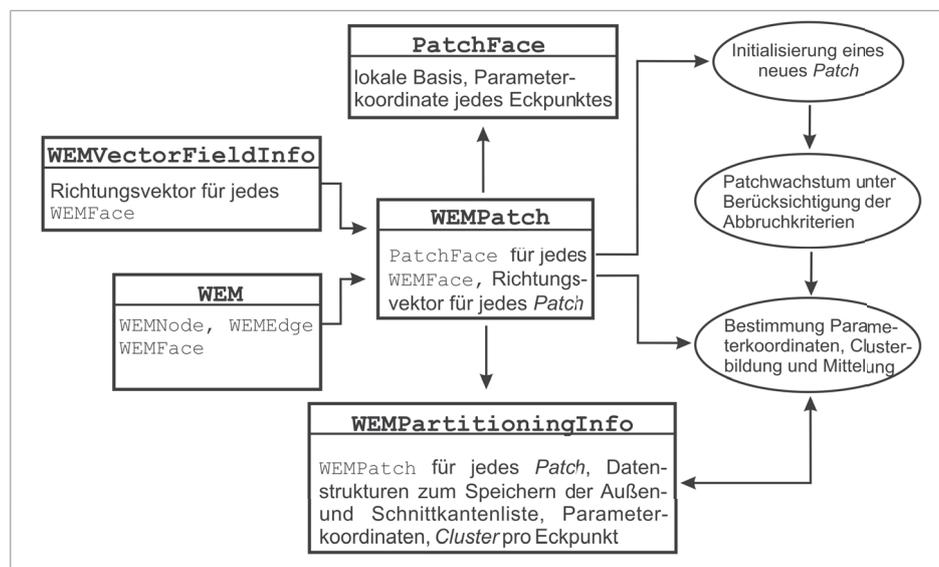
der Abstand unterhalb der Breite des Projektionsfensters, so entsprechen die gefundenen *Cluster* dem gleichen Zusammenhangsgebiet im Parameterraum. Veranschaulicht ist dies in der Abb. 5.2 (c) in Kombination mit (b). Dargestellt ist eine Liste  $L_{PatchFaces}$ , die alle Dreiecke eines *Patch* enthält und für deren Eckpunkte jeweils die gemittelte Parameterkoordinate aus den korrelierenden *Clustern* speichert. Die Eckpunkte stehen dazu für jedes Dreieck in eckigen Klammern. Beispielhaft sind dazu die Dreiecke 0 und 5 hervorgehoben. In (b) sind zu diesen ihre Eckpunkte und ihre jeweils korrelierenden *Cluster* ebenfalls markiert.

Die Mittelung und Zuweisung der Parameterkoordinaten in den gefundenen *Clustern* entspricht dem Fall 1 aus Abschnitt 4.4.4. Dabei entstehen sichtbare Texturnähte, wenn das Patchwachstum einen Zyklus beschreibt und die Eckpunkte der Schnittkante dadurch mehr als einen *Cluster* besitzen (vgl. Abb. 4.23 (b)). Diese sind dem Fall 2 aus Abschnitt 4.4.4 zuzuordnen. Eine korrelierende Übereinstimmung der verschiedenen *Cluster* eines Eckpunktes ist mit dem darin beschriebenen Relaxationsansatz möglich.

### 5.3.3 Datenstrukturen

Für die Parametrisierung der Oberfläche sind ebenfalls verschiedene Datenstrukturen erstellt worden, die die erzeugten *Patches* und Parameterkoordinaten verwalten. Die Abb. 5.3 zeigt dazu eine Übersicht der verwendeten Klassen, analog zur Generierung des Vektorfeldes. Die erzeugten *Patches* werden durch Instanzen der Klasse `WEMPatch` repräsentiert. In diesen werden die hinzugefügten `WEMFace`-Objekte aus der *WEM*-Datenstruktur als Instanzen der Klasse `PatchFace` abgespeichert. Weiterhin erhält jedes `WEMPatch` den korrespondierenden Richtungsvektor zum Referenzdreieck aus dem erzeugten `WEMVectorFieldInfo`. Mit Hilfe dieses Vektors wird zu jedem `PatchFace` eine lokale Basis aufgestellt, durch welche es anschließend im Parameterraum ausgerichtet und transformiert wird. Zur Initialisierung eines *Patch* gehört die Erzeugung des Startpatch sowie das Hinzufügen von neuen *Patches* durch Referenzdreiecke aus der Außenkonturliste bzw. durch daran angrenzende Dreiecke. Letzteres stellt den Spezialfall aus Abschnitt 4.4.3 dar. Das anschließende Patchwachstum fügt die an das Referenzdreieck angrenzende direkten und indirekten Dreiecke dem `WEMPatch` unter Berücksichtigung der Abbruchkriterien hinzu (vgl. Abschnitt 4.4.2). Das Vorgehen für die Überführung in die  $x$ - $y$ -Ebene, als den Parameterraum, entspricht den Ausführungen aus Abschnitt 4.4.1 (Startpatch) bzw. 4.4.3 (weitere *Patches*).

Nachdem ein Patchwachstum abgeschlossen ist, werden für jeden enthaltenen Eckpunkt in einem `WEMPatch` die Parameterkoordinaten bestimmt. Letztere und das erzeugte *Patch* werden anschließend in die globale Datenstruktur `WEMPartitioning` abgespeichert. Bezüglich der Koordinaten existieren darin zwei Listen. Die eine enthält zu jedem Eckpunkt alle seine ermittelten Parameterkoordinaten aus allen *Patches*, in denen dieser enthalten ist. Die zweite Liste speichert zu jedem Eckpunkt *Cluster* seiner Koordinaten, zu denen



**Abb. 5.3:** Klassendiagramm der entwickelten Datenstrukturen für die Parametrisierung der Oberfläche durch überlagerte *Patches*. Jedes *Patch* wird dabei durch eine Instanz der Klasse `WEMPatch` repräsentiert. Darin sind neben den zugehörigen *WEM*-Dreiecken in Form von `PatchFace`-Objekten, der Richtungsvektor zur Ausrichtung des *Patch* im Parameterraum enthalten. Diesen erhält das Referenzdreieck aus der `WEMVectorFieldInfo`-Struktur. Alle generierten *Patches* werden in der Struktur `WEMPartitioning` abgespeichert, in der auch die Zuweisung der korrekten Parameterkoordinat zu jedem Eckpunkt eines `PatchFace` erfolgt.

sie gruppiert sind. Die Clusterbildung fasst ähnliche Parameterkoordinaten zusammen und ist für die spätere Behandlung der Patchübergänge erforderlich.

Abschließend erfolgt innerhalb von `WEMPartitioning` für jeden Eckpunkt eines `PatchFace` die Mittelung der Koordinaten korrelierender *Cluster*. Die gemittelte Koordinat wird anschließend dem Eckpunkt des `PatchFace` zugewiesen.

## 5.4 Texturierung

Für die abschließende Texturierung wird aus den `PatchFace`-Objekten eine OPEN INVENTOR-Szene generiert, um diese als texturierte Oberfläche darzustellen. Der Abschnitt beschreibt zunächst das Vorgehen für die Erstellung der OPEN INVENTOR-Szene und anschließend wie auf Netzwerkebene die Texturierung erfolgt. Letzteres beinhaltet auch die Verwendung der *TAM*-Texturen mittels *Hardware Shader*.

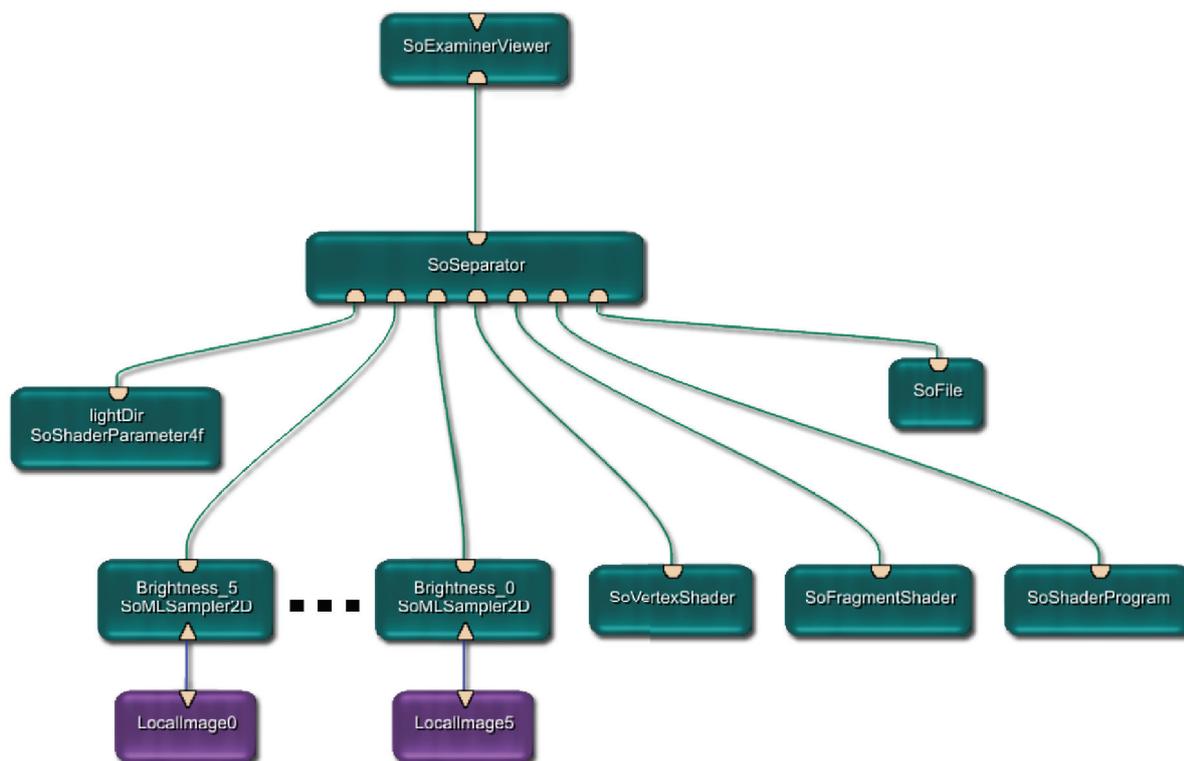
### 5.4.1 Generierung der Open Inventor-Szene

Die Generierung der OPEN INVENTOR-Szene basiert auf der `WEMPartitioning`-Datenstruktur. Für die Szene wird ein `SoTriangleStripSet`-Knoten erzeugt, der für jedes `PatchFace` aus dem `WEMPartitioning`-Objekt ein Dreieck generiert. Die Informationen über Position, Normale und Parameterkoordinate der Eckpunkte aus jedem `PatchFace` werden dazu in einem `SoVertexProperty`-Knoten abgespeichert und abschließend dem `SoTriangleStripSet`-Knoten übergeben, welcher nach außen weitergereicht wird. Die so erzeugte parametrisierte Oberfläche kann für eine spätere Wiederverwendung als OPEN INVENTOR-Format abgespeichert werden. Ein erneutes Generieren des Vektorfeldes oder der Parametrisierung ist somit nicht erforderlich.

### 5.4.2 Texturierung in MeVisLab

Für die einfache Abbildung einer Schraffurtextur auf der parametrisierten Oberfläche genügt es auf Netzwerkebene einen `SoTexture2`-Knoten vor dem Modul anzuschließen, welches die Oberfläche generiert. Der `SoTexture2`-Knoten übernimmt zum einen den Zugriff auf die darin abgespeicherte Bildtextur über die Parameterkoordinaten der Eckpunkte aus dem `SoTriangleStripSet`-Knoten. Zum anderen kann die Randbehandlung eingestellt werden, wenn die Koordinaten außerhalb des standardisierten Intervalls  $[0, 1]$  liegen. Für diese Arbeit ist aufgrund des kontinuierlichen Parameterraumes und lokalen Parametrisierung eine Kachelung notwendig, was ein wiederholbares Texturmuster voraussetzt.

Das *TAM*-Konzept von PRAUN U. A. [2001] erfordert die Verwendung von *Hardware Shader*, die die einzelnen *TAM*-Texturen entsprechend der lokalen Helligkeit an einem Eckpunkt überblenden. In MEVISLAB existiert bereits ein Netzwerk, das für eine gegebene OPEN INVENTOR-Szene dieses Konzept umsetzt und für diese Arbeit verwendet wird. Einzig die Implementierung unterschiedlich aufgelöster *TAM*-Texturen ist nicht integriert, so dass momentan keine Skalierbarkeit beim Vergrößern oder Verkleinern der dargestellten Oberfläche erfolgt. Das Netzwerk ist in Abb. 5.4 dargestellt. Zunächst wird die Position der Lichtquelle angegeben und es erfolgt durch die `SoMLSampler2D`-Module das Einlesen von sechs *TAM*-Texturen, die jeweils eine andere Helligkeitsstufe repräsentieren. Das *Vertex Shader*-Programm im Modul `SoVertexShader` implementiert die Helligkeitsbestimmung für jeden Eckpunkt auf Basis der definierten Lichtposition. Daraus wird ein Wichtungsfaktor ermittelt, der angibt, wie groß der Überblendungsanteil jeder Textur für diesen Eckpunkt ist. Anschließend wird dieser Faktor an das *Fragment Shader*-Programm im Modul `SoFragmentShader` weitergeleitet. In diesem erfolgt zunächst für jeden Eckpunkt der Texturzugriff und die Überblendung entsprechend des Wichtungsfaktors und den Parameterkoordinaten. Letztere stammen ebenfalls aus dem *Vertex Shader*-Programm und werden darin aus der angeschlossenen OPEN INVENTOR-Oberfläche extrahiert, welche über das Modul `SoFile` eingelesen wird.



**Abb. 5.4:** Darstellung des in MEVISLAB vorhandenen Netzwerkes zur Texturierung von Oberflächen mittels TAM-Konzept. Verwendet wurden vorrangig Module der SOSHADER-Bibliothek zum Einlesen der Texturen sowie deren beleuchtungsabhängige Überblendung durch einen *Vertex* und *Fragment Shader*. Für diese Arbeit musste nur noch die parametrisierte Oberfläche in eine OPEN INVENTOR-Szene konvertiert und diese dem Netzwerk angefügt werden. Wegen der besseren Lesbarkeit ist im Netzwerk nur die hellste und dunkelste Schraffurtextur durch jeweils einen `SoMLSampler2D`-Knoten dargestellt.

Mit Hilfe von letzterem wird die oben abgespeicherte Oberfläche der anatomischen Struktur geladen. Das Modul `SoExaminerViewer` dient in diesem Netzwerk zur Darstellung der texturierten Oberfläche.

## 5.5 Entwickelte MeVisLab-Module

In dieser Arbeit sind verschiedene Module entwickelt wurden, die die Erzeugung des Vektorfeldes und der Oberflächenparametrisierung realisieren. Im Folgenden soll darüber ein kurzer Überblick gegeben werden. Kombiniert wurden diese mit den bereits unter Abschnitt 5.1.2 erwähnten und in MEVISLAB integrierten Modulen. Zur besseren Veranschaulichung des Informationsflusses wird daher anschließend das im Rahmen dieser Arbeit entstandene MEVISLAB-Netzwerk präsentiert.

### 5.5.1 Entwickelte Module

Die entwickelten Module sind in Tabelle 5.1 übersichtsartig zusammengefasst und zeigen jeweils deren Namen und Funktionen. Zusätzlich wurde eine Klasse `Geometry` implementiert, die als *Wrapper*-Objekt für verschiedene geometrische Operationen wie Projektion und Rotation von Vektoren oder Punkten verwendet wird. Hierzu werden die von OPEN INVENTOR angebotenen Datenstrukturen und Funktionalitäten für Matrix- und Vektoroperationen verwendet.

Modul	Funktion
<code>UMDWEMConvertInventor</code>	Konvertierung eines OPEN INVENTOR-Szenengraphen in eine WEM-Struktur
<code>UMDWEMVectorField</code>	Erzeugung des Vektorfeldes
<code>UMDWEMParameterization</code>	Erzeugung der <i>Patches</i> mit deren anschließender Parametrisierung
<code>UMDWEMPatchRenderer</code>	Konvertierung der WEM-Struktur in einen OPEN INVENTOR-Szenengraphen
<code>UMDIllustrationPreprocessing</code>	Visualisierung von Vorverarbeitungsschritten
<code>Geometry</code>	<i>Wrapper</i> -Objekt für geometrische Operationen

**Tabelle 5.1:** Auflistung der für diese Arbeit entwickelten MEVISLAB-Module zur Erzeugung einer WEM-Datenstruktur der anatomischen Oberfläche, Generierung des Vektorfeldes, Parametrisierung der Oberfläche sowie deren Umwandlung in eine OPEN INVENTOR-Szene.

Das Modul `UMDWEMConvertInventor` ist die überarbeitete Version des oben erwähnten originalen WEM-Konvertierungsmoduls. Als Eingabe erhält es eine OPEN INVENTOR-Datei, die die Dreiecksgeometrie der anatomischen Oberfläche enthält. Die Ausgabe ist ein WEM-Objekt, in das die topologischen Zusammenhangsinformationen der Eingangsgeometrie abgespeichert sind.

Zur Erzeugung des Vektorfeldes dient das Modul `UMDWEMVectorField`, welches die Vorzugsrichtungen für die jeweilige anatomische Struktur bestimmt und diese mit den eingehenden Krümmungsinformationen und dem WEM-Objekt zu einem endgültigen Vektorfeld kombiniert. Implementiert sind darin die unter Abschnitt 5.2.4 vorgestellten Datenstrukturen. Als Ausgabe wird ein `UMDWEMVectorFieldInfo`-Objekt erzeugt, das für jedes `WEMFace` einen entsprechenden Richtungsvektor speichert. Bezüglich dem Vorgehen zur Bestimmung der Vorzugsrichtung kann zwischen den Strukturen *Muscle*, *Vessel* und *Organ* ausgewählt werden. Für Gefäße muss zum einen die erzeugte und zugehörige Skelettlinie in Form einer XML-Datei eingeladen werden. Zum anderen können der Wichtungsfaktor  $w$  und die Anzahl der Iterationen zur Laplace-Glättung der Skelettlinie eingestellt werden (vgl. Abschnitt 4.3.4 auf Seite 75). Bei Muskelstrukturen sind zudem die drei Hauptachsen notwendig, welche auf Netzwerkebene über Feldverbindungen von dem Modul `UMDSMainAxis` übernommen werden. Anschließend erfolgt in der Klasse

`MuscleStructure` die Ermittlung der längsten Achse für den elongierten Muskeltyp bzw. zusätzlich die zweitlängste Achse für federförmige Muskelstrukturen.

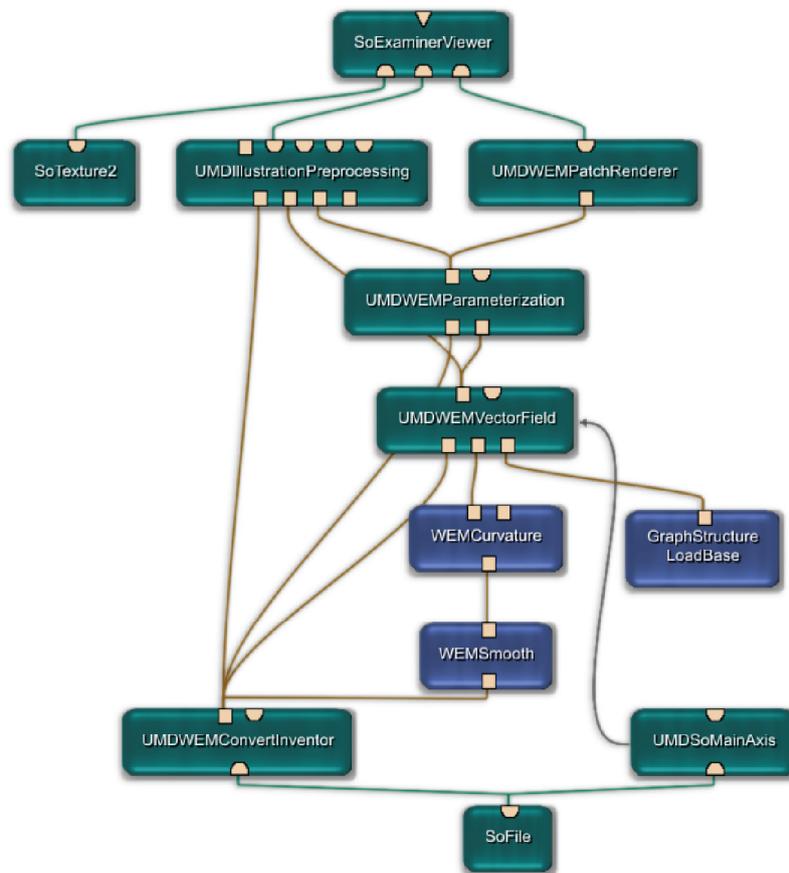
Die Parametrisierung der Oberfläche erfolgt im Modul `UMDWEMParameterization`, das als Eingabe ein `WEM`- und `UMDWEMVectorFieldInfo`-Objekt erhält. Anhand der eingehenden Informationen werden die einzelnen *Patches* für die gesamte Oberfläche erzeugt und deren Übergänge entsprechend den Ausführungen von Abschnitt 4.4.4 behandelt. Als Parameter können dabei die aus Abschnitt 4.4.2 eingeführten Winkelschwellwerte  $\alpha_{divergence}$  und  $\alpha_{rotation}$  gesetzt werden. Alle erzeugten *Patches* und Parameterkoordinaten für die Eckpunkte jedes `WEMFace` werden anschließend in einem `UMDWEMParameterizationInfo`-Objekt abgespeichert.

Das Modul `UMDWEMPatchRenderer` erzeugt aus der parametrisierten *WEM*-Oberfläche wieder eine OPEN INVENTOR-Oberfläche durch Erzeugung des oben genannten `SoTri-angleStripSet`-Knotens.

Zur Visualisierung einzelner Vorverarbeitungsschritte, wie beispielsweise der Anzeige des Vektorfeldes oder des jeweiligen Referenzdreieckes der *Patches*, dient das Modul `UMDIllustrationPreprocessing`. Es erhält die *WEM*-Struktur, das Vektorfeld sowie die Informationen über die erzeugten *Patches* und generiert für die Anzeige verschiedene OPEN INVENTOR-Szenengraphen.

## Beispielnetzwerk

Das in dieser Arbeit entstandene MEVISLAB-Netzwerk zur Schraffierung der anatomischen Strukturen ist in Abb. 5.5 dargestellt. Darin sind u. a. die oben erwähnten Module und ihre Feldverbindungen zu sehen. Das `SoFile`-Modul lädt zunächst den OPEN INVENTOR-Szenengraphen mit der entsprechenden Oberfläche. Anschließend wird diese zum einen in die *WEM*-Datenstruktur umgewandelt und zum anderen von dieser die Hauptachsen ermittelt. Letztere werden für die Bestimmung des Vektorfeldes an das Modul `UMDWEMVectorField` weitergeleitet. Auf Basis der *WEM*-Struktur wird die Oberfläche geglättet und danach dem Modul zur Krümmungsapproximation weitergereicht. Die Krümmungs- und Hauptachseninformationen sowie der geladene Skelettgraph aus `GraphStructure` werden für die Erzeugung des modellbasierten Vektorfeldes verwendet. Das generierte Vektorfeld wird für die Erzeugung der *Patches* der ungeglätteten *WEM*-Oberfläche an das Parametrisierungsmodul weitergeleitet. Die einzelnen parametrisierten Dreiecke werden anschließend wieder zu einer OPEN INVENTOR-Szene zusammengesetzt und können durch das `SoExaminerViewer`-Modul texturiert angezeigt werden. Die Bereitstellung und Zugriff auf die Textur mit Hilfe der Parameterkoordinaten erfolgt dabei durch das Modul `SoTexture2`. Letzteres wird bei der Verwendung der *TAM*-Texturen durch die entsprechenden Module der `SOSHADER`-Bibliothek ersetzt.



**Abb. 5.5:** Darstellung des in dieser Arbeit entwickelten MEVISLAB-Netzwerk, in dem alle entwickelten und Teile der verwendeten Module integriert sind. Der Informationsfluss erfolgt von unten nach oben, ausgehend von der OPEN INVENTOR-Datei einer anatomischen Oberfläche.

## 6 Ergebnisse

Dieses Kapitel präsentiert die Ergebnisse des entwickelten Verfahrens. Angewendet wurde es auf verschiedene patientenspezifische Oberflächenmodelle von Gefäßen, Muskelstrukturen sowie Knochen und Organen. Die Ergebnisse zeigen die jeweils erzeugten Vektorfelder und die sich daraus ergebene Schraffierung der anatomischen Strukturen. Diskutiert wird dabei auch der Einfluss von diversen Parametereinstellungen auf das Resultat. Neben Einzelvisualisierungen der Oberflächen erfolgt zusätzlich eine kombinierte Oberflächendarstellung von anatomischen Strukturen, wo die Schraffierungen integriert wurden. Das Kapitel schließt mit einer Laufzeitbetrachtung des entwickelten Verfahrens ab.

### 6.1 Vektorfelder

Für die Generierung der Vektorfelder wurden verschiedene zur Verfügung stehende Beispieldatensätze der jeweiligen anatomischen Struktur verwendet. Diese wurden ausgewählt, um folgende drei Aspekte hinsichtlich der Qualität des Vektorfeldes zu untersuchen:

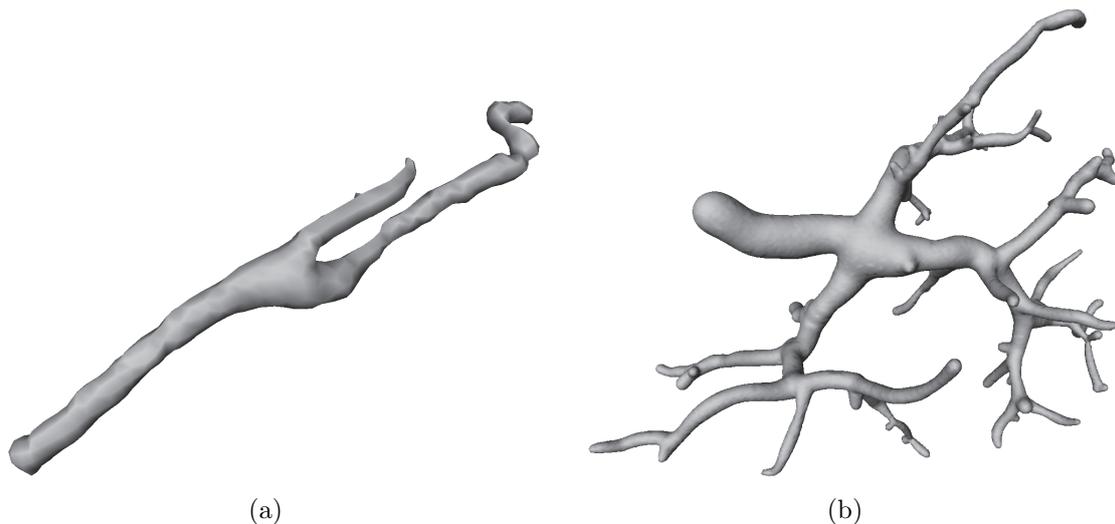
- **Korrektheit:** Dazu zählt zum einen die visuelle Beurteilung über den geodätischen Verlauf der Vektoren und zum anderen, ob dieser der vorhandenen, modellspezifischen Vorzugsrichtung entspricht. Zur letzteren gehört auch die Beurteilung, ob charakteristische Oberflächenverläufe durch das Vektorfeld beschrieben werden.
- **Robustheit:** Diese bezieht sich einerseits gegenüber der Rauigkeit der Modelloberflächen und andererseits gegenüber der Variabilität zwischen verschiedenen Datensätzen. Das Verhalten bei rauen Oberflächenverläufen hat dabei direkten Einfluss auf die oben beschriebene Korrektheit des Richtungsverlaufes.
- **Parametereinstellungen:** Die verschiedenen Parametereinstellungen zur Glättung der Oberfläche sowie der Skelettlinie sollen untersucht werden, inwieweit diese die Qualität des Vektorfeldes beeinflussen.

Dargestellt ist jeweils der Richtungsvektor für jedes Oberflächendreieck. Dieser entspricht der Mittelung des Richtungsvektors seiner Eckpunkte. Zur Krümmungsapproximation

wurden die in Abschnitt 4.3.3 gewählten Operatoren für die jeweilige anatomische Struktur gewählt. Fehlende Krümmungsinformationen sind durch den adaptiven Ansatz ermittelt worden (vgl. Abschnitt 4.3.2, Seite 67). Die anschließende Optimierung der Hauptkrümmungsrichtung wird vorrangig durch die beiden Parameter  $\delta_{ratio}$  und  $\delta_{reliable}$  gesteuert (vgl. Abschnitt 3.9, Seite 48). Für die hier verwendeten Strukturen haben sich dabei die Durchschnittswerte der jeweiligen Parameter als geeignet gezeigt. Diese können direkt aus den Krümmungsinformationen gewonnen und automatisch gesetzt werden. Eine Regulierung durch den Benutzer ist somit nicht erforderlich.

### 6.1.1 Gefäßstrukturen

Als Beispiele für Gefäßstrukturen zeigt die Abb. 6.1 jeweils die Oberfläche einer Halsschlagader (*Carotis*) (a) sowie der Pulmonalvene einer Leber (b). Auffällig ist bei der Halsschlagader die raue Modelloberfläche und bei der Pulmonalvene die starke Verästelung des Gefäßbaumes. In der Abb. 6.2 ist zunächst ein Vergleich dargestellt, inwiefern



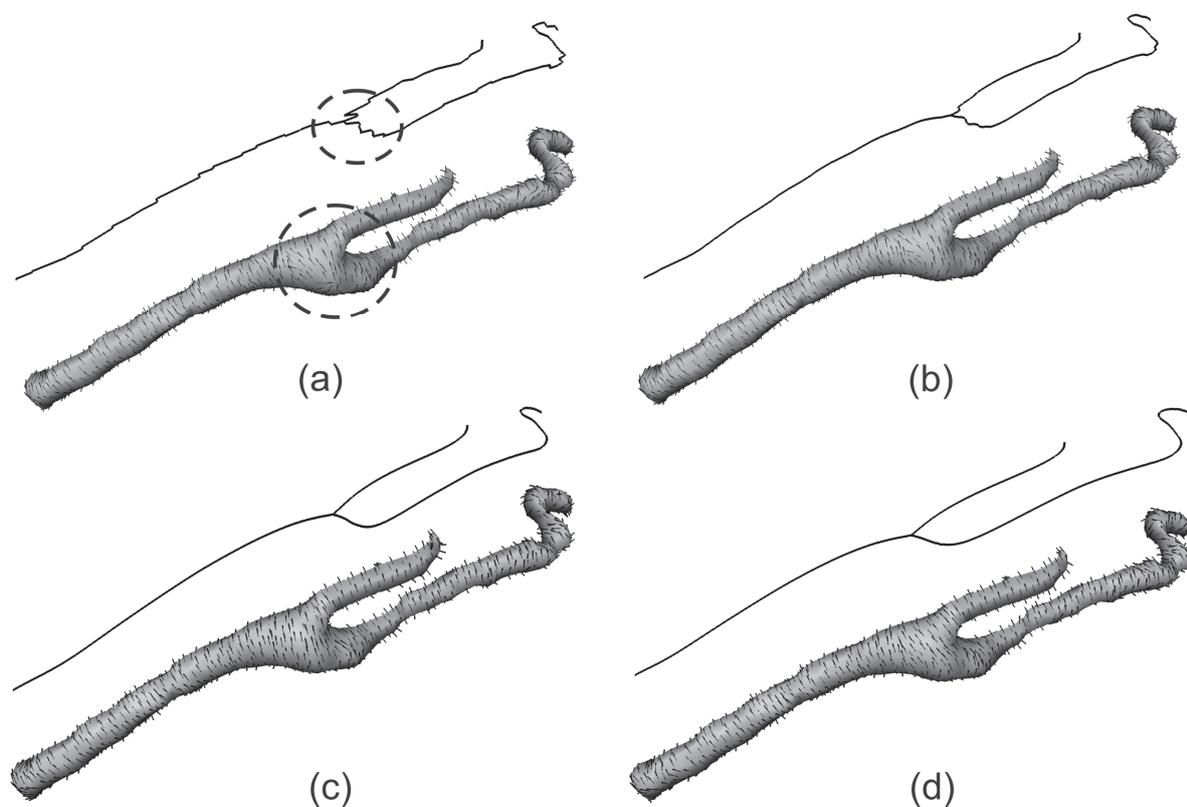
**Abb. 6.1:** Verwendung einer (a) Halsschlagader und (b) Portalvene der Leber als Beispieldatensätze für Gefäßstrukturen.

die Glättung der zugrunde liegenden Skelettlinie den Verlauf des Vektorfeldes bestimmt. Dazu zeigt 6.2 (a) das Vektorfeld der Halsschlagader, welches aus dem darüber liegenden originalen Skelettlinienverlauf erzeugt wurde. Erkennbar ist der zackige und unregelmäßige Verlauf im Gefäßskelett, besonders im markierten Bifurkationsbereich. Dadurch weichen die daraus ermittelten radialen Richtungsvektoren stark von der gewünschten Vorzugsrichtung ab. Durch die Kombination mit den Krümmungsinformationen entstehen somit Diskontinuitäten im geodätischen Verlauf des resultierenden Vektorfeldes. Die Laplace-Glättung der Skelettlinie in 6.2 (b) durch Verwendung des Glättungsfaktors

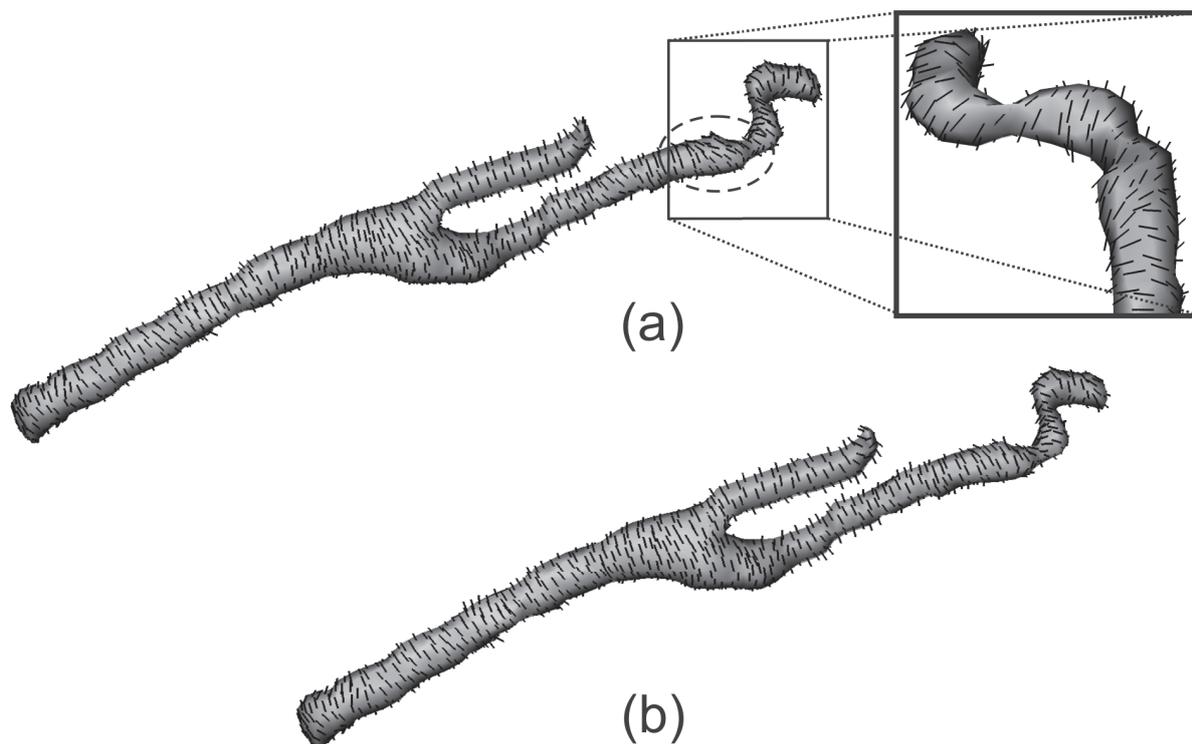
$w = 0.3$  und fünf Iterationsstufen bewirkt eine allgemeine Verbesserung im Richtungsverlauf des Vektorfeldes. Einige Unregelmäßigkeiten konnten ausgeglichen werden. Zu sehen sind jedoch noch Diskontinuitäten an der Bifurkation aufgrund einer noch nicht ausreichenden Glättung an dieser Stelle in der Skelettlinie. Erreicht wird diese in 6.2 (c) durch eine Erhöhung von  $w = 0.7$ . Das daraus generierte Vektorfeld weist einen überwiegend geodätischen Richtungsverlauf auf, das der gewünschten Vorzugsrichtung auch entspricht. Eine Erhöhung der Glättungsiterationen auf zehn Durchläufe erbrachte in (d) keine Verbesserung des Verlaufes. Im Gegenteil, eine derartige Steigerung der Iterationen in Kombination mit dem eingestellten Glättungsfaktor führte zu einem überhöhten Glätten charakteristischer Linienverläufen. Der geodätische Verlauf an der Bifurkation verschlechterte sich dadurch sogar.

Die Abb. 6.3 stellt einen Vergleich der Robustheit gegenüber Oberflächenunebenheiten und der dabei vorausgehenden Oberflächenglättung dar. Verwendet wurde ebenfalls wieder die Halsschlagader, deren Skelettlinie mit den Parametereinstellungen von Abb. 6.2 (c) geglättet wurde. In Abb. 6.3 (a) ist das Vektorfeld auf dem ungeglätteten Modell zu sehen. Bereits hier zeigt das Vektorfeld ein robustes Verhalten, was auch an dem vergrößerten und rotierten Bereich eines rauen Gefäßabschnittes zu erkennen ist. Der Richtungsverlauf besitzt nur geringe Diskontinuitäten an der markierten Stelle, die in (b) durch eine zusätzliche Oberflächenglättung mit dem Glättungsfaktor  $w = 0.7$  und drei Iterationsstufen kompensiert werden konnten. Scharfkantige Oberflächenbereiche konnten durch die Glättung reduziert werden, so dass die ermittelten Krümmungsinformationen an einem Eckpunkt weniger von ihrer Nachbarschaft abweichen. Eine erhöhte Glättung erbrachte auch hier keine sichtbaren Verbesserungen, sondern ergab neue Diskontinuitäten im Vektorfeld. Erklären lässt sich dies, dass bei einer starken Verschiebung der Eckpunkte neue scharfkantige Bereiche entstanden sind. Gerade bei gering aufgelösten Dreiecksnetzen, wie es bei dem gezeigten Gefäßmodell der Fall ist, tritt dieses Problem verstärkt auf.

Um das Verhalten bei Gefäßbäumen zu untersuchen, zeigt Abb. 6.4 das erzeugte Vektorfeld auf mehreren vergrößerten Ausschnitten der Pulmonalvene. Für die Glättung der Skelettlinie und Oberfläche wurden dabei die als angemessen beurteilten Parametereinstellungen aus Abb. 6.2 (c) sowie 6.3 (b) verwendet. Der Gesamteindruck ergibt auch hier einen geodätischen Richtungsverlauf im Vektorfeld, der der Vorzugsrichtung entspricht, aber dabei auch charakteristische Oberflächenbereiche beschreibt. In Abb. 6.4 (a) zeigt der Verlauf ein erwartetes Verhalten auch an Bi- und Trifurkationen. Die einzelnen Gefäßäste werden zudem durch den radialen Richtungsverlauf der Vektoren gut beschrieben. Die Abb. 6.4 (b) und (c) verdeutlichen die korrekte Beschreibung auch von kurzen Gefäßästen oder Oberflächenauswölbungen. An einer einzigen Stelle bildete sich jedoch ein Strudel im Vektorfeld, der in Abb. 6.4 (d) markiert ist. Diese Stelle beschreibt eine Teilung des Gefäßes in vier Äste, wodurch es lokal zu einer ungenauen Bestimmung der radialen Vorzugsrichtungen kommt.



**Abb. 6.2:** Vergleich des Einflusses vom Glättungsgrad der Skelettlinie auf den Verlauf des Vektorfeldes: (a) Die ungeglättete Skelettlinie verursacht im Bereich der Bifurkation Diskontinuitäten im Richtungsverlauf. (b) Eine Glättung des Skelettes mit  $w = 0.3$  und fünf Iterationen verbessern den Verlauf im betreffenden Gebiet. (c) Die Erhöhung auf  $w = 0.7$  reduziert auch die letzten sichtbaren Diskontinuitäten. (d) Eine weitere Erhöhung auf 10 Iterationen verursacht eine zu starke Glättung der Skelettlinie, was zu erneuten Diskontinuitäten an der Bifurkation führt.

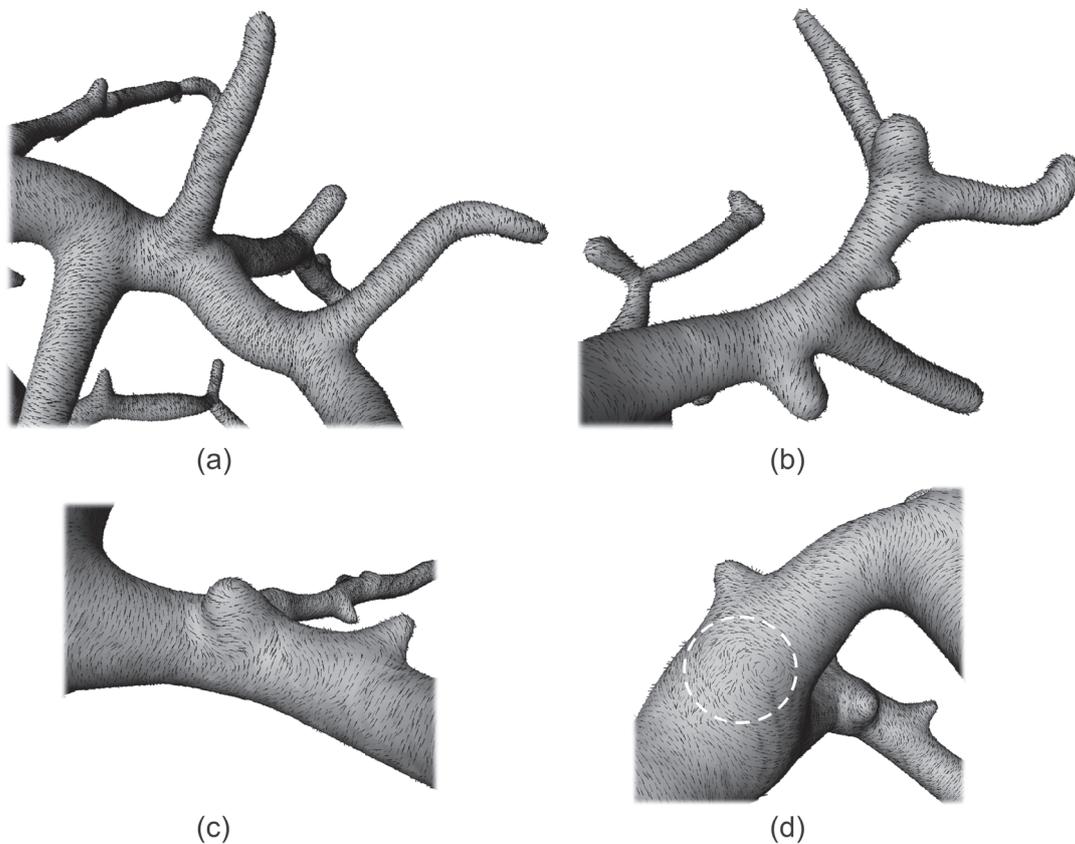


**Abb. 6.3:** (a) Das erzeugte Vektorfeld auf der ungeglätteten Oberfläche der Halsschlagader zeigt bereits ein geodätisch und radial verlaufendes Richtungsfeld. Der vergrößerte und rotierte Bereich bestätigt dies auch an einen besonders rauen Gefäßabschnitt. Die leichte Diskontinuität im markierten Bereich konnte in (b) durch eine Glättung mit  $w = 0.7$  und drei Iterationen reduziert werden.

### 6.1.2 Muskelstrukturen

Für die Beurteilung der Vektorfelder auf Muskelstrukturen standen in dieser Arbeit nur Datensätze zur Verfügung, die dem elongierten Muskeltyp entsprechen. Die Abb. 6.5 zeigt dazu jeweils zwei Halsmuskel, welche sich hinsichtlich ihrer Formorientierung unterscheiden. Der Halsmuskel in (b) zeigt dabei gegenüber der Struktur in (a) eine interne Verdrehung der Objektform auf. Überprüft werden soll dadurch, wie genau die längste Hauptachse die modellspezifischen Vorzugsrichtung approximiert. Dazu sind in beiden Strukturen jeweils Ursprung und Ansatz hell hervorgehoben. Als Parametereinstellung fließt bei Muskelstrukturen nur die Glättung der Oberfläche ein. Allgemein ist zu beobachten, dass die zur Verfügung stehenden Muskeldatensätze, im Gegensatz zu den meisten Gefäßstrukturen, bereits aus der Triangulierung heraus einen gleichmäßigeren Oberflächenverlauf besitzen.

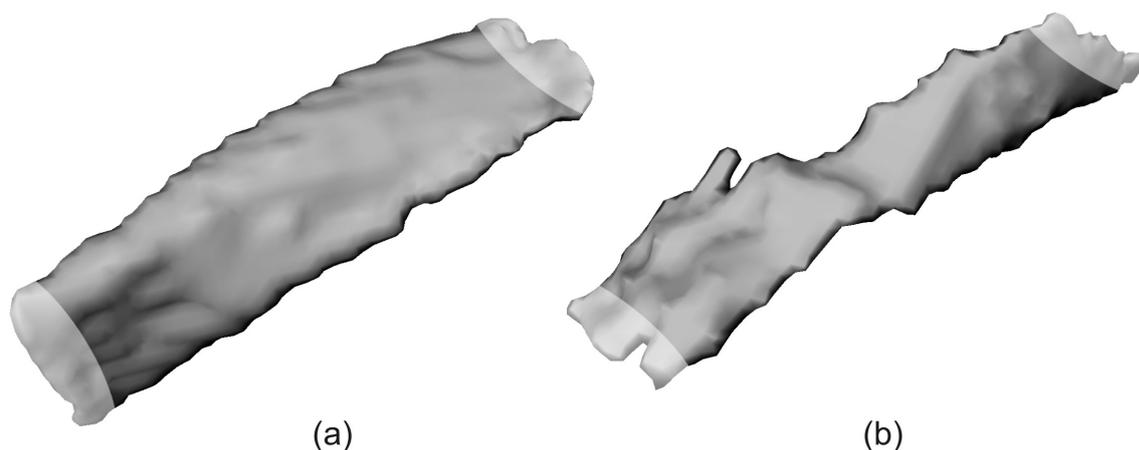
In Abb. 6.6 (a) ist das resultierende Vektorfeld auf dem ersten Halsmuskel zu sehen. Für dessen Erzeugung wurde das Modell vorher nicht geglättet. Anhand des Ergebnisses ist



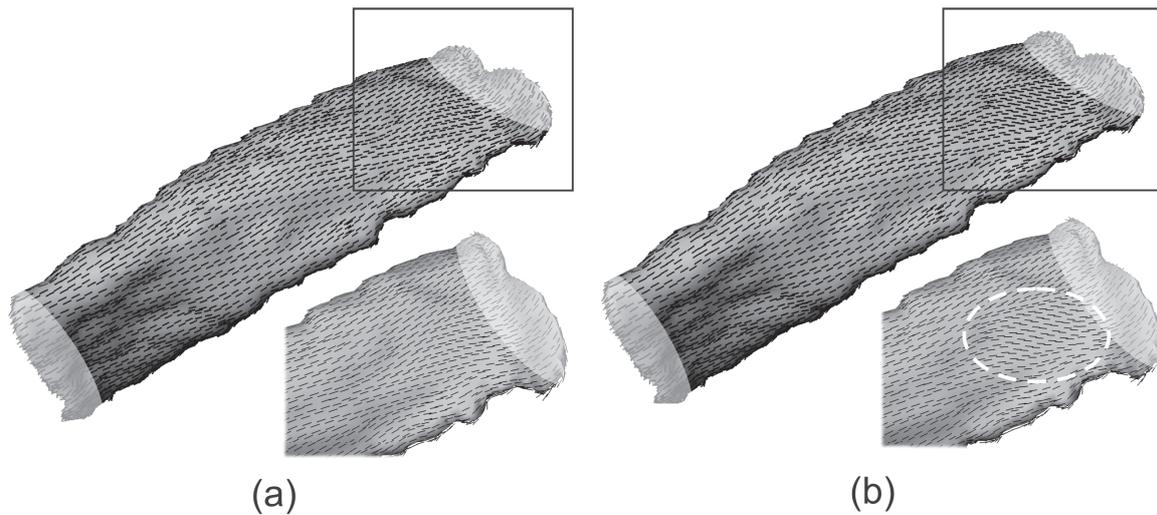
**Abb. 6.4:** Die vergrößerten Ausschnitte (a)-(c) der Portalvene lassen ein korrekten und geodätischen Verlauf des Vektorfeldes auf den verschiedenen Gefäßästen erkennen. Der Oberflächenverlauf an Bi- und Trifurkationen (a) wird ebenso gut beschrieben, wie an kleinen Oberflächenauswölbungen (b) und (c). Lediglich an einer einzigen Stelle wurde ein Strudel erzeugt (d).

bereits hier ein geodätischer Richtungsverlauf zu erkennen, der sich zwischen Ursprung und Ansatz hindurchzieht. Der vergrößerte und rotierte Ausschnitt zeigt dabei auch, dass konkave und konvexe Oberflächenbereiche bereits gut durch das Vektorfeld beschrieben werden. In Abb. 6.6 (b) wurde das Oberflächenmodell vorher geglättet, wobei als Parametereinstellungen  $w = 0.3$  und einer Iterationsstufe verwendet wurde. Der Gesamteindruck zeigt zunächst, dass sich der Richtungsverlauf im Allgemeinen nicht stark verändert hat. Der vergrößerte Bereich lässt jedoch eine markante Richtungsänderung erkennen, welche die markierte Region planarer erscheinen lässt, als sie tatsächlich ist (vgl. Abb. 6.5 (a)). Aufgrund der zu starken Glättung gingen hier kleine aber markante Oberflächendetails verloren, welches sich auch auf die Krümmungsapproximation auswirkte. Die dabei bestimmten Informationen kennzeichneten die Region entsprechend planar und bewirkten in Kombination mit der Vorzugsrichtung die Richtungsänderung.

Die Abb. 6.7 (a) zeigt das Vektorfeld auf dem zweiten Halsmuskel. Bei diesem wurde auf eine Glättung verzichtet. Der daraus resultierende Richtungsverlauf lässt ebenfalls einen geodätischen Verlauf erkennen, der der Vorzugsrichtung weitestgehend entspricht. Im vergrößerten Bereich von Abb. 6.7 (b) ist die korrekte Beschreibung des lokalen Oberflächenverlaufes bei gleichzeitiger Einhaltung der Vorzugsrichtung zu sehen. Auch in Abb. 6.7 (c) spiegelt sich dieses Verhalten wider und divergiert nur leicht am Muskelansatz. Dieser ließe sich korrigieren, wenn eine bessere Approximation der Muskelmittellinie erreicht werden würde. Aufgrund der in sich verdrehten Objektform stellt die längste Hauptachse an manchen Stellen nur eine grobe Annäherung der Vorzugsrichtung dar. Im Ausblick dieser Arbeit wird ein Verbesserungsvorschlag dafür aufgezeigt.



**Abb. 6.5:** Verwendung von zwei patientenspezifischen Halsmuskeln als Beispieldatensätze für Gefäßstrukturen. Das Modell in (a) unterscheidet sich zu (b) durch einen weniger in sich verdrehten Objektverlauf. In beiden Strukturen sind Muskelursprung und -ansatz hervorgehoben.

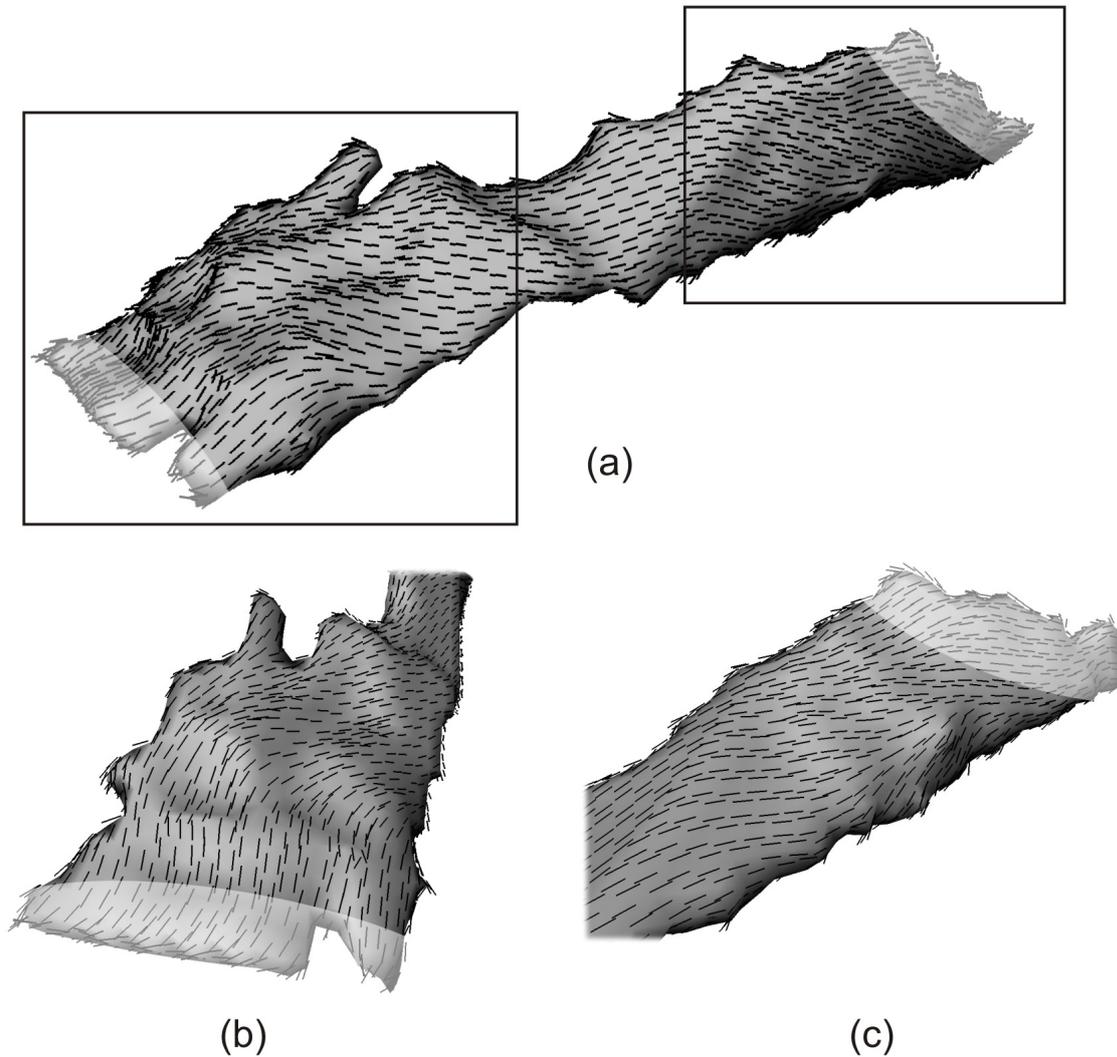


**Abb. 6.6:** Vergleich des Einflusses vom Glättungsgrad der Muskeloberfläche auf den Verlauf des Vektorfeldes. (a) Das erzeugte Vektorfeld zeigt bereits auf dem ungeglätteten Modell ein geodätisches, entlang der Kontraktionsrichtung verlaufendes Richtungsfeld. (b) Durch eine Oberflächenglättung mit  $w = 0.3$  und einer Iterationsstufe gingen in der markierten Region kleine Oberflächendetails verloren. Der Richtungsverlauf beschreibt dadurch eine mehr planare Region als tatsächlich vorhanden ist.

### 6.1.3 Organe und Knochenstrukturen

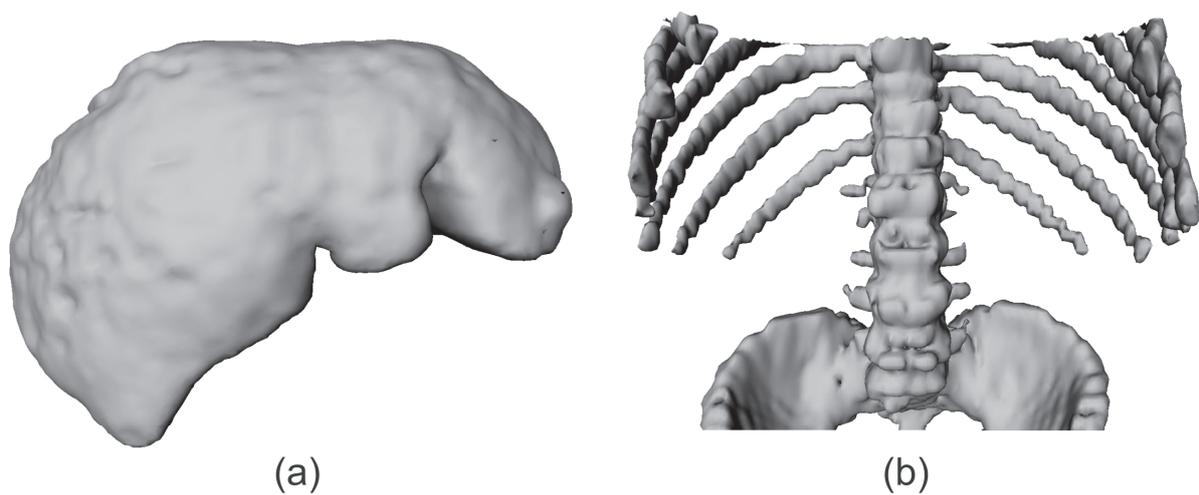
Als Beispiele für Organe und Knochen wird die in Abb. 6.8 (a) gezeigte Leber sowie die in (b) abgebildeten unteren Rippenbögen mit einem Teil der Hüfte und Wirbelsäule verwendet. Beide Strukturen sind durch einen komplexen Oberflächenverlauf und degenerierte Bereiche, in Form von Löchern, gekennzeichnet. Bezüglich der Krümmungsapproximation dieser Strukturen wurde eine erweiterte Nachbarschaft von  $r = 3$  für den *BiQuad-BezOP*-Operator verwendet. Dies resultiert aus den Beobachtungen von Abschnitt 4.3.2 auf Seite 73. Untersucht werden soll, inwiefern die Glättung der Oberfläche und über die  $k$ -Nachbarschaft Einfluss haben auf das resultierende Vektorfeld.

Die Abb. 6.9 (a) zeigt zunächst für die Leber das initiale Vektorfeld auf der ungeglätteten Oberfläche und  $k = 0$ . Erkennbar ist der ungleichmäßige und stark verwirbelte Richtungsverlauf der ersten Hauptkrümmungsrichtung. Verbessert wird das Ergebnis, wenn die Richtungsvektoren, wie in Abb. 6.9 (b), über eine 3-Nachbarschaft geglättet werden. Dabei entspricht das Vektorfeld schon mehr einem geodätischen Verlauf, welcher jedoch noch gelegentlich Strudel aufweist (Beispielmarkierung). Diese beschreiben u. a. zwar charakteristische Oberflächenbereiche, wie Senken oder Ausbuchtungen, jedoch stören sie die Gesamtwahrnehmung. Durch die zusätzliche Glättung der Oberfläche mit  $w = 0.7$  und fünf Iterationen lösen sich in Abb. 6.9 (c) die meisten dieser kritischen Bereiche auf. Das resultierende Vektorfeld verbesserte sich zudem bezüglich des geodätischen Verlaufes,

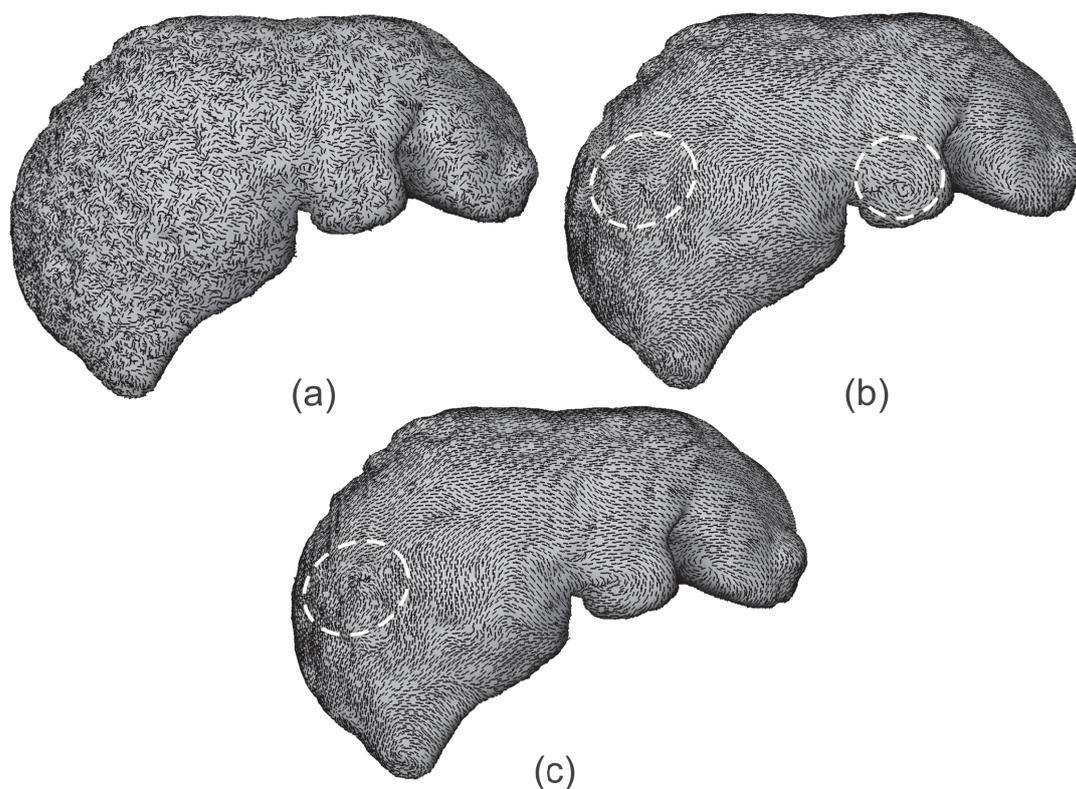


**Abb. 6.7:** (a) Das Vektorfeld auf der zweiten mehr in sich verdrehten Muskelstruktur stellt ebenfalls einen korrekten Richtungsverlauf dar. Die vergrößerten Abschnitte in (b) und (c) zeigen dabei, dass auch charakteristische Oberflächendetails beschrieben werden. (c) Nur eine leichte Richtungsänderung am Muskelende ist zu beobachten. Dies resultiert aus einer ungenauen Beschreibung der Kontraktionsrichtung, welche durch die längste Hauptachse approximiert wurde.

wobei das lokale Krümmungsverhalten dennoch angemessen beschrieben wird.

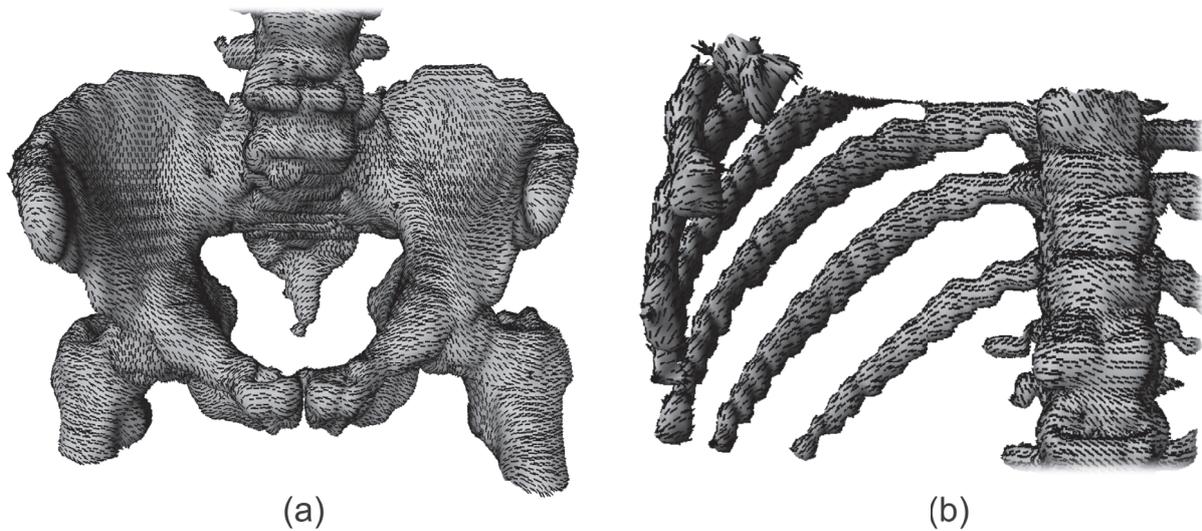


**Abb. 6.8:** Verwendung der in (a) dargestellten Leber und den in (b) abgebildeten Rippenbögen mit einem Teil der Hüfte und Wirbelsäule als Beispieldatensätze für Organe und Knochenstrukturen.



**Abb. 6.9:** (a) Das initiale Vektorfeld der Leberoberfläche, bestehend aus der ersten Hauptkrümmungsrichtung, zeigt viele Verwirbelungen und eine starke Unregelmäßigkeit im Richtungsverlauf. (b) Die Glättung über eine 3-Nachbarschaft resultiert in einem stärker geodätisch verlaufendem Vektorfeld, wobei jedoch die markierten Strudel mehr hervortreten. (c) Kompensiert werden diese durch eine zusätzliche Oberflächenglättung mit  $w = 0.7$  und fünf Iterationen.

Die gleichen Glättungsparameter sind in Abb. 6.10 zur Generierung des Vektorfeldes für die Knochenstruktur verwendet worden. Auch hier ist ein geodätischer Verlauf der Richtungsvektoren zu erkennen, welcher das lokale Krümmungsverhalten angemessen beschreibt. Besonders der konkave Beckenbereich sowie die radiale Form der noch zu sehende Oberschenkelknochen werden in Abb. 6.10 (a) durch die Richtungsvektoren gut wiedergegeben. Im Vergleich zu den anatomischen Illustrationen ist allerdings zu vermuten, dass ein Zeichner den Schraffurverlauf bei bestimmten Strukturbereichen anders wählen würde, als dieser vom Vektorfeld beschrieben wird. Erkennbar ist dies in Abb. 6.10 (b), wo das Vektorfeld auf einen vergrößerten Ausschnitt der unteren, rechten Rippen abgebildet ist. Auf diesen verlaufen die Richtungsvektoren eher seitlich, obwohl hier ein radialer Verlauf sinnvoller wäre. Auch durch eine Erhöhung des Nachbarschaftsradius für die Krümmungsapproximation sowie Variation in den Glättungsparametern konnte diese gewünschte Richtung aufgrund des fehlenden Modellwissens nicht erreicht werden.



**Abb. 6.10:** Resultierendes Vektorfeld auf den verwendeten Knochenstrukturen. Zur besseren Darstellung ist jeweils nur ein vergrößerter Ausschnitt der Gesamtstruktur abgebildet. In beiden Ausschnitten ist ein geodätisch verlaufendes Vektorfeld zu erkennen. In (a) werden durch die Richtungsvektoren auch markante Oberflächenverläufe wie der konkave Beckenbereich sowie die konvexe Form der noch zu sehenden Oberschenkelknochen beschrieben. Der in (b) dargestellte, seitliche Richtungsverlauf auf den Rippen entspricht allerdings nicht den Erwartungen. Hier wäre ein radialer Verlauf angemessener.

#### 6.1.4 Schlussfolgerungen

Die präsentierten Ergebnisse bezüglich der erzeugten Vektorfelder zeigen, dass das entwickelte Verfahren im Allgemeinen den zu erwartenden Richtungsverlauf aufweist und dabei robust gegenüber den Oberflächenunebenheiten ist. Die erzeugten Vektorfelder bei den Gefäßstrukturen beschreiben überwiegend einen geodätischen sowie modellspezifisch korrekten Richtungsverlauf. Nur vereinzelt konnte eine Diskontinuität beobachtet werden. Notwendig ist eine angemessene Glättung der Skelettlinie sowie der Oberfläche. Für die Skelettlinie zeigte sich dabei ein Wert von 0.7 für den Wichtungsfaktor mit fünf Iterationen und für die Oberflächenglättung ein Wert von  $w = 0.7$  mit einer Iteration als geeignet.

Bei Muskelstrukturen war ebenfalls ein korrektes Vektorfeld zu beobachten. Leichte Diskontinuitäten entstanden aufgrund der ungenauen Approximation der Kontraktionsrichtung durch die erste Hauptkrümmungsrichtung. Zudem gingen durch eine zu starke Oberflächenglättung charakteristische Oberflächendetails verloren. Ein Verzicht auf eine Glättung erbrachte hier teilweise genauere Ergebnisse.

Für Organe und Knochenstrukturen musste das Modell ausreichend geglättet werden, um starke Verwirbelungen und Diskontinuitäten zu reduzieren. Hier zeigte sich ebenfalls

ein Wert von 0.7 für den Wichtungsfaktor  $w$  mit fünf Iterationen als angemessen. Dadurch verbesserte sich der geodätische Richtungsverlauf. Markante Oberflächenverläufe wurden dennoch angemessen beschrieben. Auftretende Strudel konnten zusätzlich durch eine Glättung des Vektorfeldes über eine  $k$ -Nachbarschaft kompensiert werden. Dabei zeigt sich ein topologischer Radius von  $k = 3$  als angebracht. Da jedoch bei Organen und Knochen keine Modellinformationen in die Erzeugung des Vektorfeldes einfließen, waren gegenüber typischen anatomischen Illustrationen teilweise Richtungsabweichungen festzustellen.

## 6.2 Schraffierungen

Dieser Abschnitt präsentiert die erzeugten Schraffuren auf den Oberflächen der vorgestellten anatomischen Strukturen. Beurteilt werden soll hierbei der Grad an Texturverzerrung und -nähten in der Visualisierung. Untersucht werden soll dabei der Einfluss des Parameters  $\alpha_{divergence}$ , welcher den Schwellwerte bezüglich der Winkelabweichung im Vektorfeld darstellt (vgl. Abbruchkriterium 1 im Abschnitt 4.4.2 auf Seite 91).

Für die Ausrichtung der *Patches* im Parameterraum wurde die jeweiligen Vektorfelder anhand der ermittelten Parametereinstellungen aus Abschnitt 6.1.4 verwendet. Zur besseren Beurteilung der Ergebnisse erfolgt die Texturierung auf den Strukturen zunächst einfach und anschließend beleuchtungsabhängig. Weiterhin präsentiert dieser Abschnitt auch die Kombination mancher Strukturen mit anderen anatomischen Modellen in einer integrierten Visualisierung.

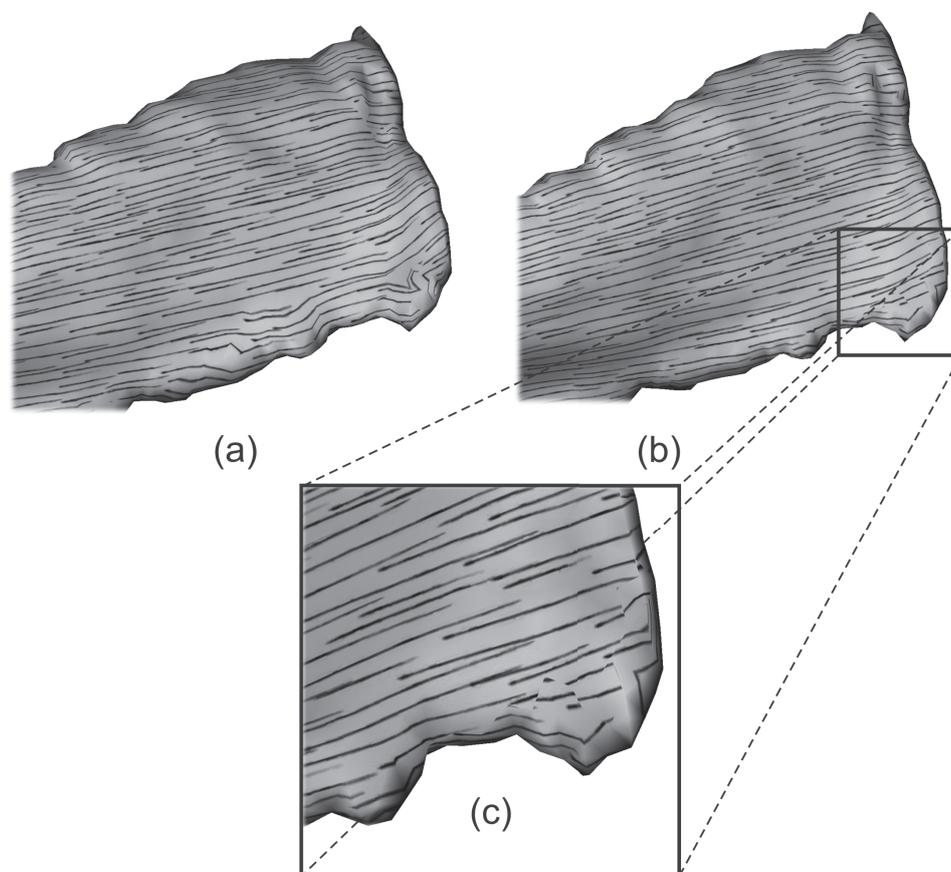
### 6.2.1 Muskelstrukturen

Die Abb. 6.11 zeigt die texturierten Oberflächen der beiden Halsmuskeln. Als Parametereinstellungen sind hierbei gewählt worden  $\alpha_{divergence} = 20^\circ$  und  $\alpha_{rotation} = 10^\circ$ . In Abb. 6.11 (a) ist die abgebildete Schraffurtextur auf der ersten Halsmuskeloberfläche zu sehen. Erkennbar ist dabei, dass die Schraffurlinien der Textur dem zugrundeliegenden Vektorfeld aus Abb. 6.6 (a) folgen. Weiterhin sind zunächst keine auffälligen Texturnähte oder -verzerrungen über der Oberfläche zu beobachten.

Die Texturierung der beiden Muskelstrukturen durch das *TAM*-Konzept, ist in Abb. 6.11 (b) und (c) zu sehen. Mit Hilfe der weichen Überblendung von unterschiedlich dichten Schraffurtexturen wird die Beleuchtungssituation gut wiedergegeben und zeigt eine attraktive Visualisierung. Entsprechend den Beobachtungen aus den illustrativen Darstellungen werden insbesondere dunkle Regionen durch Kreuzschraffuren beschrieben. In den hellen Bereichen vermitteln die dort verlaufenden Schraffurlinien die lokale Oberflächenkrümmung. Letztere ist jedoch in dunklen Regionen aufgrund der dichten Schraffurtexturen weniger wahrnehmbar.

**Abb. 6.11:** Schraffierung der verwendeten Muskelstrukturen: (a) Die einfache Texturierung der Muskeloberfläche zeigt den erwarteten Richtungsverlauf und weist keine auffälligen Verzerrungen oder Nähte auf. In (b) und (c) ist jeweils eine beleuchtungsabhängige Texturierung verwendet worden, die einen ansprechenden illustrativen Charakter erzeugt.

Aufgrund des lokal variierenden Vektorfeldes sind am markierten Muskelrand von Abb. 6.11 (a) jedoch Texturverzerrungen zu beobachten, welche zusätzlich eine leichte Richtungsänderung im Schraffurverlauf bewirken. In Abb. 6.12 (a) ist einer der beiden markierten Bereiche vergrößert dargestellt. Die Verzerrung kommt in Folge der Mittelung der Parameterkoordinaten von sich überlagernden *Patches* zustande. In diese sind aufgrund des hohen Schwellwertes von  $\alpha_{divergence}$  Dreiecke aufgenommen worden, deren Richtungsvektor entsprechend stark abweicht vom Richtungsvektor des Referenzdreieckes. Daraufhin ist die Ausrichtung im Parameterraum zu stark, so dass die spätere Mittelung der Parameterkoordinaten für die betroffenen Eckpunkte über einen relativ großen Parameterbereich erfolgt. In Abb. 6.12 (b) wurde daher  $\alpha_{divergence}$  auf  $3^\circ$  gesetzt, wodurch die Texturverzerrungen sichtbar aufgehoben sind. Allerdings erhöhte sich dabei die Anzahl der zu erzeugenden *Patches* auf das doppelte, was zu einem linearen Anstieg der Vorverarbeitungszeit und des Speicherverbrauchs führte.

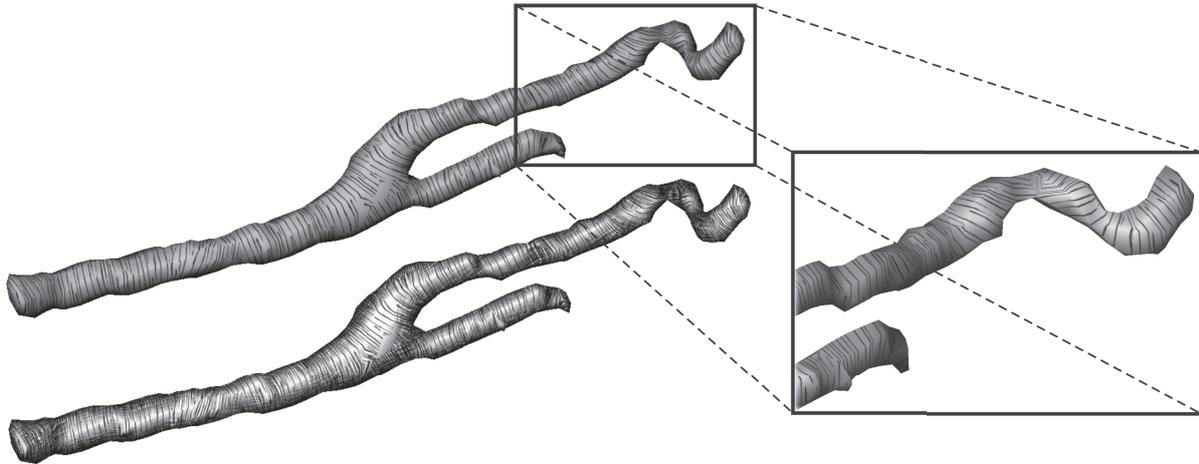


**Abb. 6.12:** Die Texturverzerrungen in (a), aufgrund lokal abweichender Richtungsvektoren, werden durch Setzen des Schwellwertes  $\alpha_{divergence}$  auf  $3^\circ$  in (b) kompensiert. Erkennbar sind in (c) Texturnähte durch Zusammentreffen von *Patches*, die im Objektraum jedoch nicht im Parameterraum benachbart sind.

Bezüglich der aufgetretenen Artefakte zeigt Abb. 6.12 (c) zusätzlich einen vergrößerten Bereich von Abb. 6.12 (b). In diesem sind die Texturnähte zu erkennen, die aufgrund zu weit entfernter Parameterkoordinatenpaare eines Eckpunktes entstanden sind. Diese entsprechen dem Fall 2 „Patches im Objekt- und Parameterraum sind nicht inzident“ aus Abschnitt 4.4.4 von Seite 100. Für deren Behandlung konnte zwar ein Konzept entwickelt werden, jedoch fehlt für diese Arbeit noch dessen Umsetzung.

### 6.2.2 Gefäße

Bei den Gefäßstrukturen zeigt die Abb. 6.13 zunächst die schraffierte Halsschlagader. Neben einer einfachen Texturierung, ist ebenfalls eine beleuchtungsabhängige Darstellung zu sehen. Die Ergebnisse lassen auch hier einen korrekten Richtungsverlauf der

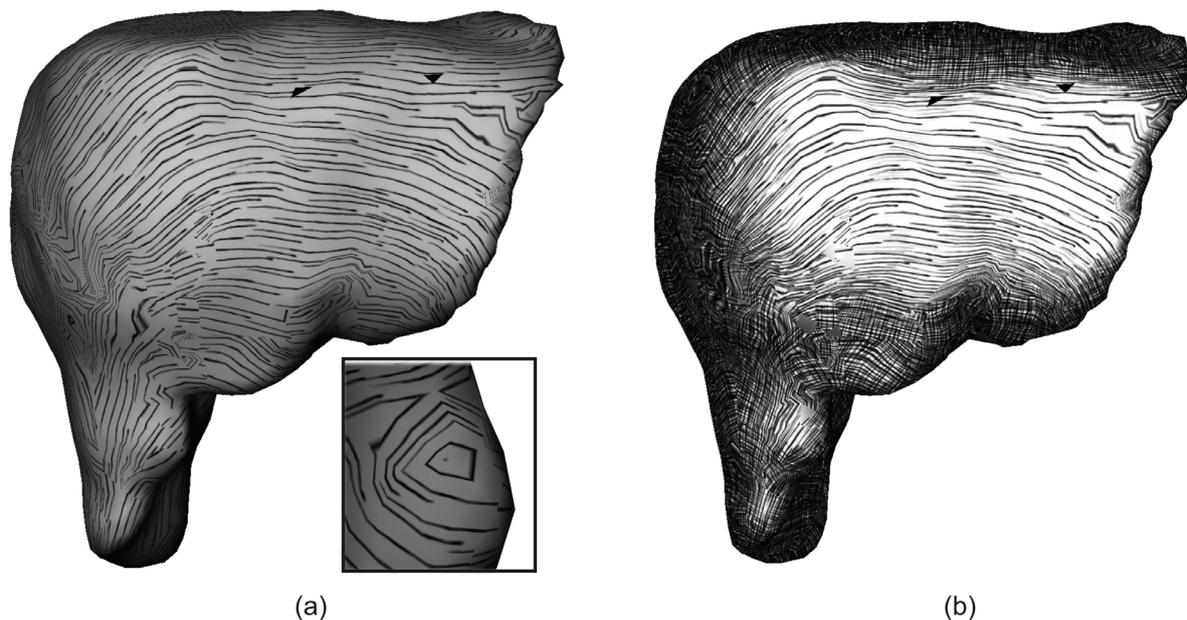


**Abb. 6.13:** Die Schraffierung der Halsschlagader zeigt ebenfalls eine korrekte Ausrichtung der Schraffurtexturen. Der vergrößerte Ausschnitt zeigt auch keine sichtbaren Texturverzerrungen in dem rauen Oberflächenverlauf. Zusätzlich ist die beleuchtungsabhängige Texturierung dargestellt.

Schraffurlinien erkennen bei gleichzeitig kaum wahrnehmbarer Texturverzerrung. Letzteres wird dabei durch den vergrößerten Ausschnitt des Gefäßabschnittes bestätigt, der eine besonders raue Oberfläche aufweist.

### 6.2.3 Organe und Knochenstrukturen

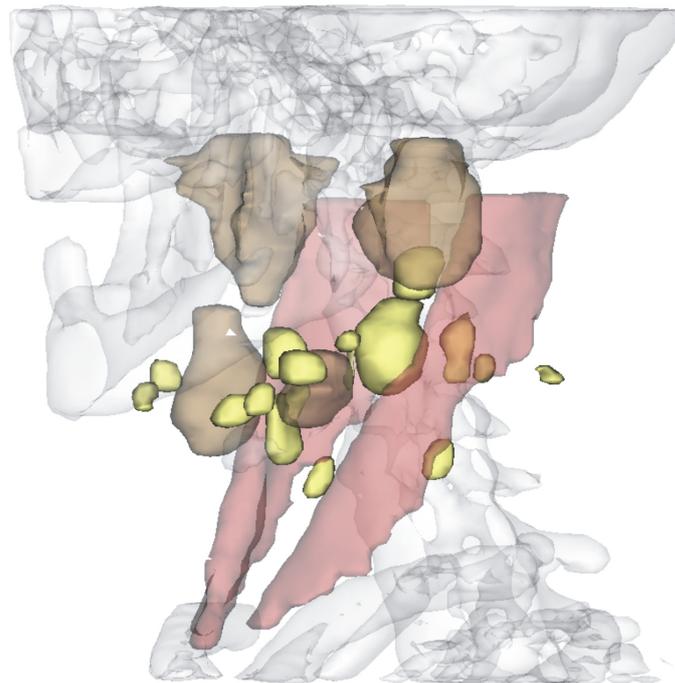
Aufgrund des momentanen, hohen Speicherverbrauches während der Parametrisierung, konnten die unter Abb. 6.8 gezeigten und hochaufgelösten Oberflächen, auf dem hier verwendeten System, nicht vollständig texturiert werden. Allgemein zeigte sich dies bei Strukturen, deren Dreiecksanzahl mehr als 10 000 betrug, wodurch auch der Gefäßbaum betroffen war. Die Abb. 6.14 zeigt daher eine andere Leberoberfläche, mit einer reduzierten Polygonanzahl. Dargestellt ist in (a) zunächst die einfache Texturierung. Erkennbar ist der geodätische Richtungsverlauf der Schraffurlinien, welcher entlang der Oberflächenkrümmung verläuft. Vereinzelt sind Texturverzerrungen zu beobachten aufgrund von Diskontinuitäten im Vektorfeld. Das *Inset* in (a) zeigt einen texturierten Strudel, der auf der Hinterseite der Leber entstanden ist. Dieser entspricht einer Singularität im Vektorfeld und verursacht starke Verzerrungen aufgrund einer erhöhten Mittelung im Parameterraum. Die Abb. 6.14 (b) präsentiert die Leber wieder mit Hilfe der beleuchtungsabhängigen Texturierung. Hierbei sind einzelne Verzerrungen aufgrund der Texturüberblendungen weniger sichtbar.



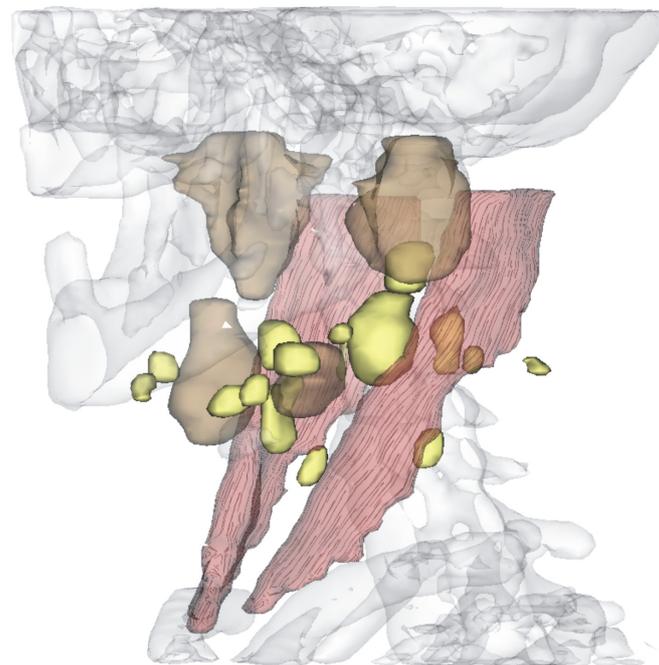
**Abb. 6.14:** (a) Die einfache Texturierung einer Leberoberfläche weist nur an wenigen Stellen sichtbare Texturverzerrungen auf. Markanter sind diese im Bereich von Singularitäten im Vektorfeld, was durch das *Inset* verdeutlicht wird. (b) Die Anwendung der beleuchtsabhängigen Texturierung mindert die sichtbare Wahrnehmung einzelner Texturverzerrungen.

#### 6.2.4 Kombinierte Oberflächendarstellung

Ein Anwendungsbeispiel für eine kombinierte Oberflächendarstellung, bei der das entwickelte Verfahren eingesetzt wurde, soll ebenfalls hier präsentiert werden. Die Abb. 6.15 (a) zeigt dazu einen patientenspezifischen Halsbereich, der von der Seite betrachtet wird. Dargestellt sind die Knochenstrukturen (grau), Drüsen (braun), Lymphknoten (gelb) sowie zwei Halsmuskelstrukturen (rot). Aufgrund der gewählten Perspektive mussten in Abb. 6.15 (a) die Halsmuskeln halbtransparent dargestellt werden, um die Sicht auf die dahinter liegenden Lymphknoten nicht zu verdecken. Dadurch ist allerdings der Oberflächenverlauf der Halsmuskeln schwer erkennbar. Die Abb. 6.15 (b) zeigt die Halsmuskeln dagegen mit einer opaken Schraffierung. Hierbei wird die Sicht auf die dahinterliegenden und relevanten Strukturen nicht behindert. Markante Oberflächenverläufe werden jedoch stärker wahrgenommen, als in der oberen Darstellung, wodurch sich der räumliche Eindruck verbessert. Ebenfalls ergibt sich daraus eine erleichternde Strukturidentifizierung der Muskeln gegenüber den umliegenden Strukturen.



(a)



(b)

**Abb. 6.15:** Anwendungsbeispiel einer kombinierten Oberflächenvisualisierung des Halsbereiches, in der die Schraffierung integriert wurde: (a) Darstellung des Knochens (grau), der Drüsen (braun), der Lymphknoten (gelb) sowie zweier Halsmuskeln (rot). Die halbtransparente Darstellung der Muskeln vermittelt keine ausreichenden Informationen über den Oberflächenverlauf. (b) Durch Abbildung der opaken Schraffur erfolgt eine bessere Oberflächenwahrnehmung bei gleichzeitig erhalten gebliebener Sicht auf die dahinterliegenden Strukturen.

### 6.2.5 Schlussfolgerung

Bei der Parametrisierung wurde ebenfalls ein robustes Verhalten gegenüber rauen Oberflächenbereichen zu beobachten. Die resultierenden Schraffurlinien verliefen entlang des zugrundeliegenden Vektorfeldes, wodurch sie korrekte Informationen über den lokalen Oberflächenverlauf vermitteln. Texturverzerrungen waren nur aufgrund von starken Diskontinuitäten im Vektorfeld zu beobachten und resultierten nicht aus der Modellrauigkeit. Eine Reduzierung der Verzerrungen konnte durch das Setzen eines geringeren Schwellwertes für  $\alpha_{divergence}$  erreicht werden. Hierbei zeigte sich ein Wert von  $3^\circ - 5^\circ$  für alle verwendeten Oberflächen als geeignet. Eine Reduktion auf  $0^\circ$  wird nicht vorgeschlagen, da sich dadurch das Ergebnis nicht sichtbar verbesserte. Jedoch stieg damit die Anzahl von zusätzlich erzeugten *Patches* sowie der verbundene Speicherverbrauch stark an. Letzteres verhinderte dabei die Parametrisierung von Oberflächen mit mehr als 10 000 Dreiecken. Im Bereich von Singularitäten, beispielsweise bei Strudeln im Vektorfeld, konnten die Texturverzerrungen nicht ausreichend vermindert werden. Dies resultiert aus der starken Mittelung der Parameterkoordinaten, die bei Dreiecken in diesen Bereichen notwendig waren.

## 6.3 Laufzeitanalyse

Das Kapitel schließt mit einer Laufzeitanalyse des hier verwendeten Verfahrens. Die Analyse bezieht sich dabei auf die jeweilige Vorverarbeitungszeit bei den verschiedenen Strukturen sowie die jeweiligen *Frames per Second (fps)* während der Visualisierung. Eine Übersicht der ermittelten Werte, welche mit dem angegebenen System aus Abschnitt ?? erzielt wurden, zeigt die Tabelle 6.1. Die Vorverarbeitung beinhaltet hierbei die Konvertierung der OPEN INVENTOR-Szene in die WEM-Datenstruktur, die Generierung des Vektorfeldes sowie die Oberflächenparametrisierung. Anzumerken ist hierbei, dass alle Zeiten (in Sekunden), bis auf die WEM-Konvertierung, im *Debug*-Modus von MEVIS-LAB gemessen werden mussten. Der Grund liegt hierfür in einem instabilen Verhalten des Modules WEMCurvature im *Release*-Modus, wodurch keine Krümmungsinformationen erzeugt werden konnten. Die einzelnen Vorverarbeitungszeiten stellen daher erhöhte Werte dar und nicht die tatsächlich erreichbaren. Sie zeigen aber den relativen Aufwand für die jeweiligen Strukturen, entsprechend ihrer Dreiecksanzahl und Parametereinstellungen.

Zunächst ist erkennbar, dass für alle Strukturen ein relativ geringer Zeitaufwand für die WEM-Konvertierung notwendig ist. Dieser steigt für die Erzeugung des Vektorfeldes an und setzt sich dabei aus der Zeit für die Krümmungsapproximation, die Bestimmung der Vorzugsrichtung sowie der anschließende Kombinierung und Optimierung zu dem Vektorfeld zusammen. In der Tabelle steht neben der jeweiligen Gesamtzeit in Klammern der Anteil für die Krümmungsapproximation. Auffällig sind dabei die hohen Zeiten bei der Leber und der Knochenstruktur. Dies ist zum einen auf die hochaufgelösten Dreiecksnetze

Struktur	Anzahl Dreiecke	WEM-Konver- tierung (s)	Generierung Vektorfeld (s)	Parametrie- sierung (s)	<i>fps</i>
Halsschlagader	878	1	(1) 3	10	(66) 146
Gefäßbaum	81668	5	(20) 370	-	-
Halsmuskel (a)	4723	1	(3) 10	65	(33) 155
Halsmuskel (b)	2810	1	(3) 7	60	(43) 170
Leber	36281	3	(107) 169	70*	(35) 150*
Knochen	90696	5	(275) 630	-	-

**Tabelle 6.1:** Ermittelte Vorverarbeitungszeiten und Darstellungsgeschwindigkeiten des entwickelten Verfahrens für die verschiedenen Strukturen. Die in Klammern stehenden Werte geben bei der Vektorfeldgenerierung den Anteil für die Krümmungsapproximation an sowie in der letzten Spalte die gemessenen *fps* mittels *TAM*-Texturen.

zurückzuführen und zum anderen auf den *BiQuadBezOP*-Operator mit der erweiterten Nachbarschaft.

Im Rahmen dieser Arbeit konnten zunächst nur die Halsschlagader sowie die beiden Halsmuskeln parametrisiert werden. Aufgrund des momentanen, hohen Speicherverbrauches, war dies für eine vollständige Parametrisierung des hochaufgelösten Gefäßbaumes, der Leber und der Knochenstruktur nicht möglich. Der mit (\*) markierte Zeitwert bei der Leber bezieht sich auf eine reduzierte Polygonanzahl von ca. 4300 Dreiecken, wodurch die Oberfläche komplett texturiert werden konnte.

Die letzte Spalte aus Tabelle 6.1 zeigt die *fps*, die während der interaktiven Visualisierung gemessen wurden. Erkennbar ist eine ausreichend hohe Darstellungsgeschwindigkeit, wodurch eine echtzeitfähige Visualisierung gegeben ist. Die Werte in Klammern stehen dabei für die Texturierung mittels dem *TAM*-Konzept. Aufgrund der Texturüberblendungen ist hier ein erhöhter Berechnungsaufwand notwendig. Jedoch zeigen die gemessenen Werte, dass auch hier eine flüssige Darstellung gegeben ist.

Insgesamt ist der Vorverarbeitungsaufwand vertretbar und würde sich erheblich reduzieren, wenn das Verfahren im *Release*-Modus durchführbar ist. Von Vorteil ist, dass die Vorverarbeitung pro Struktur nur einmal durchgeführt werden muss, da die ermittelten Parameterkoordinaten zu der anatomischen Oberfläche abgespeichert werden kann.

## 7 Zusammenfassung und Ausblick

Das Ziel der vorliegenden Arbeit bestand in der Echtzeitschraffierung von anatomischen Oberflächenstrukturen innerhalb einer interaktiven Visualisierung. Dazu wurde ein Verfahren entwickelt, welches auf Grundlage eines krümmungs- und modellbasierten Vektorfeldes sowie eines texturbasierten Ansatzes die Schraffuren auf der Oberfläche erzeugt. Die Anwendung auf unterschiedliche patientenspezifische Oberflächen stellt dabei einen neuen Beitrag für die medizinische Visualisierung dar.

Die Grundlage für den Entwurf eines solchen Verfahrens bildete zunächst die Analyse von Schraffierungen als bewährte, illustrative Darstellungstechnik. Neben stilistischen und wahrnehmungspsychologischen Aspekten wurde dabei der Einsatz in anatomischen Illustrationen untersucht. Es konnte insbesondere gezeigt werden, dass sich der Linienverlauf aus Krümmungs- und Modellinformationen zusammensetzt. Diese Informationen wurden in der vorliegenden Arbeit aus Datensätzen von Gefäßen, Muskeln sowie Organen und Knochenstrukturen automatisch extrahiert und erfolgreich zu einem Vektorfeld kombiniert. Dadurch konnte ein für den Betrachter zu erwartender und geodätischer Richtungsverlauf definiert werden, der zusätzlich charakteristische Oberflächenverläufe beschreibt.

Anschließend erfolgte eine Parametrisierung der anatomischen Oberflächen durch überlagernde *Patches* nach der Methode von PRAUN U. A. [2000]. Mit Hilfe dieses Ansatzes erfolgt eine lokale Ausrichtung von Texturen durch Integration des zugrundeliegenden Vektorfeldes. Die meisten der dabei entstehenden Texturnähte konnten durch Modifikation des Verfahrens aufgelöst werden. In diesem Zusammenhang wurde ein Konzept entwickelt, welches es ermöglicht, die noch vorhandenen Texturnähte, aufgrund von Schnittkanten während der Parametrisierung, zu beseitigen.

Die Ergebnisse dieser Arbeit zeigen, dass das Verfahren eine echtzeitfähige und *Frame*-kohärente Schraffierung der anatomischen Oberflächen innerhalb einer interaktiven Visualisierung ermöglicht. Neben Einzelvisualisierungen der Strukturen wurde die Anwendung der Schraffierung innerhalb einer kombinierten Darstellung präsentiert. Die erzeugten Vektorfelder wiesen allgemein ein robustes Verhalten gegenüber den teilweise sehr rauen anatomischen Oberflächen auf. Nur an wenigen Stellen, vor allem bei Organen und Knochen, waren Singularitäten und Diskontinuitäten zu beobachten. Bezüglich der Parametrisierung waren diese kritischen Punkte im Vektorfeld die Ursache für sichtbare Texturverzerrungen. Die Verwendung der beleuchtungsabhängigen Texturierung mittels dem *TAM*-Konzept von PRAUN U. A. [2001] ergab nicht nur eine visuell ansprechende Illustration der verwendeten Oberflächen, sondern verringerte auch die Wahrnehmung einzelner Texturartefakte.

Die Interaktion des Benutzers beschränkt sich auf die Einstellung von wenigen Parametern, für die im Rahmen dieser Arbeit Standardwerte festgelegt werden konnten. Die Vorverarbeitungszeit für die Erzeugung der Vektorfelder sowie für die Oberflächenparametrisierung ist vertretbar und deren Durchführung nur einmal für eine anatomische Struktur notwendig.

Das angestrebte Ziel einer echtzeitfähigen Schraffierung von patientenspezifischen, anatomischen Oberflächen im Rahmen einer interaktiven Visualisierung konnte somit erreicht werden.

## 7.1 Ausblick und Verbesserungsvorschläge

In dieser Arbeit wurde gezeigt, dass die Skalierbarkeit in der Schraffierung, als weitere wichtige Anforderung an eine interaktive Visualisierung, notwendig ist. In dem hier entwickelten Verfahren muss diese noch integriert werden. Dazu ist die Erstellung von *MipMap*-Texturen nach dem *TAM*-Konzept notwendig. Die hier verwendeten *Hardware Shader* müssen entsprechend angepasst werden, so dass der entfernungsabhängige Zugriff auf die entsprechend aufgelösten Texturen und deren Überblendung erfolgen kann.

Verbesserungen des Verfahrens bestehen zum einen darin, den Parametrisierungsansatz dahingehend zu modifizieren, dass Singularitäten und Diskontinuitäten im Vektorfeld besser behandelt werden. Gerade bei unterschiedlich großen Dreiecken in den betreffenden Gebieten erfolgt daraus eine starke Mittelung über die verschiedenen Parameterkoordinaten eines Eckpunktes. Eine Lösung dieses Problems besteht darin, die Größe der Dreiecke zu regularisieren. Erreicht werden kann dies beispielsweise durch ein *Remeshing* der betroffenen Dreiecke oder einem *Subdivision* von großen Dreiecken. Zum anderen verursacht das momentan umgesetzte Parametrisierungsverfahren einen hohen Speicherverbrauch. Dadurch konnten auf dem verwendeten System keine hochaufgelöster Oberflächen schraffiert werden. Erforderlich ist es daher, die Implementierung diesbezüglich zu optimieren.

Hinsichtlich der erzeugten Vektorfelder kann das Verfahren dahingehend erweitert werden, dass diese für weitere Strukturen modellspezifisch generiert werden. Gerade bei Organen oder Knochenstrukturen ist dies wünschenswert, da gezeigt werden konnte, dass Krümmungsinformationen dafür allein nicht ausreichen. Angebracht wären zum einen Strukturen, bei denen sich eine Skelettlinie automatisch extrahieren lässt und diese für eine modellspezifische Vorzugsrichtung verwendet werden kann. Beispielsweise eignen sich hierbei Rippen oder Röhrenknochen. Zum anderen wären auch Strukturen denkbar, deren Form sich durch einen einfacheren Zwischenkörper beschreiben lässt. Auf diesem erfolgt zunächst eine modellspezifische und dabei *Frame*-kohärente Vorzugsrichtung, die dann auf die komplexere Struktur übertragen wird. Dieser Ansatz ist angelehnt an die indirekte Parametrisierung des *2D-Texture Mapping*. Möglich ist auch die Definierung von einzelnen Vorzugsrichtungen an speziellen Strukturbereichen durch den Benutzer.

Zwischen diesen erfolgt anschließend eine Interpolation für die Zwischenbereiche. Dieses Vorgehen wurde von einigen der untersuchten Schraffierungsverfahren auch verwendet. Jedoch ist hierfür die zusätzliche Interaktion durch den Benutzer erforderlich, was je nach Einsatzgebiet nicht erwünscht ist.

Bei Verwendung der beleuchtungsabhängigen Texturierung fiel auf, dass durch die verwendeten Schraffurtexturen der Oberflächenverlauf in den dunklen Objektbereichen weniger erkennbar ist als in den helleren Gebieten. Der Grund liegt hierfür in den komplexen und dichten Linienmustern. Daraus lassen sich zwei Erweiterungen des Verfahrens ableiten. Die erste bezieht sich auf die Verwendung von prozedural erzeugten Schraffurmustern statt Bildtexturen. In der vorgestellten Arbeit von FREUDENBERG U. A. [2002] werden u. a. Schraffurlinien damit erzeugt. Die lokale Beleuchtung wird dabei anhand der Liniendicke beschrieben, wodurch komplexe Linienmuster vermieden werden. Attraktiv erscheint in diesem Zusammenhang auch die Integration weiterer Gestaltungsparameter wie sie in Abschnitt 2.2.1 auf Seite 9 ausgearbeitet wurden. Beispielsweise könnte hierbei zusätzlich die lokale Krümmungstärke in den Linienstil einfließen, wie dies auch bei einigen Illustrationen beobachtet wurde.

Die zweite Erweiterung bezieht sich auf die Platzierung der Schraffurlinien. Zurzeit unterstützt das Verfahren zur eine beleuchtungsabhängige Platzierung. Neben der lokalen Helligkeit beeinflussten in einigen anatomischen Zeichnungen weitere Aspekte das Setzen der Linien. Zu nennen sind beispielsweise die Differenzierung von Fokus-Kontext-Strukturen oder auch wie weit sich der zu schraffierende Bereich vom Objektrand bzw. einer Fokusstruktur befindet. Außerdem werden beim Übergang von Objektrand zu Objekthintergrund bewusst weniger Linien gezeichnet, um diesen besser hervorzuheben. Aufgrund dieser zusätzlichen Parameter konnte bewusst die Aufmerksamkeit des Betrachters gelenkt werden und die Interpretation des Bildes wurde erleichtert. Daher ist es wünschenswert, beispielsweise in einer fortführenden Arbeit, diese Aspekte zu berücksichtigen und die Schraffurplatzierung dadurch gezielt zu steuern. Dies macht es notwendig, die genannten und eventuell weitere Parameter im Vorfeld zu definieren und diese z. B. in einer Parameterkarte zu speichern. Während der Schraffierung kann dabei auf diese Karte zugegriffen und je nach Wichtung der Parameter im aktuell betrachteten Bildbereich mehr oder weniger viele Linien gezeichnet werden.

Sinnvoll ist zudem auch die Durchführung einer genauen Evaluierung, inwieweit die hier erzeugten Schraffierungen auch tatsächliche die bisherigen medizinischen Visualisierungen bereichern können. Untersucht werden kann hierbei, welchen Einfluss diese auf die Wahrnehmung des Oberflächenverlaufes und Struktur differenzierung besitzen. Dabei ist eine Integration des entwickelten Verfahrens in das *MedicalExplorationToolkit* (METK) sinnvoll. Dieses Kommunikationswerkzeug, welches in der Arbeitsgruppe „Visualisierung“ entwickelt wurde, dient zur Verarbeitung und Exploration von medizinischen Fällen und ist in MEVISLAB integriert. Durch Erweiterung der bisherigen Darstellungsstile um Schraffierung, können die damit visualisierten Oberflächen gegenüber anderen Stilen und der obigen Fragestellung bewertet werden.



## Literaturverzeichnis

- [Akenine-Möller und Haines 2002] AKENINE-MÖLLER, Thomas ; HAINES, Eric: *Real-Time Rendering*. 2nd. A K Peters, 2002
- [Andrews 2006] ANDREWS, Bill: Introduction to Perceptual Principles in Medical Illustrations. In: *Illustrative Visualization for Medicine and Science*, 2006, S. 8–30
- [Baer 2005] BAER, Alexandra: *Hardwaregestütztes Stippling von medizinischen Oberflächenmodellen*, Otto-von-Guericke University of Magdeburg, Department of Computer Science, Diplomarbeit, 2005
- [Baumgart 1972] BAUMGART, Bruce G.: Winged Edge Polyhedron Representation. In: *National Computer Conference (1972)*
- [Bier und Sloan 1986] BIER, Eric A. ; SLOAN, Ken R.: Two-Part Texture Mapping. In: *IEEE Computer Graphics and Applications* 6 (1986), September, Nr. 9, S. 40 – 53
- [Card und Mitchell 2002] CARD, Drew ; MITCHELL, Jason L.: Non-Photorealistic Rendering with Pixel and Vertex Shaders. In: *ShaderX: Vertex and Pixel Shaders Tips and Tricks* (2002), S. 319–333
- [Dong und Clapworthy 2005] DONG, Feng ; CLAPWORTHY, Gordon: Volumetric Texture Synthesis for Non-Photorealistic Volume Rendering of Medical Data. In: *The Visual Computer* 21 (2005), Nr. 7, S. 463–473
- [Dong u. a. 2003] DONG, Feng ; CLAPWORTHY, Gordon ; LIN, Hai ; KROKOS, Meleagros A.: Nonphotorealistic Rendering of Medical Volume Data. In: *IEEE Computer Graphics and Applications* 23 (2003), Nr. 4, S. 44–52
- [Ebert und Svakhine 2006] EBERT, David ; SVAKHINE, Nikolai: Volume Illustration for Medicine and Flows. In: *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*. New York, NY, USA : ACM Press, 2006, S. 4
- [Eck u. a. 1995] ECK, Matthias ; DEROSE, Tony ; DUCHAMP, Tom ; HOPPE, Hugues ; LOUNSBERRY, Michael ; STUETZLE, Werner: Multiresolution Analysis of Arbitrary Meshes. In: *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM Press, 1995, S. 173–182

- [Eckardt 2003] ECKARDT, Bernd: *Geometriebasiertes Echtzeit-Halbton-Rendering*, Otto-von-Guericke University of Magdeburg, Department of Computer Science, Diplomarbeit, 2003
- [Floater 1997] FLOATER, Michael S.: Parametrization and Smooth Approximation of Surface Triangulations. In: *Computer Aided Geometric Design* 14 (1997), Nr. 3, S. 231–250
- [Floater und Hormann 2004] FLOATER, Michael S. ; HORMANN, Kai: Surface Parameterization: a Tutorial and Survey. In: DODGSON, Neil A. (Hrsg.) ; FLOATER, Michael S. (Hrsg.) ; SABIN, Malcolm A. (Hrsg.): *Advances in Multiresolution for Geometric Modelling*. Springer Verlag, 2004, S. 157–186
- [Foley u. a. 1990] FOLEY, James D. ; DAM, Andries van ; FEINER, Steve K. ; HUGHES, John F.: *Computer Graphics: Principles and Practice*. Addison-Wesley, 1990
- [Freudenberg u. a. 2002] FREUDENBERG, Bert ; MASUCH, Maic ; STROTHOTTE, Thomas: Real-Time Halftoning: A Primitive for Non-Photorealistic Shading. In: DEBEVEC, Paul (Hrsg.) ; GIBSON, Simon (Hrsg.): *Rendering Techniques 2002: Proceedings of the 13th Eurographics Workshop on Rendering (Pisa, Italy, June 26–28, 2002)*. Aire-la-Ville, Switzerland, 2002, S. 227–231, 331
- [Freudenberg u. a. 2004] FREUDENBERG, Bert ; MASUCH, Maic ; STROTHOTTE, Thomas: Real-Time Halftoning: Fast and Simple Stylized Shading. In: KIRMSE, Andrew (Hrsg.): *Game Programming Gems 4*. Charles River Media, 2004, S. 443–440
- [Girshick u. a. 2000] GIRSHICK, Ahna ; INTERRANTE, Victoria ; HAKER, Steven ; LEMOINE, Todd: Line Direction Matters: An Argument for the Use of Principal Directions in 3D Line Drawings. In: *NPAR: Non-Photorealistic Animation and Rendering*, 2000, S. 43–52
- [Goldfeather 2003] GOLDFEATHER, Jack: Understanding Errors in Approximating Principal Direction Vectors / University of Minnesota, Department of Computer Science and Engineering. 2003. – Forschungsbericht
- [Goldfeather und Interrante 2004] GOLDFEATHER, Jack ; INTERRANTE, Victoria: A Novel Cubic-Order Algorithm for Approximating Principal Direction Vectors. In: *ACM Transaction on Graphics* 23 (2004), Nr. 1, S. 45–63
- [Gorla u. a. 2003] GORLA, Gabriele ; INTERRANTE, Victoria ; SAPIRO, Guillermo: Texture Synthesis for 3D Shape Representation. In: *IEEE Transactions on Visualization and Computer Graphics* 9 (2003), Nr. 4, S. 512–524
- [Gotsman u. a. 2003] GOTSMAN, Craig ; GU, Xianfeng ; SHEFFER, Alla: Fundamentals of Spherical Parameterization for 3D Meshes. In: *ACM Transaction on Graphics* 22 (2003), Nr. 3, S. 358–363

- [Hahn u. a. 2003] HAHN, Horst ; LINK, Florian ; PEITGEN, Heinz-Otto: Concepts for Rapid Application Prototyping in Medical Image Analysis and Visualization. In: *Proceedings SimVis, Simulation und Visualisierung: SCS, Erlangen* (2003), S. 283–98
- [Hahn u. a. 2001] HAHN, Horst ; PREIM, Bernhard ; SELLE, Dirk ; PEITGEN, Heinz-Otto: Visualization and Interaction Techniques for the Exploration of Vascular Structures. In: *IEEE Visualization (San Diego, Oktober)*, 2001, S. 395–402
- [Heckbert 1986] HECKBERT, Paul S.: Survey of Texture Mapping. In: *IEEE Computer Graphics and Applications* 6 (1986), Nr. 11, S. 56–67
- [Hertzmann und Zorin 2000] HERTZMANN, Aron ; ZORIN, Denis: Illustrating Smooth Surfaces. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), S. 517–526
- [Hodges 1988] HODGES, Elaine R.: *The Guild Handbook of Scientific Illustration*. Van Nostrand Reinhold, 1988
- [Interrante 1997] INTERRANTE, Victoria: Illustrating Surface Shape in Volume Data via Principal Direction-Driven 3D Line Integral Convolution. In: *SIGGRAPH*, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center; National Technical Information Service, distributor, 1997, S. 109–116
- [Interrante u. a. 1996] INTERRANTE, Victoria ; FUCHS, Henry ; PIZER, Stephen M.: Illustrating Transparent Surfaces with Curvature-Directed Strokes. In: *IEEE Visualization 1996 Proceedings*, 1996, S. 211–218
- [Interrante u. a. 1997] INTERRANTE, Victoria ; FUCHS, Henry ; PIZER, Stephen M.: Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture. In: *IEEE Transactions on Visualization and Computer Graphics* 3 (1997), Nr. 2, S. 98–117
- [Interrante u. a. 2002] INTERRANTE, Victoria ; KIM, Sunghee ; HAGH-SHENAS, Haleh: Conveying 3D Shape with Texture: Recent Advances and Experimental Results. In: *Human Vision and Electronic Imaging VII, SPIE 4662*, 2002, S. 197–206
- [Isenberg u. a. 2003] ISENBERG, Tobias ; FREUDENBERG, Bert ; HALPER, Nick ; SCHLECHTWEG, Stefan ; STROTHOTTE, Thomas: A Developer’s Guide to Silhouette Algorithms for Polygonal Models. 23 (2003), Juli/August, Nr. 4, S. 28–37
- [Isenberg u. a. 2006] ISENBERG, Tobias ; NEUMANN, Petra ; CARPENDALE, Sheelagh ; SOUSA, Mario C. ; JORGE, Joaquim A.: Non-Photorealistic Rendering in Context: An Observational Study. In: *Proceedings of the Fourth International Symposium on Non-Photorealistic Animation and Rendering (NPAR 2006, June 5–7, 2006, Annecy, France)*, ACM Press, 2006, S. 115–126

- [Kim u. a. 2003] KIM, Sunghee ; HAGH-SHENAS, Haleh ; INTERRANTE, Victoria: Showing Shape with Texture: Two Directions Seem Better than One. In: *Human Vision and Electronic Imaging VIII, SPIE 5007*, 2003, S. 332–339
- [Kim u. a. 2004] KIM, Sunghee ; HAGH-SHENAS, Haleh ; INTERRANTE, Victoria: Conveying Shape with Texture: Experimental Investigations of Texture's Effects on Shape Categorization Judgments. In: *IEEE Transactions on Visualization and Computer Graphics* 10 (2004), Nr. 4, S. 471–483
- [Kirk 1992] KIRK, David (Hrsg.): *Graphics Gems III*. Academic Press, 1992
- [Klein u. a. 2000] KLEIN, Allison W. ; LI, Wilmot ; KAZHDAN, Michael M. ; CORR, Wagner T. ; FINKELSTEIN, Adam ; FUNKHOUSER, Thomas A.: Non-photorealistic virtual environments. In: *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM Press/Addison-Wesley Publishing Co., 2000, S. 527–534
- [Krüger u. a. 2005] KRÜGER, Arno ; TIETJEN, Christian ; HINTZE, Jana ; PREIM, Bernhard ; HERTEL, Ilka ; STRAUSS, Gero: Interactive Visualization for Neck Dissection Planning. In: *IEEE/Eurographics Symposium on Visualization (EuroVis)*, 2005, S. 295–302
- [Lake u. a. 2000] LAKE, Adam ; MARSHALL, Carl ; HARRIS, Mark ; BLACKSTEIN, Marc: Stylized Rendering Techniques for Scalable Real-Time 3D Animation. In: *NPAR: Non-Photorealistic Animation and Rendering*, 2000, S. 13–20
- [Lipschutz 1980] LIPSCHUTZ, Martin M.: *Differentialgeometrie: Theorie und Anwendung*. McGraw-Hill, 1980
- [Lorensen und Cline 1987] LORENSEN, William E. ; CLINE, Harvey E.: Marching cubes: A High Resolution 3D Surface Construction Algorithm. In: *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM Press, 1987, S. 163–169
- [Meyer u. a. 2003] MEYER, Mark ; DESBRUN, Mathieu ; SCHRÖDER, Peter ; BARR, Alan H.: Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In: HEGE, Hans-Christian (Hrsg.) ; POLTHIER, Konrad (Hrsg.): *Visualization and Mathematics III*. Heidelberg : Springer-Verlag, 2003, S. 35–57
- [Pastor u. a. 2003] PASTOR, Oscar M. ; FREUDENBERG, Bert ; STROTHOTTE, Thomas: Real-Time Animated Stippling. In: *IEEE Computer Graphics and Applications* 23 (2003), Nr. 4, S. 62–68
- [Piegl und Tiller 1995] PIEGL, Les ; TILLER, Wayne: Algorithm for Degree Reduction of B-spline Curves. In: *Computer-Aided Design* 27 (1995), Nr. 2, S. 101–110

- [Pitz 1984] PITZ, Henry C.: *Ink Drawing Techniques*. Watson-Guption Publications, 1984
- [Praun u. a. 2000] PRAUN, Emil ; FINKELSTEIN, Adam ; HOPPE, Hugues: Lapped Textures. In: *Proceedings of SIGGRAPH 2000* (2000), S. 465–470
- [Praun und Hoppe 2003] PRAUN, Emil ; HOPPE, Hugues: Spherical Parametrization and Remeshing. In: *ACM Transaction on Graphics* 22 (2003), Nr. 3, S. 340–349. – ISSN 0730-0301
- [Praun u. a. 2001] PRAUN, Emil ; HOPPE, Hugues ; WEBB, Matthew ; FINKELSTEIN, Adam: Real-Time hatching. In: *SIGGRAPH*, 2001, S. 581
- [Putz und Pabst 1993a] PUTZ, Reinhard ; PABST, Reinhard: *Sobotta: Atlas der Anatomie des Menschen*. 20. Urban & Schwarzenbach, 1993 (Band 2: Rumpf, Eingeweide, untere Extremitäten)
- [Putz und Pabst 1993b] PUTZ, Reinhard ; PABST, Reinhard: *Sobotta: Atlas der Anatomie des Menschen*. 20. Urban & Schwarzenbach, 1993 (Band 1: Kopf, Hals, obere Extremitäten)
- [Razdan und Bae 2005] RAZDAN, Anshuman ; BAE, MyungSoo: Curvature Estimation Scheme for Triangle Meshes Using Biquadratic Bézier Patches. In: *Computer-Aided Design* 37 (2005), Nr. 14, S. 21
- [Ritter u. a. 2006] RITTER, Felix ; HANSEN, Christian ; DICKEN, Volker ; KONRAD-VERSE, Olaf ; PREIM, Bernhard ; PEITGEN, Heinz-Otto: Real-Time Illustration of Vascular Structures for Surgery. In: *IEEE Transactions on Visualization* 12 (2006), S. 877–884
- [Rogers 1992] ROGERS, Andrew W.: *Textbook of Anatomy*. W.B. Saunders Company, Januar 1992
- [Rössl und Kobbelt 2000] RÖSSL, Christian ; KOBBELT, Leif: Line-Art Rendering of 3D-Models. In: *Proceedings of Pacific Graphics 2000* 00 (2000), S. 87
- [Rössl u. a. 2000] RÖSSL, Christian ; KOBBELT, Leif ; SEIDEL, Hans-Peter: Line Art Rendering of Triangulated Surfaces using Discrete Lines of Curvature. In: SKALA, Vaclav (Hrsg.): *Proceedings of the 8th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG-00)*. Plzen, Czech Republic : University of West Bohemia, 2000, S. 168–175
- [Rost 2004] ROST, Randi J.: *OpenGL Shading Language*. Addison-Wesley Professional, 2004. – with contributions by John M. Kessenich and Barthold Lichtenbelt
- [Saito und Takahashi 1990] SAITO, T. ; TAKAHASHI, T.: Comprehensible Rendering of 3-D Shapes. In: *Computer Graphics* 24 (1990), Nr. 4

- [Salah 2006] SALAH, Zein I.: *Segmentation and Illustrative Visualization of Medical Data*, Eberhard-Karls-Universität Tübingen, Fakultät für Informations- und Kognitionswissenschaften, Dissertation, 2006
- [Salisbury u. a. 1996] SALISBURY, Michael ; ANDERSON, Corin R. ; LISCHINSKI, Dani ; SALESIN, David: Scale-Dependent Reproduction of Pen-and-Ink Illustrations. In: *SIGGRAPH*, 1996, S. 461–468
- [Salisbury u. a. 1994] SALISBURY, Michael P. ; ANDERSON, Sean E. ; BARZEL, Ronen ; SALESIN, David H.: Interactive Pen-and-Ink Illustration. New York, 1994, S. 101–108
- [Samek u. a. 1986] SAMEK, Marcel ; SLEAN, Cheryl ; WEGHORST, Hank: Texture Mapping and Distortion in Digital Graphics. In: *The Visual Computer* 2 (1986), Nr. 5, S. 313–320
- [Schöne 1975] SCHÖNE, Wolfgang: *Differentialgeometrie*. 5. Auflage. BSB Teubner, Leipzig, 1975
- [Schneider und Eberly 2003] SCHNEIDER, Philip J. ; EBERLY, David H. ; DELLAPENTA, Ken (Hrsg.): *Geometric Tools for Computer Graphics*. Morgan Kaufmann, 2003
- [Schulz 2005] SCHULZ, Christian: *Approximation von Krümmungsinformation für die medizinische Visualisierung*, Otto-von-Guericke University of Magdeburg, Department of Computer Science, Diplomarbeit, 2005
- [Schünke u. a. 2005] SCHÜNKE, Michael ; SCHULTE, Erik ; SCHUMACHER, Udo: *Pro-metheus: Hals und Innere Organe*. Georg Thieme, 2005
- [Secord u. a. 2002] SECORD, Adrian ; HEIDRICH, Wolfgang ; STREIT, Lisa: Fast primitive distribution for illustration. In: *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering*. Aire-la-Ville, Switzerland : Eurographics Association, 2002, S. 215–226
- [Selle u. a. 2000] SELLE, Dirk ; SPINDLER, Wolf ; PREIM, Bernhard ; PEITGEN, Heinz-Otto: *Mathematical Methods in Medical Image Processing: Analysis of Vascular Structures for Preoperative Planning in Liver Surgery*. S. 1039–1059, Springer, 2000
- [Sousa 2006] SOUSA, Mario C.: *Overview of NPR for Computerized Illustration*. SIGGRAPH (Veranst.), 2006
- [Sousa u. a. 2005] SOUSA, Mario C. ; EBERT, David S. ; STREDNEY, Don ; SVAKHINE, Nikolai A.: Illustrative Visualization for Medical Training. In: NEUMANN, László (Hrsg.) ; CASASAYAS, Mateu S. (Hrsg.) ; GOOCH, Bruce (Hrsg.) ; PURGATHOFER, Werner (Hrsg.): *Proceedings of the First Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging 2005 (May 18–20, 2005, Girona, Spain)*. Aire-la-Ville, Switzerland : Eurographics Association, 2005, S. 201–208

- [Sousa u. a. 2003] SOUSA, Mario C. ; FOSTER, Kevin ; WYVILL, Brian ; SAMAVATI, Faramarz F.: Precise Ink Drawing of 3D Models. In: *Computer Graphics Forum* 22 (2003), Nr. 3, S. 369–380
- [Sousa und Prusinkiewicz 2003] SOUSA, Mario C. ; PRUSINKIEWICZ, Przemyslaw: A Few Good Lines: Suggestive Drawing of 3D Models. 22 (2003), September, Nr. 3, S. 381–390
- [Sousa u. a. 2004] SOUSA, Mario C. ; SAMAVATI, Faramarz F. ; BRUNN, Meru: Depicting Shape Features with Directional Strokes and Spotlighting. In: *Computer Graphics International*, 2004, S. 214–221
- [Stone und Stone 2000] STONE, Robert J. ; STONE, Judith A. ; KANE, Kevin T. (Hrsg.): *Atlas of Skeletal Muscles*. 3. Mc Graw Hill, 2000
- [Strothotte und Schlechtweg 2002] STROTHOTTE, Thomas ; SCHLECHTWEG, Stefan: *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. San Francisco : Morgan Kaufmann, 2002
- [Sweet und Ware 2004] SWEET, Graeme ; WARE, Colin: View Direction, Surface Orientation and Texture Orientation for Perception of Surface Shape. In: *GI '04: Proceedings of the 2004 Conference on Graphics Interface*. School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada : Canadian Human-Computer Communications Society, 2004, S. 97–106
- [Tarini u. a. 2004] TARINI, Marco ; HORMANN, Kai ; CIGNONI, Paolo ; MONTANI, Claudio: PolyCube-Maps. In: *ACM Transaction on Graphics* 23 (2004), Nr. 3, S. 853–860
- [Taubin 1995] TAUBIN, Gabriel: Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation. In: *IEEE International Conference on Computer Vision*, 1995, S. 902–907
- [Tietjen u. a. 2005] TIETJEN, Christian ; ISENBERG, Tobias ; PREIM, Bernhard: Combining Silhouettes, Surface, and Volume Rendering for Surgery Education and Planning. In: *IEEE/Eurographics Symposium on Visualization (EuroVis)*, 2005, S. 303–310
- [Viola u. a. 2005] VIOLA, Ivan ; GRÖLLER, M. E. ; HADWIGER, Markus ; BÜHLER, Katja ; PREIM, Bernhard ; SOUSA, Mario C. ; EBERT, David S. ; STREDNEY, Don: Illustrative Visualization. In: *IEEE Visualization*, 2005
- [Viola u. a. 2006] VIOLA, Ivan ; SOUSA, Mario C. ; EBERT, David ; ANDREWS, Bill ; GOOCH, Bruce ; TIETJEN, Christian ; EBERT, David S. (Hrsg.) ; SOUSA, Mario C. (Hrsg.): *Eurographics Course: Illustrative Visualization for Medicine and Science*. International Conference on Computer Graphics and Interactive Techniques. 2006

- [Waldeyer und Mayet 1993a] WALDEYER, Anton J. ; MAYET, Anton: *Anatomie des Menschen*. 16. Walter de Gruyter, New York, 1993 (Band 2)
- [Waldeyer und Mayet 1993b] WALDEYER, Anton J. ; MAYET, Anton: *Anatomie des Menschen*. 16. Walter de Gruyter, New York, 1993 (Band 1)
- [Webb u. a. 2002] WEBB, Matthew ; PRAUN, Emil ; FINKELSTEIN, Adam ; HOPPE, Hugues: Fine Tone Control in Hardware Hatching. In: *NPAR: Proceedings of the 2nd International Symposium on Non-Photorealistic Animation and Rendering*. New York, NY, USA : ACM Press, 2002, S. 53–ff
- [Wernecke 1993] WERNECKE, Josie: *The Inventor Mentor: Programming Object-Oriented 3d Graphics with Open Inventor, Release 2*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1993
- [Winkenbach und Salesin 1994] WINKENBACH, Georges ; SALESIN, David H.: Computer-Generated Pen-and-Ink Illustration. In: *SIGGRAPH: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA : ACM Press, 1994, S. 91–100
- [Woolridge und Jenkinson 2004] WOOLRIDGE, Nick ; JENKINSON, Jodie: Visually-Oriented Knowledge Media Design in Medicine. In: *KMDI: Knowledge Media Design Institute, Biomedical Communications*. University of Toronto, 2004
- [Zander u. a. 2004] ZANDER, Johannes ; ISENBERG, Tobias ; SCHLECHTWEG, Stefan ; STROTHOTTE, Thomas: High Quality Hatching. In: *Computer Graphics Forum* 23 (2004), Nr. 3, S. 421–430