

Dreidimensionale Visualisierung und morphologische Analyse der tubulären Struktur der Lunge

Thorsten Denhard

Diplomarbeit

zur Erlangung des akademischen Grades
Diplom-Informatiker (FH)

Vorgelegt dem
Fachbereich Mathematik, Naturwissenschaften und Informatik
an der Fachhochschule Gießen-Friedberg

im Dezember 2003

Referent: Prof. Dr. Volker Klement
1. Korreferent: Prof. Dr. Oskar Hoffmann
2. Korreferent: Prof. Dr. Andres Kriete

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen der Anatomie und Bildverarbeitung	4
2.1	Lungenanatomie	4
2.1.1	Bronchialbaum	4
2.1.2	Respiratorische Zone	7
2.2	Digitalisierung geometrischer Baumobjekte	8
2.2.1	Röntgen-Computertomografie	8
2.2.2	Extraktion von baumartigen Strukturen	9
2.2.2.1	Segmentierung	9
2.2.2.2	Mittelachsentransformation	10
2.2.2.3	Weitere Abstraktionen	12
3	Konzeption des geometrischen Modells	14
3.1	Repräsentationsvarianten von Baumstrukturen	14
3.1.1	Volumen-Rendering	14
3.1.2	Oberflächen-Rendering	15
3.1.3	Linienhafte Darstellung	17
3.2	Das Röhrenmodell	18
3.2.1	Eingangsdaten	18
3.2.1.1	Gerichtete Graphen	18
3.2.1.2	Geforderte Baumstruktur	19
3.2.1.3	Attribute von Knoten und Kanten im Mevis-Graphen	20
3.2.2	Das Basismodell	21
3.2.3	Erweiterung der Geometrie	24
3.3	Visualisierung dynamischer Daten	30
3.4	Statistik	32
3.4.1	Eigenschaften von Ästen	33
3.4.2	Erfassung und Verarbeitung	36

4	Implementierung	38
4.1	Datenstrukturen	38
4.1.1	Mevis-Dateiformat	38
4.1.2	Die Klasse <code>MevisFile</code>	39
4.1.3	Die Klassen <code>Branch</code> und <code>Lung</code>	40
4.1.4	Alternatives Binärformat	43
4.2	Rendering mit OpenGL	45
4.2.1	Visualisierung des Lungenparenchyms	47
4.2.2	Detailgrad der Darstellung	48
4.3	GUI-Programmierung mit dem Qt-Toolkit	49
4.4	Interaktion	49
4.4.1	Navigation im Raum	51
4.4.2	Selektion von Ästen	51
4.5	Statistik	55
4.6	Schnittstelle zum Simulationsmodul	56
5	Ergebnisse und Diskussion	59
5.1	Quantitativer Längenvergleich	59
5.2	Quantitativer Durchmesservergleich	62
5.3	Performance-Betrachtungen	63
5.4	Erweiterungen, weitere Anwendungsbereiche	65
6	Zusammenfassung/Summary	67
	Abbildungsverzeichnis	69
	Literaturverzeichnis	71

Auf der beiliegenden CD-ROM befinden sich

- Die entwickelte Applikation im Quellcode (MSVC 6.0-Projekt und GNU-Makefile)
- Vorkompilierte Binärdateien für Linux/x86 und MS Windows
- HTML-Quellcode-Dokumentation (erstellt mit dem Open-Source-Tool *Doxygen*)
- Exemplarische Datensätze auf Basis eines Ausgussmodells des Bronchialbaums
- Animationen von dynamischen Simulationsergebnissen als AVI-Dateien

1 Einleitung

Die vorliegende Arbeit entstand im Rahmen eines von der Deutschen Forschungsgemeinschaft geförderten Projekts zur geometrischen und funktionellen Modellierung und numerischen Simulation der Gastransport- und Austauschvorgänge in der menschlichen Lunge. Dieses Projekt wurde vom Bildverarbeitungslabor des Instituts für Anatomie und Zellbiologie des Universitätsklinikums Gießen in Kooperation mit Mevis, dem Zentrum für medizinische diagnostische Systeme und Visualisierung an der Universität Bremen durchgeführt. Die Ziele des Projekts waren:

- Die Schaffung eines geometrischen Referenzmodells der menschlichen Lunge auf Basis realistischer struktureller Daten
- Die Entwicklung eines funktionalen Modells zur Simulation des konvektiven und diffusiven Gastransports und des alveolären Gasaustauschs
- Die Visualisierung der modellierten Geometrie und der funktionalen Vorgänge

Ein integraler Bestandteil der Studie war die realistische Modellierung der geometrischen Struktur der menschlichen Lunge. Andere Arbeiten auf diesem Gebiet legen eine rein mathematische, abstrakte Darstellung der Geometrie auf Basis von statistischem Wissen zu Grunde [HT01]; im vorliegenden Projekt wurden real existierende Strukturen mithilfe von computertomografischen Techniken (HRCT) abgebildet. Dies ermöglichte die Akquisition von detaillierten dreidimensionalen Daten über den Bronchialbaum. Mit Mitteln der digitalen Bildverarbeitung wurden diese Daten vorverarbeitet und die tubulären Strukturen des Bronchialbaums extrahiert.

Im Zuge der Validierung der Daten, des Abgleichs mit Referenzmodellen in der Literatur und zur Auswertung späterer patientenindividueller Messdaten mussten umfangreiche morphologische Parameter (zum Beispiel Astlängen, Durchmesser, Verzweigungswinkel) statistisch erfasst werden. Für die numerische Simulation des Gastransports war es notwendig, die Aufteilung des Gasflusses auf die verschiedenen Bereiche der Lunge realitätsnah zu berechnen, und die komplizierte Geometrie des Bronchialbaums musste gewissen Abstraktionen unterworfen werden. Die Art der Modellierung der Baumstruktur wurde daher von den speziellen Bedürfnissen des Projekts geprägt; sie bildete die Schnittstelle zwischen den Bereichen Simulation, Statistik und Visualisierung. Das Schema in Abb. 1.1 gibt einen Überblick über den Datenfluss innerhalb des Projekts: auf Basis der CT-Rohdaten wurde der Bronchialbaum modelliert und gegebenenfalls mit zusätzlichen geometrischen Daten (etwa für die respiratorischen Einheiten, die aufgrund ihrer kleinen Dimensionen nicht mit dem CT erfasst werden können) zu einem kompletten Lungenmodell erweitert. Dieses bildet die Grundlage für die numerische Simulation. Die in dieser Arbeit entwickelte Applikation verarbeitet strukturelle Daten des Bronchialbaums sowie Ergebnisse der dynamischen Simulation zur Visualisierung und morphologischen Analyse.

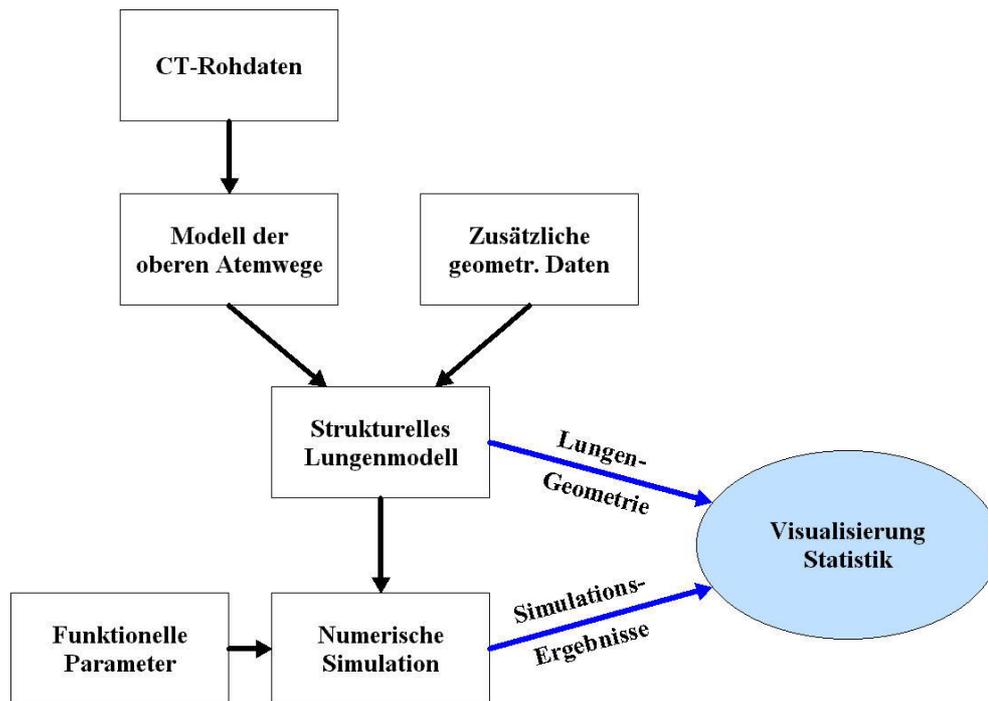


Abbildung 1.1: Schema des Datenflusses im Projekt „Computerlung“

Vor dem skizzierten Hintergrund soll eine computergrafisch-statistische Applikation entwickelt werden, die die dreidimensionale Visualisierung und Quantifizierung der Geometrie und Funktion der Baumstruktur der Lunge ermöglicht. Verschiedene Möglichkeiten zur Repräsentation von baumartigen Strukturen sollen erörtert und auf ihre Verwendbarkeit im Rahmen des Projekts hin analysiert werden. Kriterien hierfür sind

- **Detailgrad der Darstellung:**
Ein Schwerpunkt soll auf die detaillierte Modellierung der tubulären Geometrie gelegt werden. Im Bereich der oberen Atemwege, deren exakte Struktur für den Gasfluss relevant ist, ist eine detailreiche Darstellung wichtig, während feinere Verzweigungen in der Peripherie der Lunge durch einfachere Geometrien approximiert werden können.
- **Möglichkeiten zur morphologischen Analyse:**
Das Modell der Baumstruktur muss die logische Verzweigungsstruktur widerspiegeln und eine Erfassung morphologischer Parameter wie Länge, Durchmesser und Teilungsgenerationen ermöglichen.
- **Flexibilität und Kompatibilität mit vorhandenen Modulen:**
Das verwendete Simulationsmodell stellt besondere Anforderungen an die Repräsentation der Baumstruktur. Die Kompatibilität mit diesem Modell soll gewahrt werden. Das geometrische Modell und dessen Visualisierung soll verschiedene Grade der Abstraktion unterstützen, um unterschiedlichen Anforderungen des Benutzers und Hardware-Voraussetzungen gerecht zu werden.

- Hardware-Anforderungen, Speicher- und Rechenzeitaufwand:
Die Applikation soll auf verbreiteter PC-Hardware lauffähig sein und soll auch bei komplexeren Baumgeometrien eine möglichst flüssige, interaktive Benutzung ermöglichen.

Die computergrafisch-statistische Applikation soll es ermöglichen, berechnete Ergebnisse der funktionellen Simulation oder andere dynamische Daten im Zeitverlauf zu visualisieren. Darüber hinaus ist die Möglichkeit einer komfortablen statistischen Erfassung der morphologischen Parameter zu realisieren. Zu diesem Zweck muss eine sinnvolle Interaktion mit der Baumstruktur möglich sein; dazu zählen Drehungen/Verschiebung im Raum sowie die interaktive Selektion von Baumsegmenten.

2 Grundlagen der Anatomie und Bildverarbeitung

In diesem Kapitel sollen zunächst einige grundlegende fachliche Begriffe aus den Bereichen der Lungenanatomie und der bildverarbeitenden Techniken zur dreidimensionalen Erfassung des Bronchialbaums erläutert werden. Die Anforderungen, die an diese Arbeit gestellt werden, sind wesentlich von diesen Grundlagen geprägt; daher ist ein prinzipielles Verständnis für jene Gegebenheiten unabdingbar.

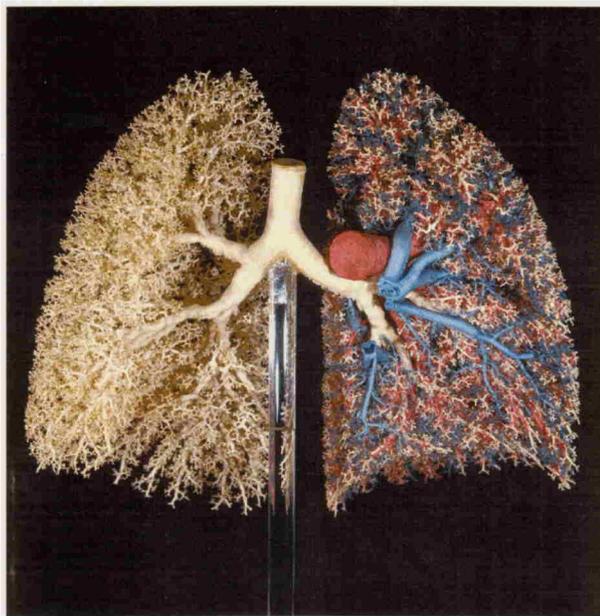
2.1 Lungenanatomie

Die menschliche Lunge (lat. *pulmo*) dient dem Austausch von Atemgasen zwischen Organismus und Umwelt. Der Mensch benötigt Sauerstoff (O_2), um in seinen Zellen Nährstoffe oxidativ zu verbrennen; bei diesem Prozess wird Kohlenstoffdioxid (CO_2) freigesetzt, welches als Zellgift aus dem Organismus transportiert werden muss. Die Lunge stellt die Schnittstelle zwischen dem Blut, in dem die Atemgase gebunden sind, und der Atmosphäre dar. Sie muss demnach einen effektiven Mechanismus bereitstellen, um den Gasaustausch zu bewerkstelligen. Zu einem umfassenden Verständnis der Atmungsprozesse sind Kenntnisse aus unterschiedlichen Bereichen notwendig. Im Rahmen dieser Arbeit ist vor allem die geometrische Struktur der luftleitenden Teile der Lunge (der Bronchialsegmente) von Bedeutung. Andere Teilgebiete, wie detaillierte atemmechanische Parameter, Diffusionseigenschaften an der Gas-Blut-Schranke oder Zelltypen in der Lunge werden nicht näher betrachtet. Der interessierte Leser sei auf die Literatur verwiesen [ST90, Wei84, Wei63].

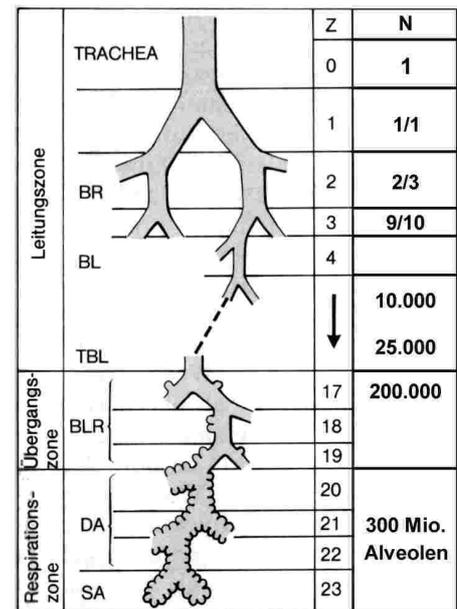
Die Lunge eines Menschen ist ein paariges Organ, das den größten Teil des Brustkorbes (*thorax*) ausfüllt. Sie teilt sich in die linke und rechte Lunge (umgangssprachlich „Lungenflügel“ genannt). Die Lungen sind durch Gewebefalten in Lappen (*lobi*) aufgeteilt, zwei in der linken und drei in der rechten Lunge. Eine weitere hierarchische Einteilung ist nach Lungensegmenten möglich; linke wie rechte Lunge können in je zehn Segmente unterteilt werden. Die Lunge erscheint als ein schwammiges, elastisches Organ; dies ist den etwa 300 Millionen Lungenbläschen (*alveoli*) zuzuschreiben, die von kleinsten Blutgefäßen, den Kapillaren, umgeben sind; sie bilden die eigentliche Schnittstelle zwischen Gas und Blutkreislauf. In dieses Gewebe eingebettet ist das Atemweg-System – der Bronchialbaum – welches durch die Luftröhre (*trachea*) mit der Außenwelt verbunden ist. Diese fein verzweigte Baumstruktur dient dem Transport der Atemluft zu den Alveolen und soll im Folgenden genauer betrachtet werden.

2.1.1 Bronchialbaum

Die Trachea stellt die Wurzel des Bronchialbaums dar. Sie beginnt am Kehlkopf (*larynx*), besitzt eine Länge von etwa 10-12 cm und einen Durchmesser von ca. 1,8 cm.



(a) Ausgusspräparat der Lunge nach Weibel, aus [HT01]



(b) Generationen des Bronchialbaums nach Weibel, aus [ST90]

Abbildung 2.1: Bronchialbaum

Am unteren (*distalen*) Ende teilt sie sich in die beiden Hauptbronchi, die in die linke und rechte Lunge verzweigen. Durch eine Vielzahl weiterer Verzweigungen (*Bifurkationen*) in jeweils zwei Tochteräste, welche immer geringere Durchmesser und Längen aufweisen, entsteht ein filigraner Baum, an dessen Enden die gasaustauschenden (*respiratorischen*) Einheiten liegen. Abbildung 2.1(a) zeigt ein Ausgusspräparat des Bronchialbaums, das von E. Weibel erstellt wurde; die feinen Verästelungen und auch die Verflechtung mit dem Adernsystem sind hier sehr gut zu erkennen.

Die Segmente des Bronchialbaums werden nach Durchmesser, strukturellen Merkmalen und Funktion gegliedert und benannt. Die oberen, dickeren Röhren nennt man *Bronchien*. Sie sind starre, durch Knorpelspangen stabilisierte Röhren. In der Folge schließen sich die *Bronchiolen* an, feinere Röhren mit zunehmend elastischen Eigenschaften. Erst am Ende des Baums findet man Alveolen, in denen tatsächlich Gasaustausch stattfinden kann; der Bronchialbaum selbst dient ausschließlich dem Transport der Atemluft, er bildet den *konduktiven* Teil der Lunge. Die Volumenänderung der Lunge, die man beim Ein- und Ausatmen (der *Inspiration* und *Expiration*) bemerkt, findet nicht im konduktiven Teil statt – dieser hat unter normalen Umständen ein konstantes Volumen von etwa 150 ml. Auf diesem relativ kleinen Volumen (nach maximaler Inspiration befinden sich insgesamt etwa fünf Liter Gas¹ in der Lunge) wird der wichtigste Effekt des Bronchialbaums erzielt: die eingeatmete Luft abzubremsen und gleichmäßig auf etwa 32.000 respiratorische Einheiten und Millionen von Alveolen zu verteilen. Durch die damit erreichte Oberflächenvergrößerung wird ein effektiver Gasaustausch durch Diffu-

¹Diese sogenannte Totalkapazität kann individuell stark schwanken, das angegebene Volumen bezieht sich auf einen statistisch ermittelten „Durchschnittsmenschen“.

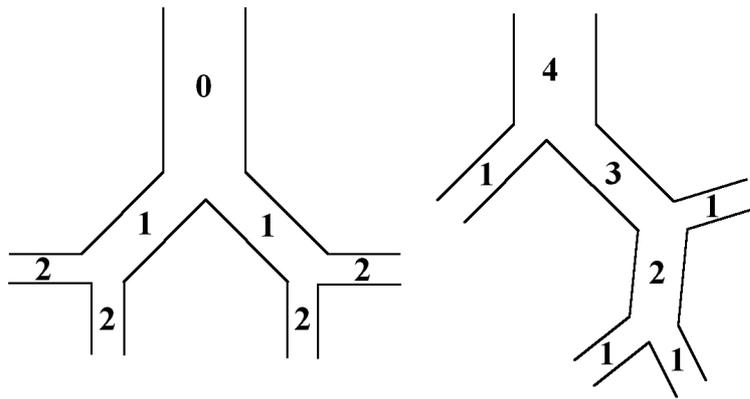


Abbildung 2.2: Vergleich von regulärer und irregulärer Verzweigungsstruktur. Links: reguläre Bifurkationen mit Weibel-Generationen. Rechts: Irreguläre Bifurkationen mit Horsfield-Ordnungen

sionsprozesse erst möglich. Abbildung 2.1(b) gibt einen Eindruck in die exponentielle Vervielfachung der Bronchialsegmente.

Klassifizierung von Ästen

Die Verzweigungsstruktur des Bronchialbaums legt eine Klassifizierung der Äste nach ihrer Tiefe im Baum nahe. Die im Folgenden als „(Weibel-)Generation“ bezeichnete Ordnungszahl gibt die Anzahl der Bifurkationen auf dem Weg von der Trachea zu einem Ast an. Die Trachea selbst habe die Generation Null, die beiden Hauptbronchi daher die Generation Eins. Geht man bei dieser Betrachtungsweise von einem vollständigen binären Bronchialbaum aus, so kann man die Anzahl der Äste auf einer Generation zu $N(g) = 2^g$ bestimmen; in jeder Generation verdoppelt sich die Anzahl der Bronchialsegmente. Das ursprüngliche Lungenmodell von Weibel [Wei63] macht unter den oben genannten Prämissen genau diese Aussage. Mithilfe dieses Modells war es möglich, beispielsweise die mittlere Generationentiefe bis zum Beginn einer respiratorischen Einheit auf Basis einer Abschätzung der Anzahl dieser Einheiten zu bestimmen. Diese Rechnung liefert die in Abbildung 2.1(b) angegebenen 23 Generationen.

Auch statistische Aussagen über Durchmesser und Längen von Bronchialsegmenten in verschiedenen Generationen basieren oft auf der vereinfachenden Annahme eines vollständigen Baums mit symmetrischen Bifurkationen, das heißt Verzweigungen mit je zwei identischen Tochterästen. Unter dieser Annahme ist eine Klassifizierung nach Weibel-Generationen in der Tat sinnvoll. Es zeigt sich jedoch, dass die Verhältnisse in der Lunge nicht so regulär sind. Bei einem irregulären Verzweigungsmuster, wie es beispielsweise in Abbildung 2.2 angedeutet ist, finden sich Äste gleichen Durchmessers auf durchaus unterschiedlichen Generationen. Das impliziert, dass die durch diese Maßzahl geschaffene Ordnung in einem realen Kontext nicht unbedingt sinnvoll sein muss. Aus diesem Grund erscheint eine andere Hierarchie angebracht, die der Klassifizierung von Flussläufen angelehnt ist [HT01, HC68]. Dies ist die sogenannte „Horsfield-Ordnung“. Hier wird mit der Zählung nicht bei der Wurzel des Baums, son-

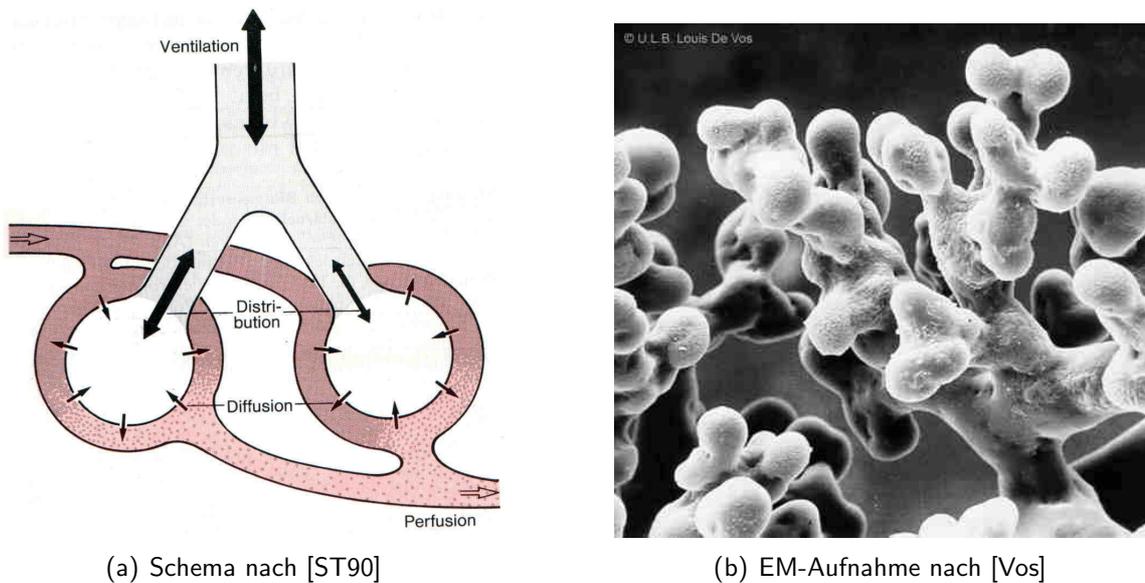


Abbildung 2.3: Alveolen

den bei seinen Endästen (Blättern) begonnen. Jedes Blatt hat die Ordnung Eins, die Ordnung eines Nicht-Blattes ist um eins höher als das Maximum der Ordnungen seiner Tochteräste. Abbildung 2.2 zeigt die verschiedenen Maßzahlen.

Statistiken über die Geometrie des Bronchialbaums sind Thema vieler Publikationen. Neben Längen, Durchmessern und Ordnungen wird eine beachtliche Anzahl weiterer Parameter betrachtet, etwa Verzweigungswinkel oder fraktale Geometrien [Man83, BEP⁺, HC67].

2.1.2 Respiratorische Zone

Etwa ab der 17. Generation gehen die konduktiven Bronchiolen in die respiratorische Zone über. In der Übergangszone, an den sogenannten *bronchioli respiratorii*, finden sich zunehmend Alveolen an den Wänden der Atemwege. Distal liegen die Alveolärsäcke, dies sind traubenförmig angeordnete Ansammlungen von Alveolen. Die einzelnen Alveolen sind polyederförmige Luftkammern mit einem Durchmesser von 0,2 bis 0,3 mm. Sie sind von Kapillaren umschlossen und nur durch eine dünne Membran (Durchmesser ca. $0,4 \mu\text{m}$) vom Blut getrennt.

In den Alveolen bestehen optimale Verhältnisse für den eigentlichen O_2/CO_2 -Austausch: durch die Verzweigungen des Bronchialbaums wurde die Geschwindigkeit des Atemgases auf wenige Millimeter pro Sekunde reduziert; dies ist im Bereich der molekularen Bewegung der Gasmoleküle [Wei84], sodass der diffusive Prozess des Gasaustauschs beginnen kann. Durch die Membran hindurch wird der Konzentrationsunterschied zwischen Alveolargas und venösem Blut ausgeglichen: es wird Sauerstoff ins Blut und Kohlendioxid in den Alveolarraum transportiert.

Es ist zu bedenken, dass das Gasgemisch (man spricht nicht von „Luft“) in den Alveolen während des Atemzyklus’ recht konstant zusammengesetzt ist. Bei der Inspiration wird der Alveolarraum keineswegs völlig durch atmosphärische Luft ersetzt, sondern vielmehr „aufgefrischt“. Dies erkennt man auch an den Volumenverhältnissen: bei normaler Atmung werden pro Atemzug lediglich 500 ml transportiert; das Restvolumen der Lunge nach der Expiration beträgt aber etwa drei Liter.

2.2 Digitalisierung geometrischer Baumobjekte

In diesem Abschnitt soll in aller Kürze auf die Methoden eingegangen werden, mit denen baumartige Strukturen in der Medizin erfasst und vorverarbeitet werden. Dies fördert das Verständnis dafür, welche weiteren Abstraktionen im Rahmen dieser Arbeit möglich bzw. notwendig sind und wo die Grenzen der biomedizinischen Abbildungstechnik liegen. Für eine ausführliche Beschreibung der Methoden sei auf [Sel99] verwiesen; die dort entwickelten Methoden für die Aufbereitung von Lebergefäßstrukturen sind jenen für den Bronchialbaum sehr ähnlich.

2.2.1 Röntgen-Computertomografie

Die Röntgen-Computertomografie, entwickelt 1972 von dem britischen Ingenieur G. Hounsfield, ist ein bildgebendes Verfahren, das es ermöglicht, die Raumstruktur anatomischer Objekte zu erfassen. Dabei wird das Organ schichtweise analysiert, indem mittels einer rotierenden Röntgenröhre Absorptionsprofile entlang verschiedener Wege durch das Objekt aufgefangen werden. Durch spezielle Rekonstruktionsalgorithmen entstehen aus diesen projizierten Absorptionsmustern diskrete Grauwert-Schichtbilder; die Abtastung einer Vielzahl benachbarter Schichten ergibt detaillierte dreidimensionale Daten, die als dreidimensionale Matrix von Volumenelementen (*Voxeln*) gespeichert werden.

Verschiedene Materialien und Gewebe weisen unterschiedliche Absorptionseigenschaften auf. Diese spiegeln sich in den Grauwerten der Schichtbilder wider und werden in *Hounsfield-Einheiten* (HE/HU) gemessen. Ein CT-Scan liefert für gewöhnlich Werte im Bereich von -1024 bis 3071 HE, also einen Bereich von 4096 (Integer-)Werten. Daher sind die entstehenden Grauwertbilder mit 12 bit/Pixel kodiert. Bestimmte Intervalle von Hounsfield-Werten deuten auf spezifische Materialien hin, beispielsweise wird Wasser mit 0 HE, Luft mit -1000 HE abgebildet. Dieses Wissen bildet die Grundlage für eine spätere Segmentierung der Daten.

Es existieren einige technische Varianten der CT. Ursprünglich wurden die Schichtbilder kreisförmig abgetastet und das abzutastende Objekt schrittweise um eine Schichtdicke weitergeschoben. Heute sind sogenannte Spiral-CTs (*helical CT*) im klinischen Alltag verbreitet, die eine kontinuierliche, eben spiralförmige Aufnahme erlauben. Eine neuere Entwicklung sind die Mehrzeilenspiral-CTs (*multislice CT*), die in einer Umdrehung mehrere Schichten in hoher Auflösung und mit höherer Geschwindigkeit erfassen können. Dies führt zu den für die Bildverarbeitung wichtigen Parametern der CT:

- Schichtdicke:
Herkömmliche Spiral-CTs arbeiten gewöhnlich mit einer Schichtdicke von 3 Millimetern; mit Multislice-CTs erfasste Bilder haben Schichtdicken von etwa einem Millimeter, und mit HRCT können Datensätze erzeugt werden, die in z-Richtung eine Auflösung von lediglich 0,4 mm besitzen².
- Räumliche Auflösung innerhalb der Schichten:
Neben der Schichtdicke spielt die zweidimensionale Auflösung in der xy-Ebene eine Rolle. Diese wird maßgeblich durch die Detektortechnik bestimmt und ist meist feiner als die z-Auflösung. Daher sind die Voxel oft nicht kubisch, sondern *anisotrop*. HRCT-Aufnahmen können innerhalb der Schichten eine Auflösung von etwa 0,35 mm haben.
- Zeitverhalten:
Bei *in vivo*-Untersuchungen von lebenden Patienten ist die Dauer der Aufnahme ein wichtiger Faktor. Beispielsweise erzeugen Atembewegungen Artefakte beim Erfassen der Lunge, daher ist es wünschenswert, eine komplette Aufnahme während des Luftanhaltens durchzuführen. Moderne CT-Scanner sind mit Aufnahmezeiten von 20-30 Sekunden dazu in der Lage.

Computertomografen können sowohl lebende Strukturen als auch anorganische Präparate erfassen. Im Rahmen des Lungenprojekts wurde das in Abb. 2.1(a) dargestellte Ausgusspräparat abgetastet, um ein hochaufgelöstes Referenzmodell zu erhalten. In diesem Präparat sind die Atemwege massive Röhren aus einer Gummisubstanz, und die nachfolgende Segmentierung wird nicht durch umgebendes Gewebe, Knochen oder Bewegungsartefakte bei der Atmung behindert. Dies ermöglicht die Erfassung auch kleinerer Bronchialäste; bei *in vivo*-Untersuchungen kann dieser Detailgrad aufgrund der vorgenannten Hindernisse nicht erreicht werden.

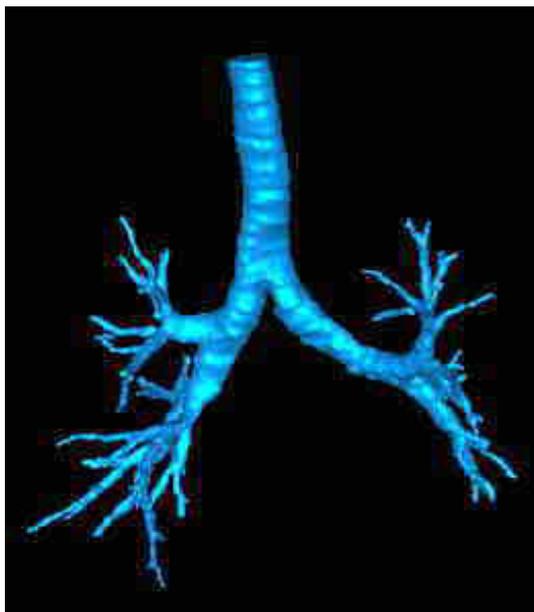
2.2.2 Extraktion von baumartigen Strukturen

Im klinischen Alltag erfolgt die Diagnostik auf Basis von CT-Untersuchungen häufig durch Begutachtung der einzelnen Schnittbilder durch den Mediziner. Relevante Strukturen und krankhafte Veränderungen können von Experten auf diese Art erkannt werden. Für die digitale Weiterverarbeitung der Rohdaten mit dem Ziel, eine realistische dreidimensionale Darstellung der untersuchten Objekte zu erhalten, ist es jedoch notwendig, zusammenhängende Strukturen wie den Bronchialbaum aus dem Datenvolumen (semi-)automatisch zu extrahieren.

2.2.2.1 Segmentierung

Strukturen wie der Bronchialbaum sind durch ihre unterschiedliche Dichte vom umgebenden Gewebe unterscheidbar. Daher gilt es zunächst, für jedes Voxel aus dem CT-

²Eine solche Auflösung wird *in vivo* bei klinischen Anwendungen nicht erreicht, da die Strahlenbelastung unverträglich hoch wäre.



(a) Segmentierter 3D-Datensatz der oberen Atemwege



(b) Mittelachsentransformation durch Skelettierung

Abbildung 2.4: Segmentierung und Skelettierung von Bronchialsegmenten, von [Mev]

Datensatz zu entscheiden, ob es Teil der Struktur ist oder nicht. Dieser Prozess wird als Segmentierung bezeichnet. Ein erster Ansatz zur Segmentierung kann sein, einen (globalen) Schwellwert zu definieren: ein Voxel wird genau dann dem Objekt zugeordnet, wenn sein Grauwert über dieser Schwelle liegt. Es wurden zahlreiche Verfeinerungen dieses Ansatzes formuliert, beispielsweise die dynamische Anpassung des Schwellenwertes aufgrund der lokalen Umgebung oder der Einsatz von Bereichswachstumsverfahren. Letzteres stellt einen iterativen Prozess dar, der ausgehend von einem vorgegebenen Punkt (Saatvoxel) benachbarte Voxel untersucht und sukzessive das Objekt segmentiert. Zusammen mit einer automatisierten Schwellenwertbestimmung können hiermit gute Ergebnisse erzielt werden (siehe [Sel99]). Da alle weiteren Transformationen unmittelbar von der Güte der Segmentierung abhängen, ist der Prozess von grundlegender Bedeutung auch für die vorliegende Arbeit.

2.2.2.2 Mittelachsentransformation

Die Segmentierung der Baumstruktur liefert eine klar abgegrenzte, volumengetreue Darstellung des relevanten Objekts. Eine detailreiche, ansprechende 3D-Visualisierung ist nun bereits möglich (siehe Abb. 2.4(a)). Der Betrachter kann die wesentlichen Merkmale klar erkennen: einzelne Äste sind zu identifizieren, die Verzweigungsstruktur ist ersichtlich, Längen und Durchmesser können abgeschätzt werden. Für eine weitergehende computergestützte Analyse des Baumes ist diese Art der Darstellung allerdings nicht optimal. Vor allem die Verzweigungsstruktur, die der menschliche Betrachter intuitiv erfassen kann, muss im Rechner auf eine formellere Art spezifiziert werden.

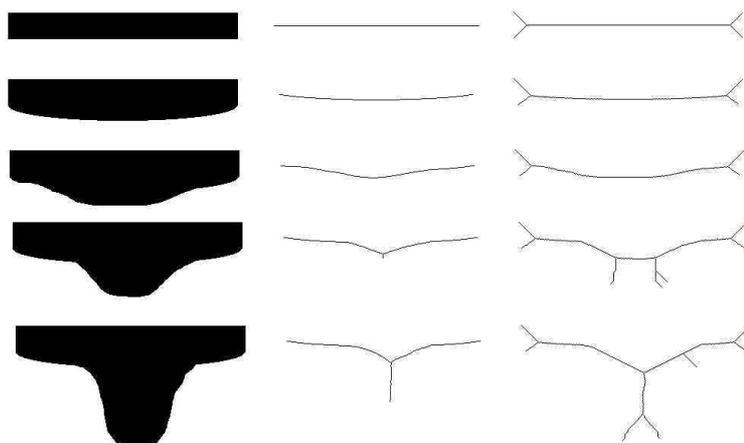


Abbildung 2.5: Vergleich von Skelettierungen einiger 2D-Rohrverzweigungen. Links: Original-Binärbild. Mitte und Rechts: Skelettierte Objekte mit zwei exemplarischen Algorithmen (siehe Fließtext)

Der Lösung besteht im Prozess der diskreten Mittelachsentransformation, auch als Skelettierung bezeichnet. Hierbei wird die „voluminöse“ Repräsentation des Baumes in eine linienartige Struktur überführt. Abbildung 2.4(b) veranschaulicht dies am Beispiel der oberen Atemwege. In dieser Darstellungsform ist es nun leicht möglich, die Verzweigungsstruktur zu formalisieren.

Die Skelettierung ist ein kritischer Schritt für die Verarbeitung des Baumes. Eine Vielzahl von Algorithmen im zwei- und dreidimensionalen Raum wurden für diesen Zweck entwickelt. Die Anforderungen an ein Skelett machen die Problematik ersichtlich:

- Das Skelett soll aus Linien bestehen, die genau einen Voxel breit sind.
- Die Skelettlinien sollen möglichst in der Mitte der Objekte liegen.
- Die Topologie der Objekte darf sich nicht verändern; zwei Objekte dürfen nicht verschmelzen, ein einzelnes nicht in Teile zerfallen.
- Das Skelett soll die Form der Originalobjekte gut wiedergeben.
- Das Skelett sollte möglichst wenige irrelevante Mittelachsen aufweisen, welche durch Unregelmäßigkeiten/Rauschen am Rand der Objekte entstehen.

Die ersten drei Forderungen sind notwendig und klar definiert, die letzten beiden Ziele nicht weniger wichtig, aber schwieriger zu erreichen. Im Falle von verzweigten Baumstrukturen ist es wünschenswert, die Verzweigungsstruktur im Skelett wiederzufinden, die man auch intuitiv als Betrachter erkennt. Konkret stellt sich beispielsweise die Frage, ab wann eine Ausbuchtung in einem dicken Ast als Verzweigung identifiziert wird. Exemplarisch werden in Abb. 2.5 einige manuell erzeugte zweidimensionale Objekte gezeigt, welche verschiedene Grade von Ausbuchtungen zeigen. Auf diese Daten wurden

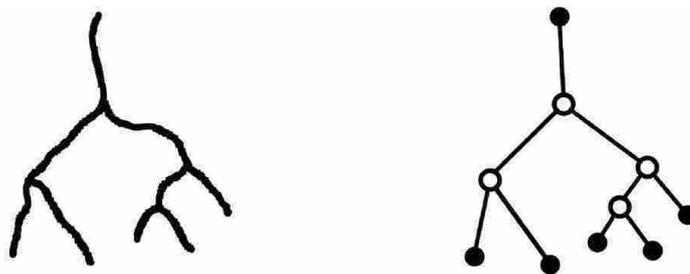


Abbildung 2.6: Transformation eines skelettierten Baums in eine Graphenstruktur

vom Autor zwei Varianten eines Skelettierungs-Algorithmus angewandt, der sukzessive Pixel vom Rand der Objekte abträgt. Auf die Funktionsweise der Verfahren soll hier nicht weiter eingegangen werden; der Vergleich zeigt allerdings typische Merkmale der Skelettierungsproblematik: Algorithmus 1 (Mitte) erzeugt ab dem vierten Teilobjekt eine einzelne Verzweigung sowie einzelne Linienenden. Algorithmus 2 (rechts) liefert beim vierten Teilobjekt (unerwünscht) drei Bifurkationen, die man nicht sinnvoll in Zusammenhang mit der Originalstruktur bringen kann. Außerdem findet sich an den Enden der Äste weitere Verzweigungen. Obwohl dies durchaus im Einklang mit der mathematischen Definition der Mittelachse steht, ist es dennoch für die Extraktion von Baumstrukturen nicht praktikabel. Die Methode zur Skelettierung, die im Rahmen des Lungenprojekts eingesetzt wird, ist optimiert auf diesen Einsatzzweck und kann durch Parameter gegebenenfalls angepasst werden. Dennoch ist es wichtig, mögliche Probleme im Bewusstsein zu behalten, da sich die hier extrahierte Struktur unmittelbar in den Daten, die im weiteren Verlauf für diese Arbeit verwendet werden, wiederfindet.

Die skelettierte Darstellung des Baumes ist bereits stark abstrahiert. Informationen über den Durchmesser von Ästen sind verloren, wenn man lediglich die Skelettlinie als binäres Bild zur Verfügung hätte. Speichert man allerdings zu jedem Skelettvoxel die minimale Entfernung zum Rand des Originalobjekts, so kann die Form des Ursprungsbildes näherungsweise rekonstruiert werden. Auch die Gesamtanzahl der Voxel, die ein (Teil-)Objekt einnimmt, also dessen Volumen, ist eine wichtige Größe, die nicht verloren gehen darf.

2.2.2.3 Weitere Abstraktionen

Die skelettierte Baumstruktur bietet ideale Voraussetzungen für einen weiteren Transformationsschritt: die Überführung in einen gerichteten Graphen, dessen Knoten End- und Verzweigungspunkte des Baumes repräsentieren. Kanten zwischen den Knoten repräsentieren die Äste. Ein Blick auf das Beispiel in Abb. 2.6 zeigt: ohne weitere Zusatzinformationen wird der Baum nunmehr auf eine Struktur mit geraden Kanten reduziert; der größtmögliche Abstraktionsgrad ist erreicht.

Der Umgang mit Graphen ist ein gut erforschtes Gebiet in der Informatik. Somit eignet sich diese Art der Darstellung gut für weitere Analysen der Verzweigungsstruktur.

Um neben den räumlichen Koordinaten der Knoten und den Verbindungsinformationen weitere Daten wie Astdurchmesser entlang der Äste zu erhalten, kann man die Kanten mit weiteren Attributen ausstatten. In der Tat ist das hier skizzierte Format eines gerichteten Graphen genau die Struktur, welche in dieser Arbeit als Datenlieferant für die Visualisierung und Analyse dient. Auch für nachgeschaltete Aufgaben, namentlich die numerische Simulation, ist dieses Format optimal: es basiert auf einem einfachen Grundmuster und enthält darüber hinaus genügend Zusatzinformationen, um eine detailliertere Darstellung des Baumes zu ermöglichen. In Abschnitt 4.1.2 wird das Dateiformat, das im Zuge der in dieser Sektion beschriebenen Transformationskette entsteht, genauer beschrieben.

3 Konzeption des geometrischen Modells

Nachdem nunmehr die Grundlagen über das fachliche Umfeld und die im Vorfeld verwendeten Techniken zur Digitalisierung der Baumgeometrien erörtert worden sind, ist es möglich, die Anforderungen an das zu entwickelnde Visualisierungsmodell konkreter zu analysieren. Zunächst werden daher mögliche Repräsentationen der Baumstruktur vorgestellt und ihre Relevanz für die Arbeit untersucht. Im Mittelpunkt stehen Fragestellungen wie Flexibilität des Modells, Detailgrad der Darstellung, vor allem aber auch Möglichkeiten zur statistischen Auswertung und Eignung der Datenstrukturen für die numerische Simulation. Die praktischen Erfordernisse der letzten beiden Punkte beeinflussen in hohem Maße die Wahl des Modells.

Auf Basis der gewählten Visualisierungsmethode wird anschließend das Konzept im Detail entwickelt. Das schließt eine genaue Betrachtung der zur Verfügung stehenden Eingangsdaten ein, die das Ergebnis einer Transformationskette sind, wie sie im Grundlagenteil beschrieben wurde. Nachdem die Repräsentation der Baumstruktur klar definiert wurde, folgen Überlegungen dazu, wie sich dynamische Daten, das heißt sich im Zeitverlauf ändernde, den Ästen zugeordnete Informationen, an dem Modell visualisieren lassen. Auch ein Konzept zur komfortablen Erfassung und Weiterverarbeitung von statistischen Daten zur Baumstruktur wird entwickelt.

3.1 Repräsentationsvarianten von Baumstrukturen

Eine kurze Übersicht über mögliche Darstellungen dreidimensionaler Strukturen soll dabei helfen, die Vor- und Nachteile dieser Varianten zu beurteilen. Dabei geht es nicht ausschließlich um die Güte der Visualisierung, sondern auch um die Frage, welche (abstrahierten) Informationen sich aus den zu Grunde liegenden Datenstrukturen ermitteln lassen.

3.1.1 Volumen-Rendering

Die direkte volumenbasierte Visualisierung von computertomografischen Daten nutzt unmittelbar die dreidimensionale Voxeldarstellung, die durch die bildgebenden Techniken erzeugt wird. Es existieren verschiedene Techniken, mithilfe derer eine zweidimensionale Projektion der Daten erzeugt werden kann, die sehr detailreich ist. Beim Raycasting und Raytracing werden virtuelle Strahlen durch die Szene geschickt, ihr Kontakt mit Datenvoxeln bestimmt das Aussehen der Visualisierung. Voxeln können dazu nach verschiedenen Kriterien Transparenzwerte zugeordnet werden, um es möglich zu machen, gleichsam in einen Körper hineinzusehen. Andere Ansätze berechnen zur Darstellung dreidimensionale Texturen aus den Daten.

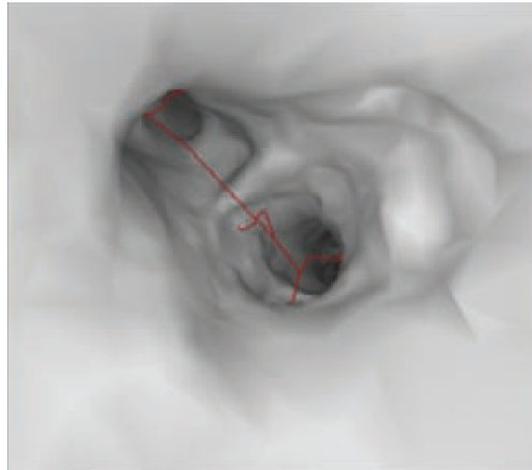


Abbildung 3.1: Virtuelle Bronchial-Endoskopie, aus [KHM⁺02]

Die zu verarbeitenden Volumendaten sind sehr umfangreich. Eine 3D-Voxelmatrix der Größe 512^3 würde bei 12 Bit pro Voxel 192 MByte unkomprimiertes Datenvolumen belegen. Auch die Algorithmen zur Visualisierung sind inhärent sehr rechenintensiv. Das Feld der Volumenvisualisierung ist allerdings ein ergiebiges Forschungsterrain, und es gibt eine Reihe von Optimierungen in Bezug auf unterschiedliche Anwendungsbereiche (Echtzeit-Verarbeitung im klinischen Alltag, statische Detailbilder) und Hardware, die den Erfordernissen optimal gerecht wird [JDRH95, SWHB95], sowie spezialisierte Software-Bibliotheken.

Volumen-Rendering kann optisch sehr ansprechende Resultate zeigen. Für den Mediziner bietet sich die Möglichkeit, Objekte mit einem Höchstmaß an Informationen zu begutachten: bei der virtuellen Endoskopie kann das Innere von tubulären Strukturen, wie dem Bronchialbaum [KHM⁺02] oder dem Darm [YHJ⁺], auf krankhafte Veränderungen hin untersucht werden (siehe Abb. 3.1). Der Nachteil dieser „ungefilterten“ Daten ist jedoch die mangelnde Eignung für eine direkte automatisierte Erfassung von statistisch auswertbaren Informationen, wie zum Beispiel Verzweigungsstruktur, mittlere Astlängen etc.. Hierzu sind nachgeschaltete Abstraktionen notwendig, wie sie etwa im Grundlagenteil angerissen werden. Oftmals werden daher hybride Ansätze verwirklicht, die den Detailreichtum der Originaldaten mit abstrahiertem Wissen verbinden.

Da die vorliegende Arbeit einen Schwerpunkt auf die statistische Erfassung morphometrischer Merkmale, auf eine schnelle Darstellung auf nicht-spezialisierter Hardware und auf die Möglichkeit der Verwendung der Datenstrukturen für die numerische Simulation kompletter Bäume legt, wird eine direkte Volumenvisualisierung hier nicht angestrebt.

3.1.2 Oberflächen-Rendering

In der Computergrafik ist es üblich, die Oberflächenstruktur komplexer dreidimensionaler Objekte mit einem Netz von Dreiecken zu visualisieren. Man operiert dabei mit

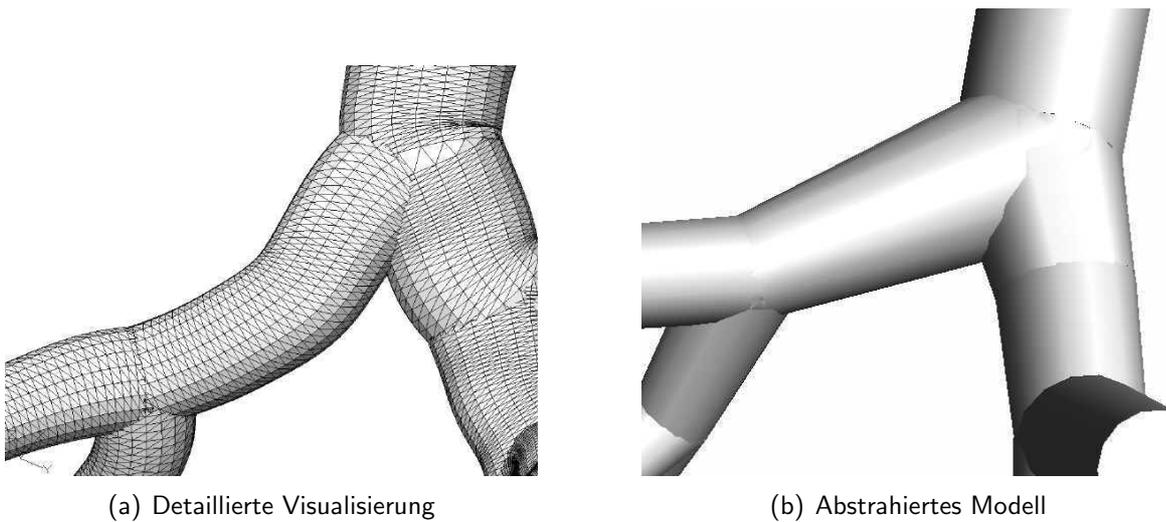


Abbildung 3.2: Oberflächendarstellungen einer Bronchialbaum-Bifurkation

einfachen grafischen Primitiven der euklidischen Geometrie, die durch Punkte (Vertices) und Verbindungslinien zwischen ihnen charakterisiert werden. Es gibt verschiedene Möglichkeiten, eine Baumstruktur, die ursprünglich als Voxeldatensatz vorliegt, in eine oberflächenbasierte Darstellung zu überführen. Die detaillierteste Variante ist in Abb. 3.2(a) zu sehen. Sie zeigt eine Bronchialbaum-Bifurkation auf Basis der HRCT-Daten des Ausgussmodells von Abb. 2.1(a). Mittels spezieller Verfahren (z.B. dem Marching-Cubes-Algorithmus) wird die segmentierte Voxelmenge, die den Bronchialbaum darstellt, in ein Dreiecksnetz überführt, das die Oberfläche wiedergibt. Der Informationsverlust bei dieser Approximation ist einstellbar und richtet sich nach der Feinheit der Dreiecke, und es ergibt sich eine detailgetreue Darstellung der tubulären Struktur. Vor allem Veränderungen im Durchmesser der Teilsegmente sind fast ohne Einschränkungen erkennbar, und die genaue Geometrie der Verzweigungsstellen bleibt erhalten.

Man erkennt an Abbildung 3.2(a), dass es möglich ist, das Dreiecksnetz schattiert und beleuchtet darzustellen. Daraus resultiert eine sehr ansprechende Visualisierung; verschiedene Beleuchtungs- und Schattierungsmodelle, die Belegung der Oberflächen mit zweidimensionalen Bildern (Texturen) und schnelle Transformationen zur Bewegung der Objekte im Raum sind verfügbar. Moderne PC-Hardware verfügt über leistungsfähige Grafikadapter, die eine detaillierte Echtzeitdarstellung solcher Geometrien möglich machen, und Softwarebibliotheken wie OpenGL [WND97] oder MS Direct3D bieten ausgereifte Schnittstellen zur Erstellung von oberflächenbasierten Darstellungen. Das macht das Verfahren grundsätzlich für die vorliegende Arbeit interessant; es bleibt die Frage nach dem Detailgrad der Darstellung und den zugrundeliegenden Daten. Die Raumkoordinaten aller Dreiecke der Darstellung von Abb. 3.2(a) bieten keinen direkten Zugang zu abstrakteren Informationen zur automatisierten Vermessung der Röhren. Hat man allerdings eine abstraktere Beschreibung der Baumstruktur als Basis, so sind solche Erfassungen nicht nur viel einfacher möglich, auch eine effiziente Visualisierung kann auf diesen vereinfachten Daten aufbauen. Abb. 3.2(b) zeigt ein sol-

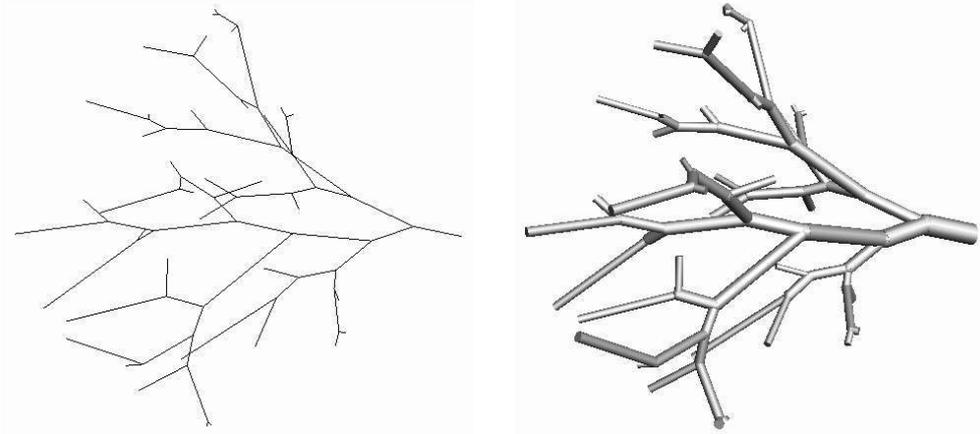


Abbildung 3.3: Vergleich linienhafter und voluminöser Darstellung

ches Oberflächenmodell, dessen Entstehung später genau betrachtet werden soll. Die oberflächenbasierte Darstellung bietet demnach alle Möglichkeiten, die für das Projekt benötigt werden. In der Hauptsache wird es bei der Konzeption des konkreten Modells nun darum gehen, die verfügbaren Daten in einer ansprechenden und flexiblen Form als Oberflächen zu visualisieren.

3.1.3 Linienhafte Darstellung

Die Verzweigungsstruktur von Baumstrukturen ist auch in sehr abstrahierten Darstellungsvarianten noch erkennbar. Eine linienhafte Darstellung ist am wenigsten rechenintensiv, aber oftmals zu rudimentär, um dem Benutzer wertvolle und räumlich gut erkennbare Informationen liefern zu können. Daten über Astdurchmesser und genaues Aussehen von Bifurkationen sind verloren. Die Skelettlinien, die bei der Mittelachsentransformation der segmentierten Voxeldaten des Bronchialbaums entstanden sind, sind ein Beispiel einer linienhaften Darstellung (Abb. 2.4(b)). Den größten Grad an Abstraktion erreicht man, wenn man die krummen Skelettlinien durch gerade Strecken ersetzt.

Abb. 3.3 zeigt am Beispiel eines Bronchial-Unterbaums den Unterschied zwischen einer voluminösen, beleuchteten Darstellung der Baumstruktur (hier mit zylindrischen Objekten) und einer reinen Linienrepräsentation. Obwohl auch das Linienmodell perspektivisch projiziert wurde, ist die Tiefeninformation verlorengegangen, da dem Betrachter Anhaltspunkte zur räumlichen Lage einzelner Äste fehlen: an Kreuzungspunkten zweier Äste erkennt man lediglich zwei schneidende Geraden, nicht die aussagekräftige Überdeckung eines flächenhaften (und durch Beleuchtung hervorgehobenen) Objekts. In Printmedien, zum Beispiel bei der Darstellung von Proteinstrukturen, findet man manchmal solche Linien-Darstellungen; die dreidimensionale Information kann dann dadurch verstärkt werden, dass man zwei versetzte Darstellungen des Objekts nebeneinander abbildet, die den Bildern entsprechen, welche die beiden Augen bei einer dreidimensionalen Betrachtung auffangen würden. Durch „gezieltes Schielen“ oder ei-

ne spezielle Sehhilfe (Lorgnette) kann ein dreidimensionaler Eindruck entstehen, und die Raumstruktur ist deutlich besser sichtbar. Solche Darstellungen finden sich beispielsweise in [Les03]. Am Computer sind komfortablere 3D-Darstellungen verfügbar, beispielsweise mithilfe von sogenannten „Shutter-Brillen“. Diese Art der Übermittlung von Tiefeninformation ist nicht auf Linienstrukturen beschränkt und kann als interessanter Zusatzeffekt betrachtet werden. Für diese Arbeit hat sie aber keine Relevanz. Eine linienhafte Darstellung kann somit sicher nicht die bevorzugte Darstellungsart des hier entwickelten Modells sein; als alternativer Modus ist sie vorstellbar, da eine Realisierung keinen übermäßigen Mehraufwand bedeutet.

3.2 Das Röhrenmodell

Im Folgenden wird das Daten- und Visualisierungsmodell der Baumstruktur definiert. Grundlage für das Modell ist eine Datenstruktur, die eine Abstraktion der originären Voxeldaten darstellt, aber noch genügend Informationen für eine detailreiche und gut handhabbare Darstellung enthält. Diese Struktur wird im Folgenden exakt untersucht, und danach ein oberflächenbasierendes Darstellungsverfahren entwickelt, welches den Bronchialbaum mittels zylindrischer oder konischer Primitivobjekte darstellt.

3.2.1 Eingangsdaten

Die Datenstruktur, welche die Basis für das geometrische Modell bildet, beschreibt den Bronchialbaum als einen gerichteten Graphen. Sie wird durch die in Abschnitt 2.2 beschriebene Transformationskette vom Kooperationspartner Mevis in Bremen erzeugt; daher wird abkürzend auch vom „Mevis-Format“, „Mevis-Dateien“ oder „Mevis-Graph“ gesprochen. Der Vorteil der zu Verfügung stehenden Daten ist, dass sie genügend abstrahierte Verzweigungsinformationen enthalten, um die Baumstruktur auf komfortable Weise zu extrahieren, andererseits auch weitere Zusatzinformationen bereitzustellen, um detailliertere geometrische Feinheiten darstellen zu können. Welche Informationen bereitgestellt werden und welche Transformationen im Rahmen dieser Arbeit notwendig sind, soll im Folgenden beschrieben werden. Zunächst sollen, in direktem Bezug zum Mevis-Format, einige Begriffe aus der Graphentheorie eingeführt werden. Weiterführendes zu Graphen findet sich zum Beispiel in [CH94].

3.2.1.1 Gerichtete Graphen

Ein gerichteter Graph ist ein Paar $G = (N, E)$ aus einer nicht-leeren endlichen Menge N von *Knoten* und einer Menge $E \subseteq (N \times N)$ von *Kanten*, die gerichtete Verbindungen zwischen den Knoten darstellen. Im Mevis-Format entsprechen die Knoten Punkten im dreidimensionalen Raum und repräsentieren End- und Verzweigungspunkte von Ästen. Letztere werden durch die Kanten repräsentiert. Eine Kante $k = (a, b) \in E$ besitzt den *Startknoten* a und den *Endknoten* b . Kanten mit dem Anfangsknoten a heißen *Nachfolgerkanten* von a , Kanten mit Endknoten b heißen *Vorgängerkanten* von b . Ein

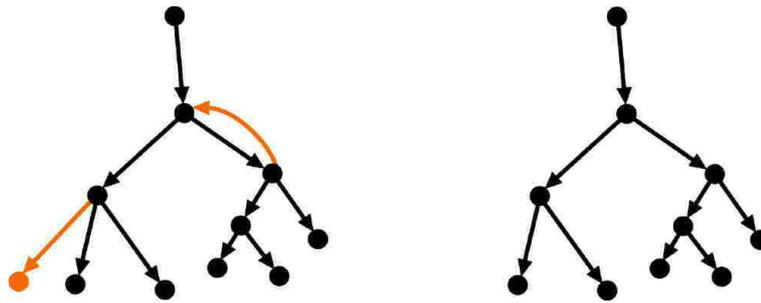


Abbildung 3.4: Beispiele für Mevis-Graphen. Links: ungültiger Graph mit Trifurkation und Zyklus. Rechts: korrekter binärer Baum

Knoten heißt

- *Wurzelknoten* genau dann, wenn er keine Vorgängerkante besitzt
- *Terminalknoten* gdw. er keine Nachfolgekante besitzt
- *Linearer Knoten* gdw. er genau eine Vorgängerkante und eine Nachfolgekante besitzt
- *Bifurkationsknoten* gdw. er genau zwei Nachfolgekanten besitzt
- *Multifurkationsknoten* gdw. er mehr als zwei Nachfolgekanten besitzt
- *kritischer Knoten* gdw. er mehr als eine Vorgängerkante besitzt

In einem *gewichteten* Graphen sind die Kanten mit einem oder mehreren Attributen (Gewichten) belegt. Im Mevis-Graphen besitzen auch die Knoten Attribute, wie weiter unten beschrieben. Ansonsten enthält eine Mevis-Datei genau die oben beschriebenen strukturellen Möglichkeiten. Die konkrete Syntax des Dateiformats wird hier zunächst vernachlässigt; Kapitel 4 führt dies genauer aus.

3.2.1.2 Geforderte Baumstruktur

Man erkennt, dass die oben formulierte Graphenstruktur sehr mächtig ist. Beliebige gerichtete Graphen können Zyklen, mehrere oder keinen Wurzelknoten enthalten oder disjunkte Teilgraphen bilden. Die Definition eines *Baumes* ist restriktiver und für den hier entwickelten Anwendungsbereich angemessen. In der Tat soll gefordert werden, dass der Graph einen *binären* Baum darstellt, also eine Struktur, die nur Bifurkationen oder lineare Knoten aufweist. Verfügbare Daten des Bronchialbaumes zeigen, dass diese Einschränkung sinnvoll ist. Trifurkationen oder mehrfache Verzweigungen treten in der Biologie nicht auf, nötigenfalls könnte eine Dreiteilung in Form von zwei nahegelegenen Bifurkationen modelliert werden. Die Vorteile dieser Einschränkungen überwiegen in jedem Fall mögliche Nachteile: die interne Struktur des Baumes ist, wie im Implementierungskapitel 4 beschrieben, weitaus effizienter zu handhaben, statistische Daten sind

einfacher zu erheben und auch eine nachgeschaltete Simulation ist leichter möglich. Ein Mevis-Graph muss demnach folgende Eigenschaften besitzen, um für die vorliegende Arbeit genutzt zu werden:

- Er muss genau einen Wurzelknoten besitzen.
- Er darf keine Multifurkationsknoten aufweisen.
- Er darf keine kritischen Knoten aufweisen.

Diese Bedingungen reichen aus, um einen Baum zu erhalten, welcher ausgehend vom Wurzelknoten durchlaufen werden kann. Zyklen werden durch die erste und dritte Bedingung ausgeschlossen. Etwaige disjunkte (bedeutungslose) Teilgraphen, die nicht mit dem Wurzelknoten verbunden sind, können so zwar nicht erkannt werden, allerdings ist ein solcher Test in der Praxis nicht von Bedeutung. Enthält ein Mevis-Graph Triefurkationen oder Zyklen, so können sie bei Bedarf eliminiert werden (durch Entfernen entsprechender Kanten oder Teilgraphen).

3.2.1.3 Attribute von Knoten und Kanten im Mevis-Graphen

Bei der Beschreibung der Verzweigungsstruktur des Mevis-Graphen konnte außer Acht gelassen werden, dass tatsächlich voluminöse dreidimensionale Bäume repräsentiert werden. Diese geometrischen Daten liegen in Form von Attributen der Knoten und Kanten vor, die nun erörtert werden sollen. Wie diese Attribute genutzt werden, bestimmt maßgeblich den Detailgrad der Visualisierung.

Die Knoten des Graphen bilden die Anfangs- und Endpunkte der Äste. Dementsprechend muss jeder Knoten einen Satz dreidimensionaler Koordinaten enthalten, der seine Position bestimmt. Beinahe alle weiteren wichtigen Attribute finden sich in den Kanten; einzig das versorgte Volumen wurde den Terminalknoten zugeordnet. Dieses Maß ist spezifisch für den Bronchialbaum und gibt an, welchen Teil der gesamten Lunge ein Terminalast versorgt. Dieser Parameter ist für die Baumgeometrie nicht relevant, aber für die Statistik von Bedeutung; eine genaue Beschreibung findet sich daher in Abschnitt 3.4.

Die Kanten enthalten Informationen über die Form des Astes, den sie repräsentieren. Der wichtigste Parameter ist das Gesamtvolumen des Astes; dieses wurde beim Erstellen der Struktur bestimmt durch die Anzahl der Objektvoxel, die der Skelettlinie des Astes zugeordnet worden sind. Es gibt jedoch auch weitere Daten, die zu einer detaillierteren Beschreibung der Geometrie herangezogen werden können. Der ursprüngliche, meist leicht krummlinige Verlauf der Skelettlinie, ist im Mevis-Format nicht verloren gegangen. In der Tat sind die Koordinaten aller Voxel der Skelettlinie gespeichert, begonnen mit den beiden End- oder Verzweigungsvoxeln, die die Koordinaten des Anfangsknotens \vec{P} und des Endknotens \vec{Q} bestimmen:

$$v_0 = \vec{P}, v_1, v_2, \dots, v_n = \vec{Q}$$

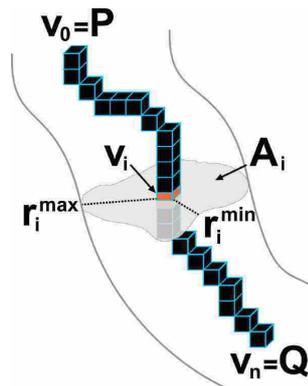


Abbildung 3.5: Erweiterte geometrische Informationen im Mevis-Graphen

Mit diesen Informationen kann der genaue Verlauf der Achse des Astes rekonstruiert werden. Zusätzlich sind zu jedem Skelettvoxel v_i einige andere Daten aufgeführt, die in Abb. 3.5 illustriert sind:

- Astquerschnittsfläche A_i :
Die Fläche (in mm^2) eines Schnitts durch den Ast senkrecht zur Mittelachse.
- Inkreisradius r_i^{\min} :
Der minimale Radius vom Skelettvoxel zur Grenze des Astquerschnitts.
- Umkreisradius r_i^{\max} :
Der maximale Radius vom Skelettvoxel zur Grenze des Astquerschnitts.

In Abb. 3.5 ist zu beachten, dass der angedeutete Astquerschnitt natürlich ebenfalls aus Voxeldaten besteht und die schematische Darstellung nur dem Verständnis wegen gewählt wurde. Die Querschnittsfläche berechnet sich aus der Voxel-„Scheibe“, die um das Mittelachsenvoxel liegt und ist daher nur ein Näherungswert. Vor allem im Bereich von Bifurkationen, wo drei Äste miteinander verschmelzen, ist es schwierig, ein korrektes Maß für den Astquerschnitt zu bestimmen. In- und Umkreisradius sind bei annähernd kreisrunden Ästen fast gleich, unterliegen aber ähnlichen Toleranzen wie der Astquerschnitt, da sie auf der gleichen Voxelmenge basieren: prinzipiell sind es die minimalen bzw. maximalen Entfernungen vom Mittelachsenvoxel zu den Voxeln der Querschnittsscheibe. In der Praxis werden diese Maßzahlen noch durch Glättungs- und Mittelungsoperationen stabilisiert. Details zu diesen Verfahren finden sich in [Sel99].

Im Folgenden sollen nun zwei Konzepte entwickelt werden: ein Basismodell, das komplette Kanten in Form je eines Objekts darstellt, und darauf aufbauend eine Geometrie, welche die erweiterten Voxelinformationen verwertet.

3.2.2 Das Basismodell

Im Basismodell der Baumgeometrie soll die tubuläre Struktur zwischen je zwei Bifurkationspunkten durch ein einzelnes geometrisches Objekt repräsentiert werden. Das

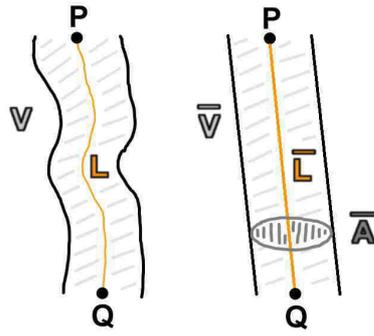


Abbildung 3.6: Vergleich krummliniger und zylindrischer Äste

bedeutet, dass jede Kante im Mevis-Graphen durch ein zylindrisches oder konisches Grundelement dargestellt wird. Die erweiterten geometrischen Informationen zu Durchmessern entlang der Skelettlinie werden für die Darstellung nicht verwertet.

Zylindrische Äste

Von einem beliebigen Ast sind durch die Angaben im Mevis-Graphen folgende Größen bekannt (vgl. Abb. 3.6):

- Das Gesamtvolumen des Astes: V .
- Die tatsächliche (krummlinige) Länge des Astes: L .
- Die Koordinaten des Anfangs- und Endpunkts des Astes: \vec{P} und \vec{Q} .

Will man diesen Ast durch einen Zylinder approximieren, so müssen folgende Daten bestimmt werden:

- Die Querschnittsfläche des Zylinders: \bar{A} .
- Die Koordinaten des Anfangs- und Endpunkts des Zylinders: \vec{P} und \vec{Q} .

Die Koordinaten der beiden Punkte sind notwendigerweise die selben wie die des tatsächlichen Astes. Die Länge \bar{L} des Zylinders wird dann eindeutig bestimmt durch den direkten Abstand zwischen seinen Endpunkten: $\bar{L} = |\vec{Q} - \vec{P}|$. Diese Länge wird in der Regel kleiner sein als die tatsächliche Länge des Astes, der einen krummlinigen Verlauf hat: $\bar{L} \leq L$. Diese Verkürzung lässt sich nicht vermeiden und ist integraler Bestandteil der Abstraktion. Es bleibt nun die Frage nach dem Querschnitt des Zylinders. Sinnvollerweise sollte dieser dem mittleren Querschnitt des krummen Astes entsprechen. Dieser berechnet sich zu V/L , also aus dem Volumen und der krummlinigen Länge des Astes. Setzt man also $\bar{A} = V/L$, so ergibt sich das Volumen des Zylinders zu

$$\bar{V} = \bar{A} \cdot \bar{L} = V \cdot \bar{L}/L \leq V \text{ wegen } \bar{L}/L \leq 1$$

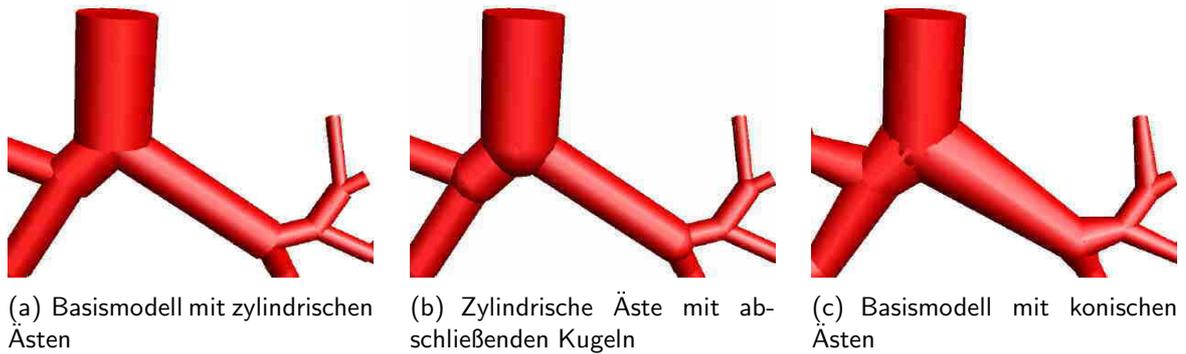


Abbildung 3.7: Mögliche Darstellungen der Baumstruktur im Basismodell

Das Volumen des Zylinders ist nun also kleiner als das reale Gesamtvolumen. Stellt man die Bedingung auf, dass die Volumina von Ast und Zylinder exakt gleich sein sollen, so müsste sich der Querschnitt auf Basis dieses Volumens und der *direkten* Länge zu $\bar{A} = V/\bar{L}$ berechnen. Dann ist die Volumengleichheit gewahrt, der Querschnitt jedoch größer als der tatsächliche mittlere Querschnitt des realen Astes. Es ist einleuchtend, dass man nicht beide Bedingungen erfüllen kann, wenn man das krumme Segment in einen verkürzten Zylinder transformiert. Die exakte Repräsentation des mittleren Querschnitts scheint allerdings die bessere Möglichkeit zu sein; daher soll diese Formel verwendet werden. Das Zylindermodell verkürzt also die Längen der Segmente und verkleinert deren Volumen entsprechend.

Man kann nun jeden Ast, also jede Kante im Mevis-Graphen, als Zylinder visualisieren. Hierbei zeigt sich (Abb. 3.7(a)), dass die Enden der Zylinder an den Bifurkationen unansehnlich scharf begrenzt sind. Daher werden abschließende Kugeln an die Zylinder gesetzt, um den optischen Eindruck zu verbessern. Diese Kugeln tragen nichts zum Volumen der Äste bei und haben daher streng genommen nichts mit der modellierten Geometrie zu tun; sie sind nur für die Visualisierung relevant.

Konische Äste

Beim Einfügen der abschließenden Kugeln in der zylindrischen Darstellung ist klar geworden, dass man grundsätzlich unterscheiden muss zwischen den verfügbaren Informationen und der gewählten Visualisierung: ein zylindrischer Ast kann nur seine (direkte) Länge, seine Querschnittsfläche und damit sein Volumen zur Verfügung stellen; die runden Elemente der Visualisierung sind durch diese Parameter im Grunde nicht zu rechtfertigen. Diese Problematik setzt sich fort, wenn man eine weitere alternative Darstellungsform wünscht: die Visualisierung mit konischen Elementen, die sich nach „unten“ hin verjüngen. Im Bronchialbaum haben Äste auf einer tieferen Generation gemeinhin geringere Durchmesser, daher könnte eine solche Darstellung eher den tatsächlichen Verhältnissen entsprechen. Auch optisch beurteilen Betrachter oftmals ein Bild wie in Abb. 3.7(c) positiver als die zylindrische Variante. Der Sinn der konischen Äste ist es, die Querschnitte von benachbarten Ästen aneinander anzupassen; das Aussehen eines Astes wird demnach von seinen Nachbarn mitbestimmt. Eine solche

Anpassung ist nicht volumenerhaltend, sie erhält ihre Berechtigung lediglich durch die Vorkenntnis über die allgemeine Struktur des Bronchialbaums.

Für die Modellierung konischer Segmente werden nunmehr jedem Ast folgende Attribute zugeordnet:

- Die proximale Querschnittsfläche : A^{prox} .
- Die distale Querschnittsfläche : A^{dist} .
- Die Koordinaten des Anfangs- und Endpunkts des Zylinders: \vec{P} und \vec{Q} .

Im Basismodell kann man definieren, dass $A_i^{prox} = \bar{A}_{i-1}$ und $A_i^{dist} = \bar{A}_i$, wie in Abb. 3.7(c) zu sehen ist. Das heißt, ein Ast beginnt mit dem Querschnitt des Elternastes und verzüngt sich am Ende zu seinem tatsächlichen Durchmesser. Mercer et al. [MAMC91] schlagen in ihrem Simulationsmodell vor, dass $A_i^{prox} = \bar{A}_i$ und $A_i^{dist} = \bar{A}_{s1} + \bar{A}_{s2}$, wobei $s1, s2$ die beiden Tochteräste von i sind. Diese Definition hat ihren Hintergrund im zugrunde liegenden Berechnungsmodell, ist aber für die Visualisierung wenig sinnvoll, da Äste sich nach unten hin verdicken würden, weil die Summe der Tochterquerschnitte im Allgemeinen größer ist als der des Elternastes. Beide Alternativen bringen Darstellungen ein, die nicht auf gemessenen Daten basieren, und verändern das Volumen der Äste; daher sollen im Basismodell generell zylindrische Äste (mit $A^{prox} = A^{dist}$) verwendet werden. Im erweiterten Modell wird diese Einschränkung aufgehoben.

3.2.3 Erweiterung der Geometrie

Die Repräsentation eines kompletten Astes durch ein einziges Objekt ist für relativ kleine Äste (in tieferen Verzweigungsordnungen) durchaus angemessen. Für die oberen Verzweigungen, in denen verhältnismäßig große Längen und Durchmesser auftreten, ist es jedoch wünschenswert, die erweiterten geometrischen Daten, die in Form der Mittelachse und den ermittelten Durchmesserinformationen vorliegen, zu verarbeiten und ein detaillierteres Modell zu generieren. Dies verbessert einerseits den optischen Eindruck, bietet aber auch für die numerische Simulation Vorteile. So kann ein krankhaft veränderter, langer Bronchus, der an einer scharf lokalisierten Stelle eine Verengung (Stenose) aufweist, im Basismodell nur unzureichend modelliert werden: da nur der mittlere Durchmesser entlang des gesamten Bronchus zur Verfügung steht, würde sich die starke Verengung an einer Stelle „verwaschen“ und nur marginal auf den Gesamtquerschnitt des Astes auswirken. Der zu erwartende deutliche Anstieg des Strömungswiderstands bliebe aus. Ein weiterer Faktor sind die Astlängen: ist ein relativ langer Ast merklich gekrümmt, so wird seine Länge im Basismodell möglicherweise signifikant reduziert (siehe oben). Auch diese Kontraktion hat Auswirkungen, zum Beispiel auf den Strömungswiderstand (Diese sind jedoch nicht so gewichtig wie die eines unkorrekt modellierten Astdurchmessers, da bei der Berechnung des Strömungswiderstandes der Durchmesser in der vierten Potenz eingeht, die Länge nur linear).

Durch die Extraktion von Voxeldaten wird ein Bronchus, der bisher durch ein Segment dargestellt wird, in eine Vielzahl kurzer, linear verketteter Segmente zerlegt. Dabei

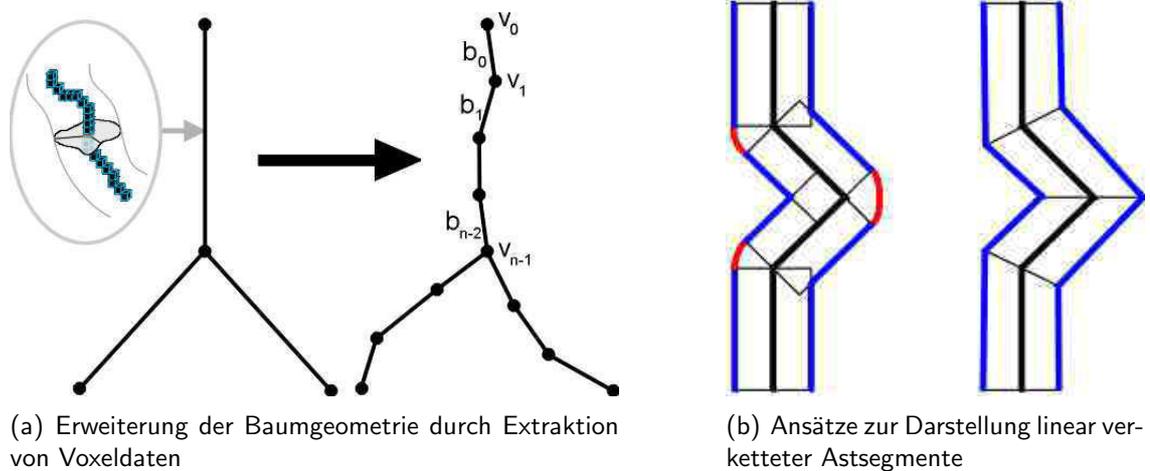


Abbildung 3.8: Erweitertes Modell

werden alle n Voxel v_i der Skelettlinie durch $n - 1$ Äste b_i verbunden (siehe Abb. 3.8(a)). Jedes dieser Teilsegmente ist nunmehr ein eigenständiger Ast, der wie jeder andere durch Anfangs- und Endpunkt sowie proximale und distale Querschnittsfläche definiert wird. Hierbei gilt $A_i^{prox} = A_i$ und $A_i^{dist} = A_{i+1}$, das heißt proximale und distale Querschnittsflächen eines Astes sind genau die gespeicherten Querschnitte der Skelettvoxel, die Start- und Endpunkt definieren (vgl. Abb. 3.5).

Die Attribute eines Teilsegments sind identisch mit den konischen Ästen im Basismodell, und auch die (lineare) Verknüpfung fügt sich nahtlos in das bereits erarbeitete Konzept ein. Die Herausforderung liegt in einem einheitlichen Visualisierungsmodus, der sowohl für das Basismodell als auch für die erweiterte Struktur befriedigende Ergebnisse liefert. Dabei müssen die Schwerpunkte dieser Arbeit in Betracht gezogen werden: die interaktive Auswahl von Ästen mit anschließender morphologischer Statistik und die Visualisierung dynamischer Daten. Dazu ist es vor allem anderen notwendig, jeden (Teil-)Ast identifizieren zu können. Andere Arbeiten [Zim98] modellieren tubuläre Strukturen mittels Spline-Funktionen, wobei ein gesamter Gefäßabschnitt/Bronchus in Form eines einzigen komplexen Objekts dargestellt wird. Diese Methode liefert detaillierte Visualisierungen vor allem im Bereich der Bifurkationen und ist für diagnostische Anwendungen optimiert; für die hier zu lösende Fragestellung muss ein anderer Weg gefunden werden.

Im Basismodell wurden Äste als gerade zylindrische bzw. konische Elemente dargestellt. Die Kreisflächen werden durch N -seitige Polygone angenähert und liegen senkrecht zur Astringung. An den Übergängen zwischen Ästen (linear oder Bifurkation) überlappen sich die Zylinder, und die Lücken werden durch eingesetzte Kugeln geschlossen (vgl. Abb. 3.8(b) links). Für Äste im Basismodell ist dieser Ansatz sinnvoll, für die Teiläste im erweiterten Modell muss jedoch ein anderer Ansatz gefunden werden. Diese Teiläste besitzen nämlich sehr geringe Längen im Voxelbereich (0,5-3 mm) bei teilweise hohen Durchmessern (Trachea: 18 mm). Dadurch würden sich, auch bei moderaten Richtungsänderungen sukzessiver Äste, die Zylinder nicht nur überlappen,

sondern durchstoßen und auf der anderen Seite große Lücken hinterlassen. Der erste Schritt zur Lösung dieses Problems besteht darin, die Segmente durch *schiefe* Kegelstümpfe darzustellen, die jeweils eine Kreisebene mit ihrem Vorgänger bzw. Nachfolger teilen. Die Lage dieser Kreisebene wird durch einen Normalenvektor bestimmt, der auf geeignete Weise zu ermitteln ist. Im Normalfall von relativ langen Ästen (Quotient Länge/Durchmesser ≥ 1) halbieren die Ebenen den Winkel zwischen den beiden Ästen, wie in Abb. 3.8(b) rechts angedeutet. Dies ist die gängige Art der Modellierung in anderen Arbeiten (z.B. [FKFW02]). Dort wird jedoch das Problem der kurzen Äste explizit umgangen, indem die Astlängen durch Zusammenfassen von Ästen auf die Dimensionen des Astdurchmesseres normiert werden. Für die vorliegende Arbeit soll gerade dies vermieden werden, da die hochaufgelösten Durchmesserinformationen und die feine Ortsdiskretisierung für die numerische Simulation verloren gingen. Zur Ermittlung der Lage der Kreisebenen muss daher ein neuer Ansatz gefunden werden. Um befriedigende Ergebnisse zu erreichen, sind mehrere Schritte notwendig. Zunächst ist eine Glättung der Mittelachse unerlässlich.

Glättung der Mittelachse

Durch die diskreten, ganzzahligen Koordinaten der Mittelachsenvoxel ist die Mittelachse, die die Äste des erweiterten Modells definiert, sehr rauh und enthält sprunghafte Winkeländerungen. Dies verfälscht die Länge der Äste und verschärft die Probleme bei der Visualisierung. Wie von [Sel99] vorgeschlagen, soll die Mittelachse daher einer Mittelwertglättung unterworfen werden. Dabei werden die x/y/z-Koordinaten jedes Skelettvoxels separat mit denen ihrer unmittelbaren Nachbarn gemittelt. Die Anfangs- und Endvoxel bleiben unverändert:

$$v'_i = \frac{v_{i-1} + 2v_i + v_{i+1}}{4}, \quad i = 1, 2, \dots, n - 2$$

Diese Glättung kann bei Bedarf mehrmals ausgeführt werden, um den Effekt zu verstärken. Die Auswirkungen auf Länge und Aussehen der Mittelachse werden in Kapitel 5 am konkreten Beispiel diskutiert.

Neben den Koordinaten der Skelettvoxel können auch die Querschnittsangaben geglättet werden, wie von [HPSP01] vorgeschlagen. Dadurch können Diskontinuitäten, die bei typischen radiologischen Datensätzen auftreten, vermindert werden. Das Vorgehen ist analog, in obiger Gleichung sind lediglich die Voxelkoordinaten v_i durch die entsprechenden Querschnittsflächen A_i zu ersetzen. Die Stärke beider Glättungsoperationen soll vom Benutzer parametrisierbar sein.

Ermittlung der Kreisebenen

Eine Kreisfläche wird durch ihren Mittelpunkt \vec{v}_i , den Normalenvektor \vec{n}_i und einen sogenannten Up-Vektor \vec{up}_i definiert. Mittelpunkt und Normalenvektor definieren die Ebene des Kreises; der Up-Vektor ist notwendig, da es sich bei dem dargestellten Objekt nicht tatsächlich um einen Kreis handelt, sondern um eine Approximation in Form eines regelmäßigen N -seitigen Polygons. Bei der Verbindung von je zwei Kreisflächen zu einem konischen Segment ist es wichtig, dass sich die Polygonpunkte möglichst „auf gleicher Höhe“ befinden und die Umlaufrichtung der Punkte identisch ist; andernfalls

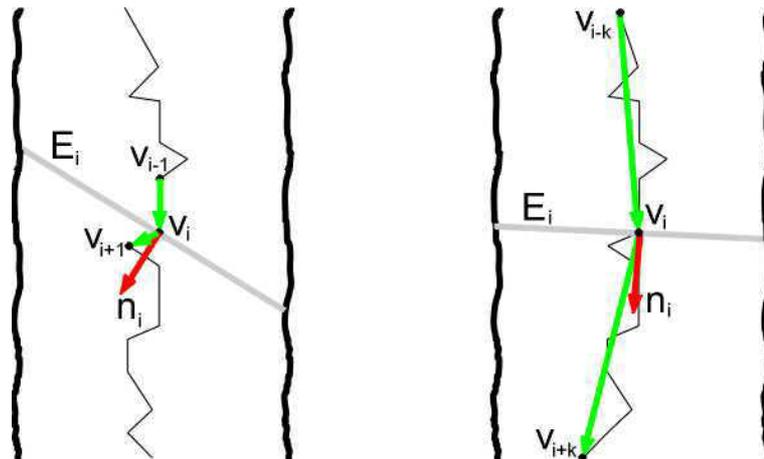


Abbildung 3.9: Bestimmung von Normalenvektoren der Kreisebenen. Links: Nur direkte Nachbarn werden einbezogen, entstehende Ebene ist durch lokale Störungen stark geneigt. Rechts: Einbeziehung fernerer Nachbarn, Störungen werden geglättet.

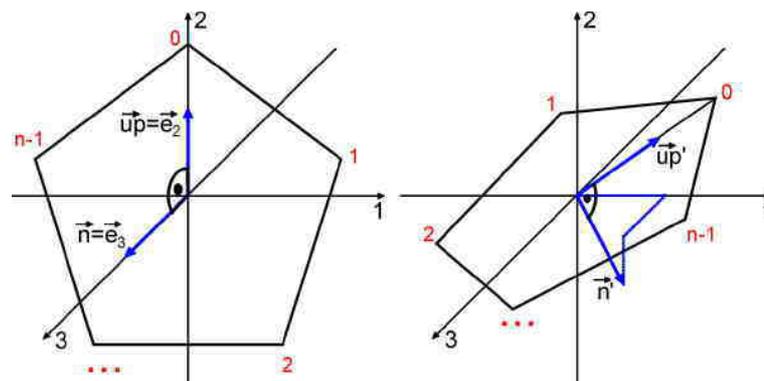


Abbildung 3.10: Rotation einer Kreisfläche, durch Normalenvektor und Up-Vektor definiert

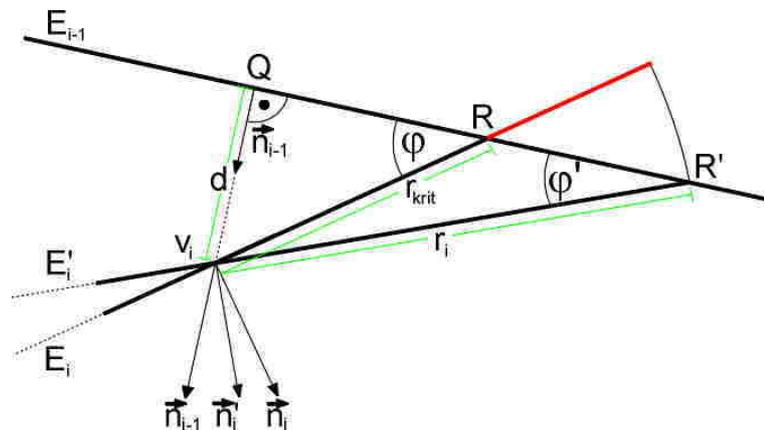


Abbildung 3.11: Lagekorrektur von Kreisflächen; E_i überlappt vorhergehende Ebene E_{i-1} .

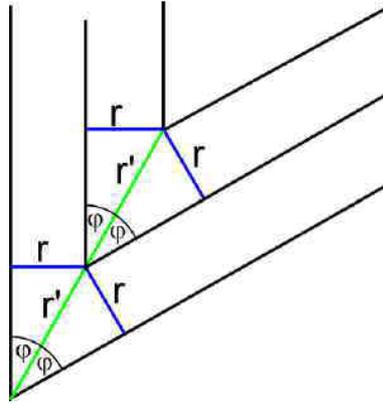


Abbildung 3.12: Korrektur des Kreisflächenradius

kommt es zu Verwindungen der Seitenflächen des Konus. Abb. 3.10 zeigt exemplarisch ein fünfseitiges Polygon im Ursprung. Dieses wird nach Ermittlung von \vec{v}_i , \vec{n}_i und $\vec{u}\vec{p}_i$ entsprechend rotiert und verschoben.

Der Normalenvektor der Kreisebene um den Punkt v_i wird üblicherweise definiert als ein Vektor, der den gleichen Winkel zu den beiden Astvektoren $\vec{b}_{i-1} = \vec{v}_i - \vec{v}_{i-1}$ und $\vec{b}_i = \vec{v}_{i+1} - \vec{v}_i$ bildet, also die Winkelhalbierende. Für sehr kurze Äste werden allerdings nicht die unmittelbaren Nachbarnpunkte mit Indizes $i \pm 1$ verwendet, sondern weiter entfernte Punkte, sodass kleine Störungen herausgemittelt werden. Sei

$$\vec{u} = \vec{v}_i - \vec{v}_{i-k}, \quad \vec{v} = \vec{v}_{i+k} - \vec{v}_i$$

für das kleinste $k > 0$ mit

$$|\vec{u}| \geq |\vec{v}| \geq d_i \quad \left(d_i = \sqrt{\frac{4A_i}{\pi}} \right)$$

Dann ist der Normalenvektor \vec{n}_i ein Vektor mit

$$\angle \vec{u}\vec{n} = \angle \vec{v}\vec{n} \quad \wedge \quad |\vec{n}| = 1$$

Es werden demnach so viele Punkte übergangen, bis der Abstand zum Ursprungspunkt im Bereich des Astdurchmessers liegt. Existieren nicht genügend entfernte Nachbarn, dann muss freilich vorher abgebrochen werden. Mit diesem Vorgehen erreicht man eine sinnvolle Lage der Astebenen, die nicht durch kleine Störungen beeinträchtigt werden, ohne jedoch die hohe Ortsauflösung (=Gesamtanzahl der Äste) zu verlieren. Abb. 3.9 skizziert die Methode.

Die Up-Vektoren von aufeinander folgenden Kreisflächen sollen möglichst in die selbe Richtung weisen. Für die erste Fläche einer Folge von linearen Ästen wird der Up-Vektor willkürlich definiert als irgendein Vektor senkrecht auf \vec{n}_0 . Bei der Berechnung

jeder Folgefläche i ist nun der vorhergehende Up-Vektor $\vec{u}\vec{p}_{i-1}$ bekannt; dieser wird in Richtung \vec{b}_{i-1} auf die neue Kreisfläche projiziert.

Korrektur der Kreisebenen

Mit dem beschriebenen Verfahren ist es nicht ausgeschlossen, dass sich eine neu berechnete Kreisebene mit der darüberliegenden überlappt. Daher ist es notwendig, eine Prüfung und gegebenenfalls Korrektur jeder neuen Ebene durchzuführen. Abb. 3.11 zeigt die Ebene E_{i-1} einer bereits berechneten Kreisfläche und die neu ermittelte Kreisfläche E_i . Diese bildet einen zu großen Winkel zu E_{i-1} und überlappt jene daher. Die Abbildung zeigt einen zweidimensionalen Schnitt in der Ebene, die durch die beiden Normalenvektoren \vec{n}_{i-1} und \vec{n}_i aufgespannt wird. Der Mittelpunkt v_i der Kreisfläche E_i liegt in dieser Ebene, ebenso die Punkte $Q, R \in E_{i-1}$. Q ist der Lotfußpunkt von v_i auf E_{i-1} . R ist derjenige Punkt auf der Schnittgeraden der beiden Ebenen, der v_i am nächsten ist. Die Distanz $d = |\overrightarrow{QV_i}|$ ist bekannt, ebenso der Winkel φ zwischen \vec{n}_i und \vec{n}_{i-1} , sodass man den kritischen Radius r_{krit} bestimmen kann:

$$r_{krit} = \frac{d}{\sin \varphi}$$

Ist der tatsächliche Radius der Kreisfläche r_i größer als r_{krit} , so muss E_i durch Rotation des Normalenvektors \vec{n}_i in Richtung zu \vec{n}_{i-1} in eine neue Ebene E'_i überführt werden, für die $r'_{krit} = r_i$. Dazu ermittelt man den gewünschten Winkel zu

$$\varphi' = \arccos \frac{d}{r_i}$$

und rotiert \vec{n}_i um $(\varphi - \varphi')$ um die Achse $\vec{u} = \vec{n}_i \times \vec{n}_{i-1}$ ¹.

Neben der Lage der Kreisflächen muss auch deren Radius bestimmt werden. Die verfügbare Durchmesserinformation stellt den Astquerschnitt an einer Stelle dar. Da die Kreisebenen nicht senkrecht auf den Ästen stehen, muss ihr Radius entsprechend vergrößert werden. Abb. 3.12 zeigt ein deutlich ausgeprägtes Beispiel; in der Praxis sind die Winkelunterschiede weitaus geringer. Der Radius der Kreisfläche r' berechnet sich aus dem gegebenen Astradius zu

$$r' = \frac{r}{\sin \varphi}$$

wobei φ der Winkel zwischen einem Ast² und der Kreisebene ist.

Bifurkationspunkte

Mit den bisher beschriebenen Mitteln lässt sich ein unterteilter Ast von einem Bifurkationspunkt zum nächsten visualisieren. Die Modellierung der Bifurkationen selbst

¹Dies ist das Kreuzprodukt, es gilt daher $\vec{u} \perp \vec{n}_i \wedge \vec{u} \perp \vec{n}_{i-1}$.

²Bei Mittelung über die k nächsten Nachbarn werden freilich diese Richtungsvektoren und nicht die unmittelbaren Astvektoren zugrunde gelegt.

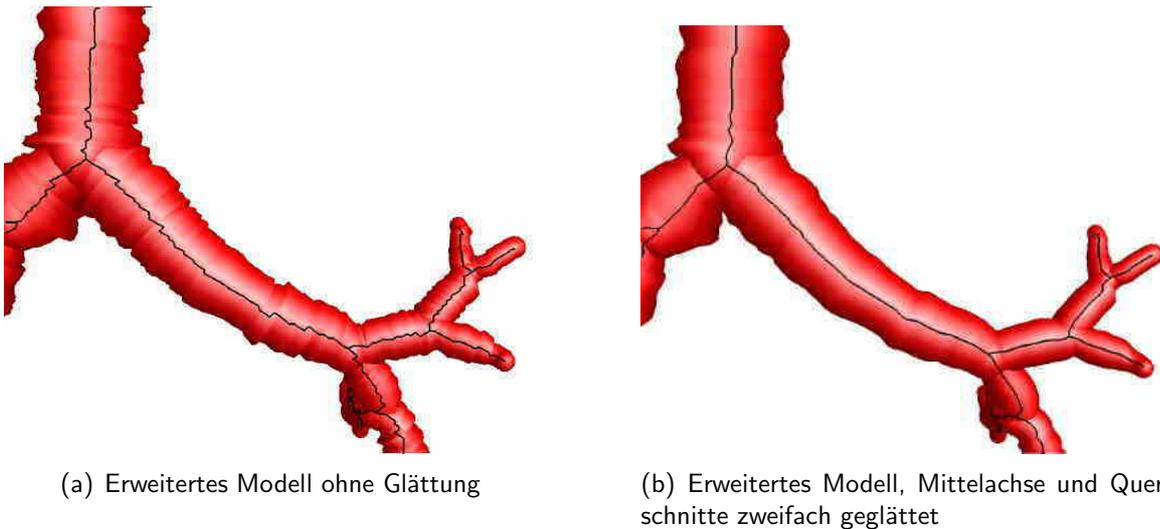


Abbildung 3.13: Obere Atemwege im erweiterten Modell, Oberflächendarstellung inklusive überlagerter Mittelachse

wird wie im Basismodell gehandhabt, indem zuerst jeder Ast separat dargestellt wird und schließlich eine Halbkugel in Richtung des Eltern-Astvektors ins Zentrum der Bifurkation modelliert wird. Dies führt zu annehmbaren Ergebnissen für den anvisierten Anwendungszweck. Andere Autoren modellieren die Bifurkationen mit erheblich größerem Aufwand; für die morphologische Analyse und die Visualisierung von dynamischen Daten ist die hier gewählte Darstellung ausreichend. Abb. 3.13 zeigt die komplette Visualisierung der oberen Atemwege im erweiterten Modell exemplarisch auf. Man erkennt die deutliche Verbesserung durch Glättung der Skelettlinie auf der rechten Seite.

3.3 Visualisierung dynamischer Daten

Eine wichtige Anwendung der hier entwickelten Applikation im Rahmen des Lungenprojekts ist die Visualisierung von Simulationsergebnissen, beispielsweise Sauerstoffkonzentrationen in der Lunge. Es ist daher ein Konzept zu entwickeln, um solche dynamischen Daten, die der Baumstruktur zugeordnet werden können, darzustellen. Der Einsatz verschiedener Farbkodierungen ist das Mittel der Wahl zur Visualisierung dieser Daten.

Jeder Ast des visualisierten Baums besitzt eine eindeutige ID; diese wird für die Zuordnung von dynamischen Daten genutzt. Der dynamische Aspekt entsteht dadurch, dass Daten zu verschiedenen, diskreten Zeitpunkten vorliegen können, die sukzessive visualisiert werden sollen. Die zu verarbeitenden Informationen kann man sich daher als zweidimensionales Feld $\text{DATEN}[\text{AstID}][\text{Zeit}]$ vorstellen. Der Benutzer soll die Möglichkeit haben, einen Zeitpunkt zu wählen oder automatisch eine Folge von Bildern zu generieren, um die aktuell anzuzeigenden Daten zu wählen. Die konkrete Realisierung

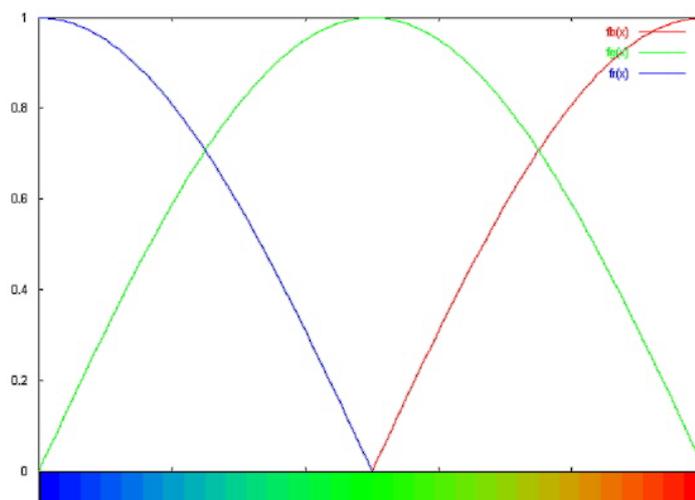


Abbildung 3.14: Farbfunktion Temperaturskala

der Benutzerführung findet sich im Implementierungskapitel; hier werden lediglich die Möglichkeiten zur Kodierung der Daten in Farbwerte diskutiert.

Farbkodierung

Gesucht ist eine Methode, die einem Konzentrationswert oder einem beliebigen anderen numerischen Wert eine Farbe zuordnet. Die numerischen Werte sind Fließkommazahlen; eine Farbe lässt sich auf verschiedene Arten definieren. Im RGB-Farbmodell (vgl. [Hab91]) wird eine Farbe als Zahlentripel (R,G,B) definiert, das die Intensitäten der drei Grundfarben Rot, Grün und Blau angibt. Das Tripel (0,0,0) stellt Schwarz dar, (1,1,1) die Farbe Weiß (setzt man Intensitäten im Einheitsintervall voraus). Nimmt man der Einfachkeit halber ebenfalls an, dass die numerischen Werte auf das Intervall $[0;1]$ normiert sind, dann ist eine Zuordnungsvorschrift

$$f : [0; 1] \mapsto [0; 1]^3$$

gesucht, die einem reellen Wert zwischen Null und Eins eindeutig ein RGB-Tripel zuweist. Mit dieser Zuordnung kann eine Farbkodierung realisiert werden. Es gilt nun, sinnvolle Funktionen dieser Art zu definieren, die dem Benutzer Informationen vermittelt. Eine Möglichkeit ist eine temperaturartige Skala, die große Werte mit Rot („heiß“), kleine mit Blau („kalt“) kodiert. Für eine Darstellung der entsprechenden vektorwertigen Funktion kann man die drei Teilfunktionen f_r, f_g, f_b , die jeweils eine Farbkomponente zuordnen, grafisch darstellen. Die beschriebene Temperaturskala ist in Abb. 3.14 zu sehen, die einzelnen Funktionen lauten:

$$f_r(x) = \max\{\sin(\pi(x - 0.5)), 0\}, \quad f_g(x) = \sin(\pi x), \quad f_b(x) = \max\{\sin(\pi(x + 0.5)), 0\}$$

Eine simple Möglichkeit ist die, eine Grauwertskala zu definieren, die den Werten von

Null bis Eins Grautöne von Schwarz bis Weiß zuordnet. Die drei Teilfunktionen sind dann

$$f_r(x) = x, \quad f_g(x) = x, \quad f_b(x) = x,$$

also gerade Identitätsfunktionen, die dem Wert x das RGB-Tripel (x,x,x) zuweisen:



Eine weitere bekannte Skala sind die „Feuerfarben“ (engl. *heatmap*), die von Schwarz über Rottöne nach Weiß gehen:



Zumindest diese drei Alternativen sollen in der Applikation verfügbar sein. Eine interaktive Auswahl der Transferfunktionen wäre prinzipiell machbar, soll aber im Rahmen dieser Arbeit nicht realisiert werden.

Lookup-Tabellen

Bisher wurden die Farbkodierungen formal als reellwertige Funktionen beschrieben, die einen Wert $x \in [0; 1]$ in ein RGB-Tripel umsetzen. In der Praxis hat man Fließkommazahlen in einem anderen Bereich $[0; x_{max}]$. Es wäre nun ungünstig, für die Bestimmung jeder Farbe erneut drei möglicherweise aufwendige mathematische Funktionen wie $\sin(x)$ zu berechnen. Stattdessen kann man anstelle des kontinuierlichen Wertebereichs eine feste Anzahl diskreter Farbwert im Voraus berechnen (üblicherweise 256) und in einer Lookup-Tabelle (LUT) die entsprechenden RGB-Werte speichern. Der Farbwert eines gegebenen x findet sich dann an der Stelle

$$\text{index} = \lfloor x/x_{max} \cdot 255 \rfloor^3$$

in der LUT. Bildbearbeitungsprogramme erlauben es, solche Tabellen zu erstellen und zu editieren. In der hier entwickelten Applikation sollen zunächst nur die drei vorgestellten Farbkodierungen wählbar sein. Eine frei editierbare LUT wird nicht realisiert.

3.4 Statistik

Es existiert eine Vielzahl von Veröffentlichungen zur Morphometrie des Bronchialbaums [Wei63, HC67, HC68, BEP⁺], die auf einer exakten Messung und statistischen Auswertung von geometrischen Daten über die Äste und deren Zusammenhang basieren. E.

³ $\lfloor x \rfloor$ bezeichnet die größte ganze Zahl $\leq x$.

Weibel, ein bedeutender Anatom, hat 1963 ein Standardwerk zur Lungenmorphometrie verfasst. Mit Präparationstechniken erzeugte er ein Ausgussmodell des Bronchialbaums und vermaß dieses manuell mit der Schieblehre. Auf Basis dieser Daten entwarf er das symmetrische Generationenmodell; in Lehrbüchern findet man seine Daten zu mittleren Längen und Durchmessern pro Generation. Er wies allerdings auch auf Asymmetrien in der Struktur hin. Neuere Publikationen stellen diesen Aspekt in den Vordergrund und beschäftigen sich auch mit fraktalen Eigenschaften, beispielsweise unter dem Gesichtspunkt, wie die Form des Bronchialbaums Energiebedarf und Luftflüsse optimiert.

Die digitale Erfassung mit hoher Auflösung, wie sie zumindest für Ausgusspräparate mittels HRCT erzielt werden kann, weckt den Wunsch nach einer effizienten Möglichkeit, solche statistischen Erhebungen computergestützt durchzuführen. Die in dieser Arbeit verwendete Repräsentation des Bronchialbaums macht dies möglich. Es soll daher im Folgenden untersucht werden, welche Eigenschaften man dem Bronchialbaum zuordnen kann und auf welche Art und Weise sie am besten dem Benutzer zugänglich gemacht werden. Dabei spielen die Aspekte Flexibilität, Schnelligkeit und Benutzerfreundlichkeit eine prominente Rolle.

3.4.1 Eigenschaften von Ästen

Eigenschaften, die man Bronchialsegmenten zuordnen kann, können einfach oder zusammengesetzt sein. Unter einfachen Eigenschaften seien Attribute eines Asts gemeint, die dem Ast als solchem zueigen sind. Es sind dies zum Beispiel Länge und Volumen. Zusammengesetzte Eigenschaften ergeben sich erst aus der Lage eines Asts in der Gesamtstruktur. Die Generationentiefe ist ein Beispiel, aber auch Maßzahlen, die das Verhältnis eines Astes zu seinen direkten Nachbarn bestimmen (siehe unten). Dennoch lassen sich alle diese Eigenschaft genau einem Ast zuordnen, und können daher gleich behandelt werden. Einzig in der Berechnung unterscheiden sie sich. Es folgt nun eine Beschreibung der Eigenschaften, die erfasst werden sollen.

Länge [mm]

Die Länge eines Astes ist eine einfache Eigenschaft, die ohne Zweifel erfasst werden muss. Da im vorliegenden Modell die Äste als gerade, konische Elemente angesehen werden, ist die Länge die direkte Entfernung zwischen Start- und Endpunkt.

Volumen [mm³]

Das Gesamtvolumen eines konischen Astes ergibt sich aus dessen Länge sowie distaler und proximaler Querschnittsfläche.

Durchmesser [mm] und Querschnittsfläche [mm²]

Äste sind zylindrisch und haben daher einen kreisrunden Querschnitt; Durchmesser und Querschnittsfläche berechnen sich aus Volumen und Länge des Astes.

Weibel-Generation

Gibt die Tiefe eines Astes im Baum wie in Abschnitt 2.1.1 beschrieben an. Dabei werden nur Bifurkationen gezählt, lineare Ketten von Segmenten haben jeweils die

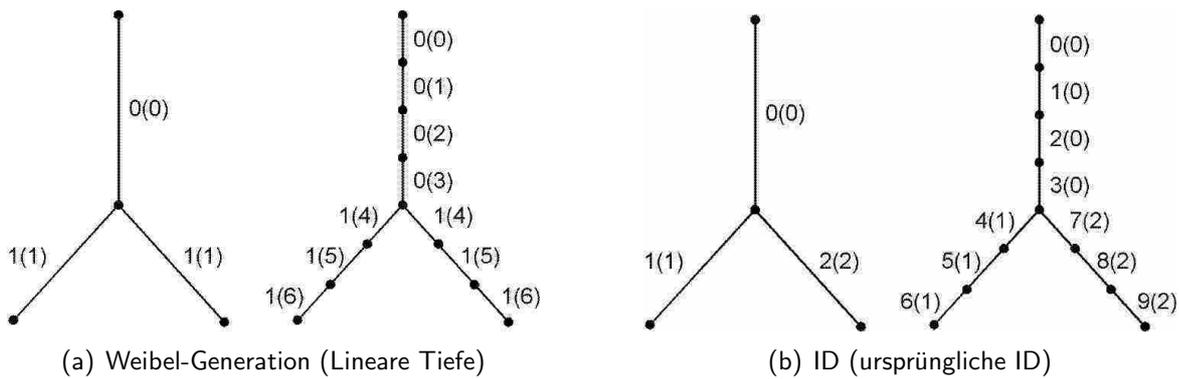


Abbildung 3.15: Vergleich verschiedener Maßzahlen im Basis- und im erweiterten Modell

gleiche Generation.

Lineare Tiefe

Gibt die Tiefe eines Astes im Baum an, wobei im Gegensatz zur Weibel-Generation auch lineare Verknüpfungen gewertet werden. Im Basismodell, in dem ausschließlich Bifurkationen existieren, ist diese Tiefe mit der Weibel-Generation identisch. Im erweiterten Modell wird jedem Teilsegment eines Bronchus' eine neue lineare Tiefe zugewiesen, wohingegen die Weibel-Generation gleich bleibt. Abb. 3.15(a) macht diesen Unterschied deutlich.

Horsfield-Ordnung

Gibt die Lage eines Astes im Baum wie in Abschnitt 2.1.1 beschrieben an. Auch hier werden wie bei der Weibel-Generation nur Bifurkationen gewertet.

Horsfield-Delta

Der Horsfield-Delta-Wert ist ein Maß für die Asymmetrie eines Baums an einer bestimmten Stelle. Er berechnet sich aus dem Betrag der Differenz der Horsfield-Ordnungen der beiden Kinder des betrachteten Astes.

Bifurkationsexponent

Der Bifurkationsexponent berechnet sich aus dem Verhältnis der Durchmesser eines Astes und seiner Kinder. Sei d der Durchmesser des Elternastes und d_1, d_2 die Durchmesser der beiden Kinder; dann erfüllt der Bifurkationsexponent die Gleichung $d^x = d_1^x + d_2^x$.

Physikalische Resistance [kg/(s·m⁴)]

Die Resistance gibt den viskosen Strömungswiderstand eines Objektes an. Für annähernd zylindrische Körper berechnet er sich zu $R = K \cdot L/r^4$, wobei L die Länge und r der Radius des Astes ist. K ist eine Konstante.

Pulmonaler Druck [Pa]

Diese Eigenschaft ist nur für Blätter definiert. Gibt den lokalen Druck(-unterschied) an, der bei der Inspiration durch die Ausdehnung des Brustkorbes erzeugt wird und der

den Atmungsprozess möglich macht. Diese Drücke werden für die Simulation benötigt, können aber nicht aus der Baumgeometrie selbst errechnet, sondern müssen anderweitig ermittelt werden. Dies ist nicht Aufgabe dieser Arbeit, aber es ist möglich, die Druckwerte beliebiger Blätter manuell zu verändern (analog zu den Eigenschaften „Lappen- und Segmentnummer“).

Pfadlänge zur Carina [mm]

Die Carina ist der Punkt, in dem sich die Trachea in die beiden Hauptbronchi verzweigt, allgemein also die erste Bifurkation distal zur Wurzel des Baumes. Er wird oft als Referenzpunkt herangezogen. Die Pfadlänge zur Carina eines jeden Astes (außer der Wurzel selbst) ist die Summe aller Astlängen auf dem Weg von der Carina zum gegebenen Ast. Der Ast selbst wird nicht mitgezählt; der Pfad endet also beim Beginn des Astes.

Versorgtes Volumen [ml]

Dies ist eine Eigenschaft, die speziell der Lunge zueigen ist. Jedes Blatt im segmentierten Bronchialbaum, also ein Bronchus oder eine Bronchiole an der Grenze des Auflösungsvermögens des CT, versorgt ein gewisses Lungenvolumen, das heißt eine Menge von respiratorischen Einheiten mit Alveolen. Dieses Maß ist für die Flussaufteilung in der Simulation essentiell und wird bei der Segmentierung des Baumes für jedes Blatt bestimmt. Für alle anderen Äste ist das versorgte Volumen rekursiv definiert als die Summe der versorgten Volumina der Tochteräste.

Flächenasymmetrie

Vergleicht die Flächen der beiden Kindäste. Die Flächenasymmetrie berechnet sich aus A_1/A_2 , wobei A_1 die Querschnittsfläche des dickeren Tochterastes ist, A_2 die des dünneren.

Lappen- und Segmentnummer

Die anatomische Einteilung des Bronchialbaums in Lappen und Segmente erfolgt nach einem festen Schema. Da für verschiedene Segmente unterschiedliche morphologische Ausprägungen (zum Beispiel zur mittleren Pfadlänge zur Carina oder dem versorgten Volumen) erwartet werden, ist es sinnvoll, Äste derart einzuteilen. Diese Einteilung muss manuell vom Benutzer vorgenommen werden. Es wurden Möglichkeiten zu einem automatisierten Labeling vorgeschlagen [KPT⁺02], die jedoch in dieser Arbeit nicht implementiert werden.

Ast-ID

Zur eindeutigen Identifizierung ist selbstverständlich auch die ID des Astes abfragbar. Damit können Bezüge zu anderen Daten, beispielsweise Simulationsergebnissen, hergestellt werden.

Ursprüngliche Ast-ID

Im erweiterten geometrischen Modell werden die Voxelinformationen dazu verwendet, einen Ast in mehrere kurze Teilsegmente zu zerlegen, die linear hintereinander angeordnet sind. Um Vergleiche mit dem Basismodell ziehen zu können und die Teilsegmente

mevis-id	weibel-gen	diameter
0	0	18.3403
1	1	12.3037
14	1	11.1726
2	2	7.96637
9	2	8.37932
15	2	7.04088
21	2	7.04327
3	3	5.42413
6	3	6.54837
10	3	7.69997
11	3	9.92601
16	3	5.21813
18	3	6.31037
22	3	7.19603
25	3	4.89591

 \Rightarrow

weibel-gen	MW diameter
0	18.3403
1	11.7382
2	7.60746
3	6.65237

Tabelle 3.1: Durchmesser pro Weibel-Generation

bei der statistischen Auswertung zusammenzufassen, wird bei der Unterteilung der Äste die ID des ursprünglichen Astes gespeichert. Über die Gruppierungsfunktion (s. unten) kann so beispielsweise die Länge eines Astes als die Summe der Längen seiner Teilstücke bestimmt werden. Abb. 3.15(b) zeigt den Zusammenhang zwischen den beiden Kennzahlen auf.

Neben diesen Eigenschaften sollen auch Abfragen möglich sein, ob ein Ast die Wurzel des Baums darstellt, ob er ein Blatt ist oder ob er zwei Kinder besitzt.

3.4.2 Erfassung und Verarbeitung

Die Applikation soll die Möglichkeit bieten, die genannten Eigenschaften für größere Mengen von Ästen zu erfassen, in tabellarischer Form darzustellen und eine Weiterverarbeitung etwa in MS Excel oder einem Statistiktool wie Origin zu ermöglichen. Hierzu ist eine Exportmöglichkeit im Textformat vorzusehen. Vor der Erfassung der Daten muss der Benutzer eine Teilmenge von Ästen auswählen können, deren Eigenschaften betrachtet werden sollen. Dies wird eingehend in Abschnitt 4.4.2 beschrieben. Die einfachste Möglichkeit zur Ausgabe von Asteigenschaften ist es, eine Menge von n Attributen auszuwählen und eine n -spaltige Liste zu generieren, die eine Zeile für jeden Ast der Teilmenge aufweist. So könnte man sich beispielsweise einen Überblick über die Durchmesser der Bronchien auf den ersten drei Generationen verschaffen, indem man zuerst alle Äste mit Weibel-Generation von 0 bis 3 selektiert und dann durch Angabe der benötigten Attribute eine Tabelle generieren lässt (Tab. 3.1 links).

Zusätzlich zu dieser einfachen Ausgabe soll die Möglichkeit bestehen, nach einer bestimmten Eigenschaft zu gruppieren: im Beispiel führt eine Gruppierung nach Weibel-

Generation und Mittelwertbildung der Durchmesser-Eigenschaft zu der in Tab. 3.1 rechts angegebenen Tabelle, die die mittleren Durchmesser pro Generation angibt, wie sie in Lehrbüchern angegeben werden. Die Gruppierung soll nicht auf ganzzahlige Werte beschränkt sein; durch Angabe des gewünschten Wertbereichs und der Gruppierungsschrittweite sollen beliebige Attribute als Gruppierungseigenschaft wählbar sein. Damit ist beispielsweise folgende Abfrage möglich: „Welche mittleren Längen haben Äste mit den Durchmessern in den Intervallen 1-2mm, 2-3mm, . . . 9-10mm“ ?

Ein letztes Anwendungsbeispiel zeigt, dass eine weitere Asteigenschaft im Zusammenhang mit der Gruppierung nützlich ist: ein Attribut namens **count**, das konstant den Wert Eins liefert. Zusätzlich soll die Option möglich sein, nicht den Mittelwert (wie oben), sondern die Summe der gruppierten Einträge zu bilden. Stellt man sich die Aufgabe, die Anzahl der Äste in jeder Verzweigungsgeneration zu analysieren, so kann man folgendermaßen vorgehen:

- Gruppierere nach Eigenschaft „Weibel-Generation“ (Schrittweite 1).
- Gib die Eigenschaft „count“ aus.
- Bilde nicht den Mittelwert, sondern die Summe der gruppierten Einträge.

Als Ergebnis erhält man eine Tabelle mit zwei Spalten, „weibel-gen“ und „SUM count“ (nicht abgebildet). Da jeder Ast den Wert Eins für „count“ beisteuert, ist diese Summe gerade die Anzahl der Äste auf der jeweiligen Generation. Eine Mittelwertbildung wie oben macht hier keinen Sinn, der Mittelwert wäre konstant Eins. Stattdessen wird die Summe gebildet.

Die beschriebenen Mittel zur Datenerfassung reichen aus, um flexibel und schnell Rohdaten zu erheben; eine statistische Analyse oder grafische Darstellung mit externen Programmen ist problemlos möglich. Da auch die ID jedes Astes als Eigenschaft verfügbar ist, sind Verknüpfungen mit anderweitig erhobenen Daten möglich, beispielsweise der Vergleich dynamischer Simulationsergebnisse mit strukturellen Daten.

4 Implementierung

Das erarbeitete Konzept soll in Form einer eigenständigen Applikation mit grafischem Benutzungsinterface (GUI) realisiert werden. Die Implementierung erfolgt in C++, die Visualisierung der Baumstruktur erfolgt mithilfe der Grafikbibliothek OpenGL [WND97]. Das GUI wird auf Basis der Klassenbibliothek Qt der Firma Trolltech entwickelt, die sowohl unter MS Windows wie auf UNIX-Derivaten verfügbar ist. Dies ermöglicht einen portablen Einsatz des Programms.

4.1 Datenstrukturen

Zunächst werden die grundlegenden internen (Klassen) und externen (Dateiformate) Datenstrukturen beschrieben, die in der Applikation zum Einsatz kommen. In der Hauptsache sind dies zwei Repräsentationen der Baumgeometrie: eine Klasse, die die verfügbaren Daten des gerichteten Graphen ungefiltert übernimmt, und eine nachgeschaltete Klasse, die die in der Konzeption erläuterten Transformationen durchführt und alle Schnittstellen zur Interaktion und Analyse bereitstellt.

4.1.1 Mevis-Dateiformat

Die Mevis-Daten liegen im Textformat vor, in dem die Knoten und Kanten des Graphen und einige globale Informationen gespeichert werden. Der Kopf der Datei hat üblicherweise folgenden Inhalt:

```
TreeData Version 6
  ImageSize: 512 512 572
  VoxelSize: 0.3516 0.3516 0.4000
  nodeNum: 1634
  edgeNum: 1642
  rootNum: 1 rootIDs: 0
```

`ImageSize` gibt die Größe der 3D-Voxelmatrix an, auf dessen Basis die Struktur erzeugt wurde. Die Koordinaten von Knoten müssen daher in diesem Bereich liegen. `VoxelSize` spezifiziert die Dimensionen eines Voxels in Millimetern. Meist sind die Größen in x- und y-Richtung identisch; die Dimension entlang der z-Achse, die senkrecht zu den CT-Schnitten verläuft, ist in der Regel unterschiedlich. Man sagt, die Voxel sind *anisotrop*. `nodeNum` und `edgeNum` geben die Gesamtanzahl der Knoten bzw. Kanten in der Datei an. `rootNum` und `rootIDs` sind wichtige Angaben, die den Einstiegspunkt bei der Analyse des Graphen markieren. Wie in Abschnitt 3.2.1 erläutert, muss ein Baum genau eine Wurzel besitzen; `rootNum` muss also Eins sein. Dann gibt `rootIDs` die Identifikationsnummer dieses Wurzelknotens an.

Der Rest der Datei besteht aus einer Liste aller Knoten und Kanten. Die Einträge für Knoten haben folgendes Format:

```

NodeID: 1
  Koord: 461 214 366
  minDist: 761
  edgeNum: 3
    IDs: 0 1 2
    angles: 0.000000 135.818710 165.019714
            135.818710 0.000000 58.592804
            165.019699 58.592804 0.000000

```

Jeder Knoten besitzt eine eindeutige `NodeID`; die IDs beginnen bei Null und sind in aufsteigender Reihenfolge ohne Lücken vergeben. Unter `Koord` findet man die Voxelkoordinaten des Knotens, die mithilfe der bekannten Voxeldimension in Millimeter umgerechnet werden können. `edgeNum` gibt Auskunft darüber, wie viele Kanten mit dem Knoten verbunden sind. Für jede dieser Kanten folgt danach ihre Kanten-ID unter `IDs` sowie die Winkel der Kante zu den drei Koordinatenachsen unter `angles`. Nachdem alle Knoten definiert wurden, folgen die Kanten:

```

EdgeID: 5
  predID: 3, succID: 4
  cyc: 0
  Hiera: 3
  avminDist: 726.333313
  Length: 1.133981
  Volume: 56.310589
  VoxLabInd: 0 0 0
  VoxNum: 3
    Koords: 462 216 361 461 216 362 460 215 363
    minDist: 721 729 729
    maxDist: 770 770 0
    NumVox: 1102 34 0
    Area: 43.064620 74.433911 90.238836
    LabNum: 4
      Labs: 0 0 0
      Labs: -13108 -13108 -13108
      Labs: -13108 -13108 -13108
      Labs: -13108 -13108 -13108

```

Auch Kanten besitzen eine eindeutige `EdgeID`. `predID` sowie `succID` geben die `NodeIDs` der Anfangs- und Endknoten der Kante an (*predecessor* und *successor*). Der Angabe `VoxNum` folgen alle Daten zur Skelettlinie (vgl. Abschnitt 3.2.1.3), vor allem die Querschnittsfläche zu jedem Skelettvoxel (`Area`).

4.1.2 Die Klasse `MevisFile`

Die Klasse `MevisFile` bildet die Schnittstelle zu den als Datei vorliegenden Graphenstrukturen, die die Basis für die Baumgeometrie bilden. Als solche stellt sie eine mehr oder minder direkte Repräsentation der Daten dar, wie sie in der Datei zu finden sind. Die Informationen werden bei Objektkonstruktion aus der ASCII-Datei oder der binären Darstellung gelesen. Intern werden zwei Hilfsklassen, `Node` und `Edge`, benutzt, die reine Datenklassen sind und genau die in der Datei enthaltenen Attribute speichern.

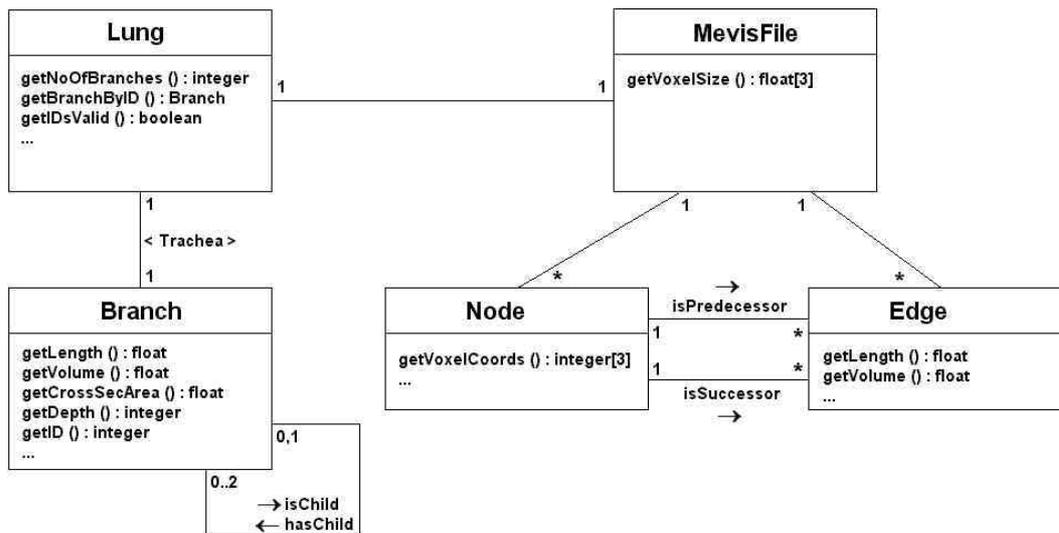


Abbildung 4.1: Klassendiagramm zum Verhältnis der Klassen MevisFile und Lung

Ein `MevisFile`-Objekt besitzt also zwei dynamische Felder für diese Knoten- und Kantenobjekte und Elementvariablen für die bereits beschriebenen globalen Attribute.

Eine Analyse der Graphenstruktur findet an dieser Stelle nicht statt. Lediglich die Anzahl der Kanten und Knoten in der Datei müssen mit den im Dateikopf angegebenen Werten übereinstimmen und die IDs eindeutig und vollzählig sein. Auch werden Graphen mit mehr als einem Wurzelknoten nicht akzeptiert. Multifurkationen oder Zyklen werden hingegen nicht beanstandet.

4.1.3 Die Klassen Branch und Lung

Ein Objekt vom Typ `MevisFile` bildet die Rohdaten aus der Graph-Datei unmittelbar ab. Im Grunde könnte man auf dieser Struktur von Knoten und Kanten bereits operieren; alle Informationen sind dort vorhanden. Wie in Abschnitt 3.2.1 beschrieben, ist die Graphenstruktur jedoch mächtiger als für dieses Projekt gewünscht. Die ungefilterte Nutzung einer solchen Struktur würde einige Nachteile mit sich bringen:

- Es ist nicht garantiert, dass der Graph die geforderte Struktur besitzt (binärer Baum).
- Die Informationen über die geometrischen Äste sind noch „roh“ und über Knoten und Kanten verstreut.
- Die Navigation über Knoten und Kanten ist umständlich.

Aus diesen Gründen erscheint es angebracht, die Graphenstruktur in eine einfacher zu handhabende Form zu transformieren. Gewünscht ist eine Form, die alle Daten einzelner Äste zusammenfasst und die Verzweigungsstruktur mittels einfacher „ist-Kind-von“ und „ist-Elternast-von“ Beziehungen ausdrückt. Der Vorteil einer solchen Struktur wird

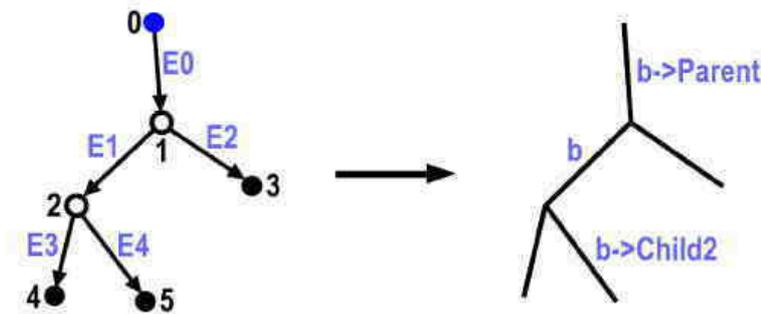


Abbildung 4.2: Transformation eines Mevis-Graphen in eine Lung-Struktur. Links: Knoten und Kanten des Mevis-Graphen. Rechts: Transformierte Struktur verbundener Branches mit exemplarischer Parent-Child-Beziehung

schnell ersichtlich, wenn man sich an Hand eines Beispiels die Navigation in einem Baum verdeutlicht. Abb. 4.2 zeigt einen einfachen Baum, links im Mevis-Format mit sechs Knoten und fünf Kanten. Die Laufrichtung der gerichteten Kanten sei implizit von oben nach unten festgelegt. Die entsprechenden Einträge in der Mevis-Datei sehen wie folgt aus, wobei aus Gründen der Übersichtlichkeit lediglich die Verzweigungsinformationen angegeben und geometrische Angaben weggelassen wurden. Trotzdem sei daran erinnert, dass die End- und Anfangskoordinaten eines Astes durch die Knoten, andere geometrische Daten aber durch die Kanten transportiert werden (vgl. Abschnitt 4.1.1).

```

rootNum: 1          rootIDs: 0

NodeID: 0          NodeID: 1          NodeID: 2
  edgeNum: 1        edgeNum: 3          edgeNum: 3
    IDs: 0           IDs: 0 1 2          IDs: 1 3 4

NodeID: 3          NodeID: 4          NodeID: 5
  edgeNum: 1        edgeNum: 1          edgeNum: 1
    IDs: 2           IDs: 3             IDs: 4

EdgeID: 0          EdgeID: 1
  predID: 0, succID: 1  predID: 1, succID: 2

EdgeID: 2          EdgeID: 3
  predID: 1, succID: 3  predID: 2, succID: 4

EdgeID: 4
  predID: 2, succID: 5

```

Ein Abstieg in den Baum, der beispielsweise zu dem Ast führt, der durch Kante E4 und Knoten 3 und 5 definiert wird, müsste folgendermaßen aussehen:

- Ermittle den Wurzelknoten \Rightarrow Knoten 0
- Knoten 0 ist mit einer Kante verbunden \Rightarrow Kante E0

- Folgeknoten von Kante E0 \Rightarrow Knoten 1
- Knoten 1 ist mit drei Kanten verbunden \Rightarrow Kanten E0, E1, E2
- E0 ignorieren, diese Kante ist Vorgängerkante zum aktuellen Knoten.
- E1 u. E2 haben Knoten 1 als Vorgänger, sind daher dessen Nachfolgerkanten.
- Wähle E1. Ermittle ihren Endknoten \Rightarrow Knoten 2
- Knoten 2 ist mit drei Kanten verbunden \Rightarrow Kanten E1, E3, E4
- E1 ignorieren, diese Kante ist Vorgängerkante zum aktuellen Knoten.
- E3 u. E4 haben Knoten 2 als Vorgänger, sind daher dessen Nachfolgerkanten.
- Wähle E4. Die benötigten geometrischen Informationen finden sich nun in dieser Kante und dem Vorgänger- und Nachfolgerknoten 2 und 5.

Es sind also umständliche Prüfungen notwendig, die wechselnde Einsicht in die Informationen von Knoten und Kanten erfordern und wenig verständlich sind. Darüber hinaus müsste an jeder Stelle getestet werden, ob auch tatsächlich eine korrekte Baumstruktur vorliegt. Eine eingängigere Art des Umgang mit einem Baum ist die folgende: jeder Ast wird durch genau ein Objekt vom Typ **Branch** repräsentiert, das die Informationen einer Kante und ihrer Knoten enthält. Die Navigation erfolgt durch Ausnutzen einfacher (durch Zeiger realisierter) Verknüpfungen zwischen den **Branches**. Ist nun der Wurzelast bekannt, können durch einfache Operationen beliebig viele Ebenen des Baumes referenziert werden:

- Ermittle den Wurzelast \Rightarrow **Branch root**, entspricht Graphkante E0
- $b := \text{root} \rightarrow \text{Child1}$ vorhanden; entspricht Graphkante E1
- $b \rightarrow \text{Child2}$ vorhanden; entspricht Graphkante E4. Alle benötigten geometrischen Informationen finden sich nun in diesem Ast.

Dies ist freilich ein konstruiertes Beispiel, es demonstriert aber gut die vereinfachte Navigation. In der ersten Phase der Transformation werden in der Tat nur die Verknüpfungsbeziehungen auf diese Art und Weise verändert, indem das **MevisFile**-Objekt durchlaufen und verbundene **Branch**-Objekte erzeugt werden. Alle Attribute, die sich in der Mevis-Struktur finden, werden zunächst einfach übernommen. Es sind dies die **EdgeID**, die Koordinaten des Start- und Endknotens, das Volumen und auch die gesamte Voxelinformation, die die originäre Skelettlinie beschreibt. In einem späteren Transformationsschritt können letztere Informationen ausgewertet werden, um einen einzelnen Ast in kleine detaillierte Teilsegmente zu zerlegen.

Wie die drei Klassen **MevisFile**, **Lung** und **Branch** zusammenhängen, ist im UML-Klassendiagramm in Abb. 4.1 zu sehen. Die Mevis-Datei ist als lineares Feld von Knoten und Kanten abgelegt, deren Beziehungen durch Listen von IDs realisiert sind. Globale

Parameter wie die Voxelgrößen finden sich direkt im `MevisFile`-Objekt. Auch ein `Lung`-Objekt enthält diese globalen Werte und zusätzlich genau eine Referenz auf den Wurzel-`Branch`. Der Baum selbst ist durch lokale Verknüpfungen in jedem `Branch`-Objekt definiert; dort finden sich Referenzen auf den Elternast und maximal zwei Kinder. Jeder `Branch` stellt außerdem alle verfügbaren geometrischen Daten zur Verfügung. Auch indirekte Attribute, wie die Generationentiefe eines Astes, sind als Methoden der Klasse `Branch` realisiert.

Die Transformation der Graphenstruktur aus dem `MevisFile`-Objekt erfolgt zum Zeitpunkt der Objektkonstruktion von `Lung`. Hier kommt ein rekursiver Algorithmus zum Einsatz, der den Graphen nach den in Abschnitt 3.2.1.2 definierten Regeln analysiert und sukzessive neue `Branch`-Objekte erstellt.

4.1.4 Alternatives Binärformat

Die ASCII-Darstellung des Graphen hat den Vorteil, dass sie menschenlesbar ist. Es zeigt sich jedoch, dass sie in Bezug auf Einleseaufwand und Speicherbedarf keineswegs ideal ist. Ein Graph enthält eine Vielzahl gleichartig formatierter Knoten und Kanten, die immer die gleichen *Tags* zur Bezeichnung der Attribute aufführen. Dies führt zu einem nicht zu unterschätzenden Overhead. Auch in Bezug auf das Parsen der Datei ergeben sich Schwierigkeiten: dadurch, dass das Dateiformat von einer (aus Sicht dieser Arbeit) externen Stelle stammt, können Änderungen an der Anzahl und Abfolge der Attribute nicht ausgeschlossen werden. Auch beim späteren Export der in der hier entwickelten Applikation Binärbäume fehlen Angaben, etwa zu den Skelettvoxeln (siehe Abschnitt 3.2.1.2). Um flexibel auf solche Unterschiede reagieren zu können, muss beim Einlesen explizit nach jedem Schlüsselwort gesucht werden, und zwar wiederholt in jedem Knoten und jeder Kante. Durch eine exaktere Spezifikation der Grammatik des ASCII-Formats könnte man dem entgegenwirken. Um jedoch unabhängig zu bleiben und auch den Nachteil des Speicher-Overheads zu vermeiden, wurde der Weg gewählt, eine alternative Darstellung des Baumes in einer Binärdarstellung zu ermöglichen.

Eine Speicherung der in der Applikation dargestellten Bäume im Binärformat wird alternativ zum originären ASCII-Format angeboten. Die Vorteile in Bezug auf den Speicherbedarf seien in einer Überschlagsrechnung deutlich gemacht. Nimmt man exemplarisch das Attribut `Volume` einer Kante und nimmt an, dass das eigentliche Volumen mit sieben signifikanten Stellen angegeben ist, so ergibt sich im ASCII-Format folgendes Bild:

```
Volume: XXX.XXXX
```

Man benötigt demnach 16 Zeichen, also 16 Bytes, um das Volumen zu speichern. Eine binär abgelegte `float`-Variable mit einfacher Genauigkeit nimmt in der Regel lediglich vier Byte ein, man muss also in der ASCII-Darstellung einen Overhead von 300% in Kauf nehmen. Bei kleinen Integer-Werten wie der Ast-ID, die durch den String `'EdgeID: '` eingeleitet wird, ist der Platzverlust noch ausgeprägter.

Ein binäres Format verlangt eine genaue Spezifizierung des Formats; da dieses Format nur vom hier entwickelten Programm erzeugt wird, ergeben sich hier keine Probleme.

Typ	Größe[Bytes]	Beschreibung
string	11	Erkennungsstring 'LUNG-BINARY'
int16	2	Versionsnummer = 1
int32	4	Gesamtanzahl der Äste = n
Nun folgen n gleichartige Blöcke für jeden Ast:		
float	4	X-Koordinate des Startpunkts
float	4	Y-Koordinate des Startpunkts
float	4	Z-Koordinate des Startpunkts
float	4	X-Koordinate des Endpunkts
float	4	Y-Koordinate des Endpunkts
float	4	Z-Koordinate des Endpunkts
float	4	Proximale Querschnittsfläche in mm^2
float	4	Distale Querschnittsfläche in mm^2
byte	1	Lappen-Nr.
byte	1	Segment-Nr.
int32	4	Original-ID des Astes
int32	4	ID von Kind 1 (-1 wenn nicht vorhanden)
int32	4	ID von Kind 2 (-1 wenn nicht vorhanden)
Wenn beide Kind-IDs gleich -1, der Ast also ein Blatt ist:		
float	4	Vom Blatt versorgtes Volumen
float	4	Pulmonaler Druck am Blatt

Tabelle 4.1: Binärformat zur Speicherung der Lung-Struktur

Das Einlesen wird dadurch erheblich beschleunigt, da aufwendige Textvergleiche entfallen. Ein Schwachpunkt bildet lediglich der Punkt der Portabilität. Eine einfache binäre Ausgabe mit den von der C-Standardbibliothek gebotenen Mitteln resultiert in einem Format, welches von der Rechnerarchitektur abhängt (Größe von Datentypen, Byte-Ordnung). Es ist daher eine Transformation in ein unabhängiges Format notwendig, das sogenannte *Marshalling*. Erfreulicherweise bietet die Qt-Bibliothek [Her01] einen komfortablen Mechanismus für diesen Zweck. Die Plattformunabhängigkeit bleibt so gewahrt.

Die Binärdarstellung basiert auf der Baumstruktur im `Lung`-Objekt und lässt die ursprüngliche Graphenstruktur außer Acht. Es ist daher garantiert, dass eine derartige Datei einen korrekten binären Baum darstellt; der Einlesevorgang wird deutlich vereinfacht. Das Format ist wie in Tab. 4.1 gezeigt spezifiziert.

Man erkennt, dass keine Knoten und Kanten gespeichert werden, sondern, wie in der `Lung`-Klasse beschrieben, nur noch Ast-Objekte. Die Verknüpfungen zwischen den Ästen sind hier durch die beiden Integer-IDs realisiert; beim Einlesen werden daraus die entsprechenden Zeiger-Referenzen generiert. Im Kopf fehlen im Vergleich zum ASCII-Format einige Angaben: die Voxelgröße ist implizit als (1/1/1) definiert, die Koordinaten der Start- und Endpunkte sind also nicht mehr umzurechnen. Der Wurzelast ist immer der erste Ast in der Datei, daher ist die Angabe `RootID` nicht mehr

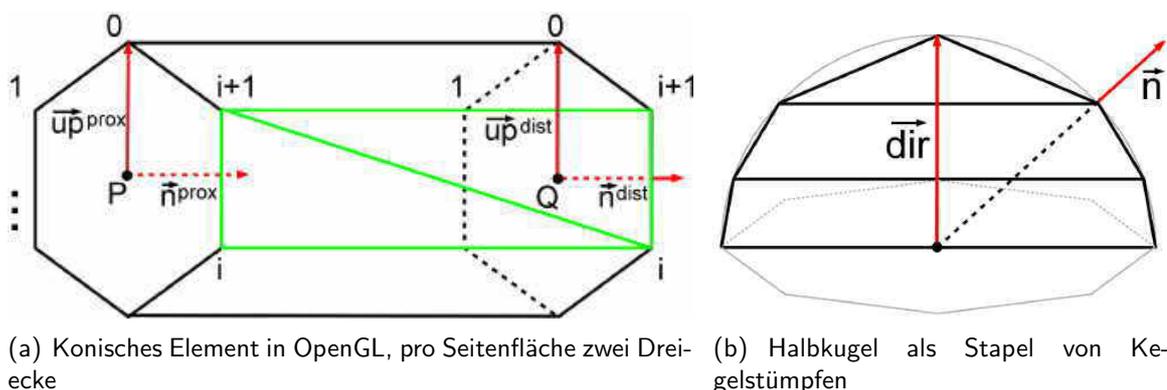


Abbildung 4.3: Modellierung der grafischen Grundelemente in OpenGL

notwendig. Weiterhin werden die Äste in der Reihenfolge ihrer IDs abgelegt, beginnend bei Null. Die ID jedes Astes muss daher nicht explizit angegeben werden. Zur Bedeutung Attribute „Lappen-/Segment-Nr.“, „pulmonaler Druck“, „versorgtes Volumen“ und „Original-ID“ siehe Abschnitt 3.4.1.

Durch die Beseitigung des ASCII-Overheads, die vereinfachte Modellierung der Kind-Beziehungen und die Beschränkung auf die in dieser Arbeit notwendigen Attribute kann der Speicherbedarf eines Baumes erheblich reduziert werden. Ein Bronchialbaum im erweiterten Modell auf Basis des Ausgussmodells von Abb. 2.1(a) mit insgesamt 18157 Ästen belegt im ASCII-Format 4,5 MByte; im hier vorgestellten Binärformat sind es lediglich 822 KByte. Zum Einlesen der Textdatei waren (unter MS Windows) ca. 20 Sekunden notwendig, die Binärdatei wurde in vier Sekunden geladen.

4.2 Rendering mit OpenGL

Zur Visualisierung der in Kapitel 3 entwickelten Baumstruktur, sowohl im Basis- als auch im erweiterten Modell, sind zwei grundlegende Elemente notwendig: schiefe Kegelstümpfe zur Darstellung einzelner Äste und sphärische Elemente an Bifurkationspunkten und Blättern. Die von der OpenGL-Hilfsbibliothek `glu` zur Verfügung gestellten Funktionen zur Darstellung ähnlicher Elemente sind für das entwickelte Konzept nicht flexibel genug. Vor allem können Kegelstümpfe nur mit parallelen Ebenen modelliert werden. Andere Arbeiten implementieren die Darstellung tubulärer Strukturen mithilfe der `gle`-Erweiterung von OpenGL (GL Tubing and Extrusion Library), zum Beispiel [HPSP01, Dom99]. Diese Bibliothek erlaubt das einfache Modellieren verbundener Kegelsegmente mit frei definierbaren Durchmessern entlang eines beliebigen Pfads. Allerdings lassen sich die notwendigen Anpassungen an den Kreisebenen, wie im Konzept beschrieben, nicht durchführen, und so versagt die Bibliothek für kurze Teiläste im erweiterten Modell. Auch ist eine Nutzung des OpenGL Selektions- und Rückkopplungsmodus zur Auswahl von Ästen unmöglich. Somit scheidet vorgefertigte Lösungen aus, und alle Elemente müssen manuell aus Dreiecken bzw. Dreiecksstreifen zusammengesetzt werden.

Die Berechnung der darzustellenden Elemente auf Basis eines Lung-Objekts ist ein relativ zeitaufwendiger Prozess, der mehrere Sekunden in Anspruch nehmen kann. Da die Baumstruktur starr ist und sich während der Visualisierung nicht mehr verändert, ist es von großem Vorteil, diese Berechnungen nur zu Beginn vorzunehmen und alle nötigen Vertices, Dreiecke und OpenGL-Befehle in vorkompilierten *Display-Listen* zu speichern. Diese Listen werden dann zur Darstellung ohne weitere Berechnungen abgerufen, und eine flüssige Darstellung wird ermöglicht. Der erhöhte Speicherbedarf der Liste, die sämtliche Vertices bereithalten muss, ist ein Nachteil, der in Kauf genommen werden muss.

Die Erzeugung der Display-Listen erfolgt in einem rekursiven Durchlauf durch die Baumstruktur. Dabei wird pro Rekursionsschritt ein kompletter Bronchus berechnet, der sich im erweiterten Modell aus einer Vielzahl von Ästen von einer Bifurkation bis zur nächsten zusammensetzen kann. Jeder (Teil-)Ast wird durch einen schiefen Kegelstumpf dargestellt, wobei zwei benachbarte Äste jeweils eine Fläche gemeinsam haben. Dementsprechend werden die Äste wie folgt erzeugt:

1. Ermittle Position, Normalenvektor und einen beliebigen Up-Vektor der ersten proximalen Kreisfläche wie in Kapitel 3 dargelegt. Berechne daraus die erste Punktmenge `points_prox[]`.
2. Ermittle Position, Normalenvektor und Up-Vektor der distalen Kreisfläche.
3. Berechne daraus die Punktmenge `points_dist[]`, welche die Kreisfläche als N -seitiges regelmäßiges Polygon darstellt.
4. Die proximale Punktmenge `points_prox[]` ist bereits bekannt; erstelle den Konus durch Verbindung der folgenden Punkte zu jeweils einem Dreieck (vgl. Abb. 4.3(a)):

```
points_prox[i], points_dist[i] points_prox[(i+1)%(N-1)]
points_prox[(i+1)%(N-1)], points_dist[i], points_dist[(i+1)%(N-1)]1
i=0,1,...N-1
```

Zusätzlich muss zu jedem Vertex ein Normalenvektor angegeben werden, der für die Berechnung der Beleuchtung verwendet wird. Dieser wird definiert als der Richtungsvektor vom Mittelpunkt der Kreisfläche (Punkte P bzw. Q in Abb. 4.3(a)) zum jeweiligen Dreieckspunkt².

Jeder Konus wird in eine eigene Displayliste kompiliert, um ihn später separat ansteuern zu können (Selektion und Färbung).

5. Fahre mit dem nächsten Teilast bei 2. fort, wobei die bisherige distale Punktmenge zur neuen proximalen wird.

¹Aufeinander folgende Dreiecke haben jeweils zwei Punkte gemeinsam; sie werden als Dreiecksstreifen (`TRIANGLE_STRIP`) spezifiziert, sodass die Punkte nicht wiederholt angegeben werden müssen.

²Diese Definition ist bei nicht-parallelen Kreisebenen nicht vollständig korrekt; merkliche Auswirkungen sind nicht zu beobachten.

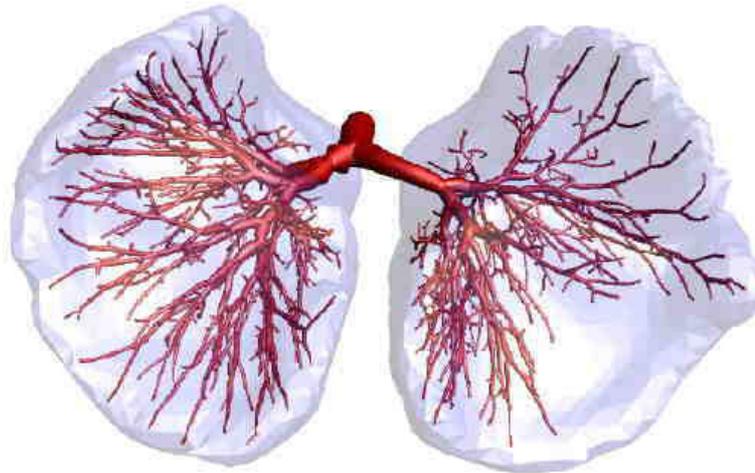


Abbildung 4.4: Bronchialbaum mit umhüllenden Parenchym

An jeder Bifurkation und jedem Blatt wird zusätzlich eine Halbkugel modelliert, um einen glatten Abschluss zu gewährleisten. Der Äquator dieser Halbkugel wird definiert durch die zuletzt berechnete distale Punktmenge. In Richtung des letzten Astvektors wird nun ein Stapel von Kegelstümpfen berechnet, dessen letztes Element schließlich zu einer Pyramide wird (siehe Abb. 4.3(b)). Es zeigt sich, dass bei schattierter und beleuchteter Darstellung fünf Kegelsegmente ausreichen, um die Kugel zu approximieren. Die Normalenvektoren zu jedem Punkt auf der Oberfläche der angenäherten Kugel sind definiert als die Richtungsvektoren vom Kugelmittelpunkt zum Oberflächenpunkt.

4.2.1 Visualisierung des Lungenparenchyms

Die Oberfläche des sogenannten Lungenparenchyms definiert das gesamte vom Bronchialbaum versorgte Volumen. Diese „Einhüllende“ des Baums ist nicht Teil der bisher beschriebenen Datenstrukturen, sie kann jedoch auf Basis der originalen Volumendaten rekonstruiert werden. Diese Konstruktion ist nicht Bestandteil dieser Arbeit; eine optionale Darstellung des Parenchyms kann jedoch den optischen Eindruck deutlich verbessern. Daher wurde die Möglichkeit geschaffen, das zu einem Baum gehörende Parenchym als eigenständige Textdatei einzulesen und semi-transparent darzustellen. Die Datei enthält unmittelbar die Vertices und Normalenvektoren der Dreiecke, welche die Oberflächenstruktur beschreiben. Diese Informationen können eingelesen und direkt an die entsprechenden OpenGL-Routinen übergeben werden.

Abb. 4.4 zeigt einen Datensatz mit umhüllendem Parenchym. Um Speicher- und Rechenzeitaufwand nicht ungebührlich zu erhöhen, sollte das Parenchym mit einer mäßigen Anzahl von Dreiecken modelliert sein. Die abgebildete Hülle besteht aus 12.340 Dreiecken. Zur transparenten Darstellung bietet OpenGL die Möglichkeit, jedem Punkt einen „Transparenzwert“ zuzuweisen. Die Darstellung kann durch Nutzung des Tiefenpuffers (Depth-Buffer, z-Buffer) wie folgt realisiert werden:

- Zeichne zuerst alle nicht-transparenten Objekte, das heißt die gesamte Baumstruktur, mit voll aktiviertem Tiefenpuffer.
- Zeichne nun die Umhüllende, etwa mit einem Transparenzwert von 0,4, wobei auf den Tiefenpuffer nur lesend zugegriffen wird. Dies verhindert einerseits, dass transparente Teile hinter nicht-transparenten Objekten dargestellt werden, andererseits werden alle transparenten Objekte unabhängig von ihrer Reihenfolge dargestellt.

Nach Zeichnen der Baumstruktur wird für jeden projizierten Punkt³ der Umhüllenden getestet, ob er von dem bereits dargestellten Baum überdeckt wird. Ist dies nicht der Fall, so errechnet sich der Farbwert an der entsprechenden Stelle als „Mischung“ aus der Farbe der Umhüllenden und der bereits vorhandenen Farbe an der Stelle (Hintergrundfarbe oder Astfarbe). Die Mischung ist ein multiplikativer Prozess unter Einbeziehung des Transparenzwertes (vgl. [WND97]).

4.2.2 Detailgrad der Darstellung

Die Gesamtzahl der Dreiecke bzw. Vertices in einer Szene ist das wichtigste Kriterium für die Geschwindigkeit der Darstellung. Da die visualisierten Baumstrukturen recht komplex sein können, ist es wichtig, Qualität und Geschwindigkeit der Darstellung beeinflussen zu können. Der einzige veränderliche Parameter ist, wie viele Eckpunkte pro Polygon für die Darstellung der Kegelstümpfe (und der Halbkugeln) verwendet werden. Genau dies ist vom Benutzer einstellbar, und zwar mittels zwei Maßzahlen:

- Die Anzahl der Polygonpunkte für den Wurzelast ($N_0 \in \mathbb{N}, N_0 \geq 4$).
- Die Abnahme der Polygonpunkte pro Teilungsgeneration ($dec \in \mathbb{R}, dec \geq 0$).

Für die Polygonpunkte N_k eines Astes auf Generation k gilt nun

$$N_k = \max(4, \lfloor N_0 - k \cdot dec \rfloor)$$

Damit erreicht man mit einfachen Mitteln eine Reduzierung des Darstellungsaufwands für kleinere Äste (deren Anzahl hoch ist) im Vergleich zu den wenigen dicken Hauptästen. Es ist anzumerken, dass diese Detailreduzierung statisch ist; verkleinert der Betrachter interaktiv den Kameraabstand zu einem kleinen Ast, ändert sich die Anzahl der verwendeten Punkte nicht. In anderen Anwendungsbereichen, wie beispielsweise bei Simulatoren, ist die dynamische Anpassung des „Level Of Detail“ (LOD) ein wichtiger Qualitätsfaktor. In dieser Arbeit genügt ein statisches Modell, das komfortabel mit den zwei beschriebenen Maßzahlen parametrisiert werden kann. Eine quantitative Betrachtung der Darstellungsgeschwindigkeit findet sich in Abschnitt 5.3.

³Genauer: jedes OpenGL-Fragment

4.3 GUI-Programmierung mit dem Qt-Toolkit

Die C++-Klassenbibliothek Qt, die von der Firma Trolltech [Tro] entwickelt wurde, bietet einen geschlossenen Satz von Klassen zur Programmierung von grafischen Benutzerschnittstellen sowie allgemeine Container und Algorithmen an. Sie ist sowohl für MS Windows als auch für Linux/UNIX verfügbar. Auf letzterer Plattform sind alle Versionen bis zur aktuellen (3.2) frei im Quelltext verfügbar; unter Windows existiert lediglich die Version 2.3 als nichtkommerzielle Variante, die ohne Abgabe von Lizenzgebühren für frei verfügbare Software eingesetzt werden darf. Aus diesem Grund basiert die hier entwickelte Applikation auf dieser Version, die uneingeschränkte Funktionalität bietet (lediglich ein Hinweis „Freeware“ wird der Titelleiste jeder Applikation hinzugefügt).

Es existieren andere, native APIs (Application Programming Interfaces), die Schnittstellen zum jeweiligen Fenstersystem anbieten, beispielsweise die MFC-Bibliothek (MS Foundation Classes) für das MS Windows Betriebssystem oder OSF/Motif für das X-Window System. Im exemplarischen Vergleich zu diesen beiden bietet Qt eine wesentlich kompaktere, einfachere Schnittstelle, eher im Sinne von Java/AWT. Der essentielle Vorteil liegt jedoch in der Plattformunabhängigkeit der entwickelten Applikationen: sowohl das GUI wie auch die Programmlogik inklusive der externen Datenformate (vgl. 4.1.4) sind ohne weitere Änderungen auf beiden Plattformen kompilier- und lauffähig.

Die entwickelte Applikation besteht aus einem Hauptfenster, in das die benötigten Elemente (Widgets) eingebunden werden. Es entsteht eine Klassenhierarchie, wie sie in Abb. 4.5 dargestellt ist. Dabei steht das von der Klasse `QGLWidget` abgeleitete Element `GLTree` im Vordergrund, das für die Darstellung und Interaktion mit der Baumstruktur zuständig ist. Qt integriert die OpenGL-Schnittstelle und sorgt mit dem genannten Widget für eine einfache Integration eines OpenGL-Darstellungskontexts in das GUI. Abbildung 4.6 zeigt einen Screenshot der Applikation⁴.

4.4 Interaktion

Die Interaktion mit der dreidimensionalen Baumstruktur beinhaltet zwei Punkte: die Navigation im Raum, um den Baum aus beliebigen Blickwinkeln betrachten zu können, und die Auswahl von Teilmengen der Äste. Letzteres ist insbesondere für die statistische Auswertung interessant, aber auch, um zur besseren Übersicht Äste selektiv ausblenden zu können. Das Eingabegerät der Wahl für diese Bedürfnisse ist eine Standardmaus mit zwei Tasten. Es existieren auch spezialisierte Eingabegeräte, zum Beispiel sogenannte „Spaceballs“ für die dreidimensionale Navigation, die mehr Freiheitsgrade besitzen als eine Maus. Allerdings wird im Rahmen dieser Arbeit auf eine Unterstützung dieser

⁴Man beachte die Durchmesserchwankungen in der Trachea des abgebildeten Bronchialbaums. Diese entstehen durch die Knorpelspangen, die die Luftröhre umgeben (vgl. auch Abb.2.4(a)). Die Durchmesser des rechten Hauptbronchus (links im Bild) sind hingegen nicht realitätsgetreu modelliert; dieser Fehler entstand jedoch bereits bei der Erstellung der Graphenstruktur nach den Methoden von [Sel99] und wäre nur durch manuellen Eingriff zu korrigieren. Eine solche Möglichkeit wird in dieser Arbeit nicht realisiert (vgl. Kapitel 5).

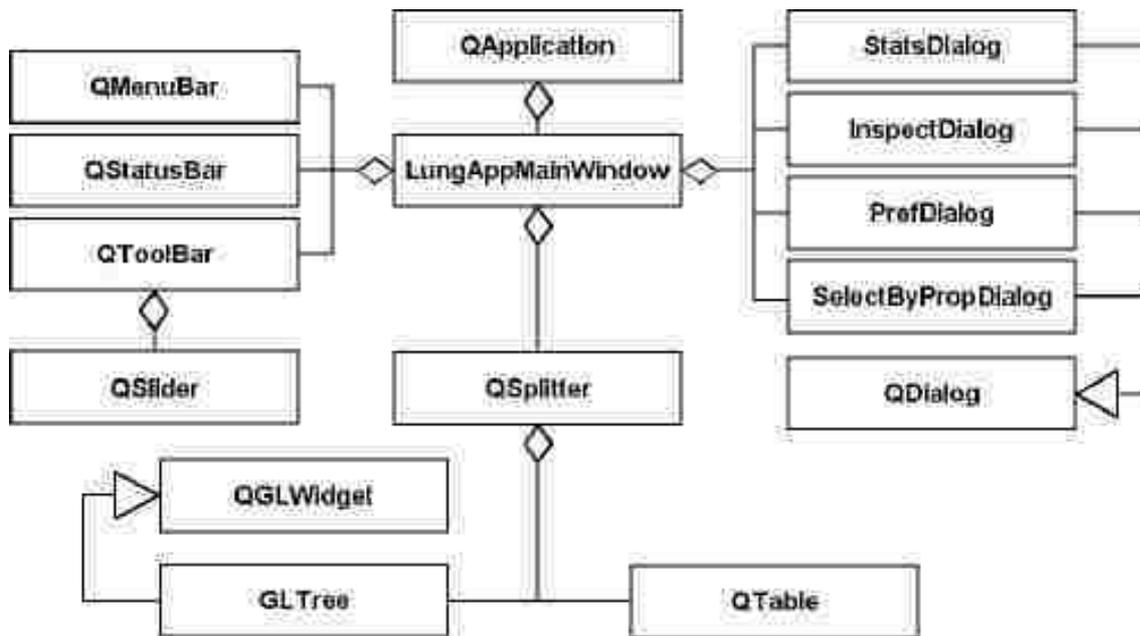


Abbildung 4.5: Klassendiagramm zur GUI

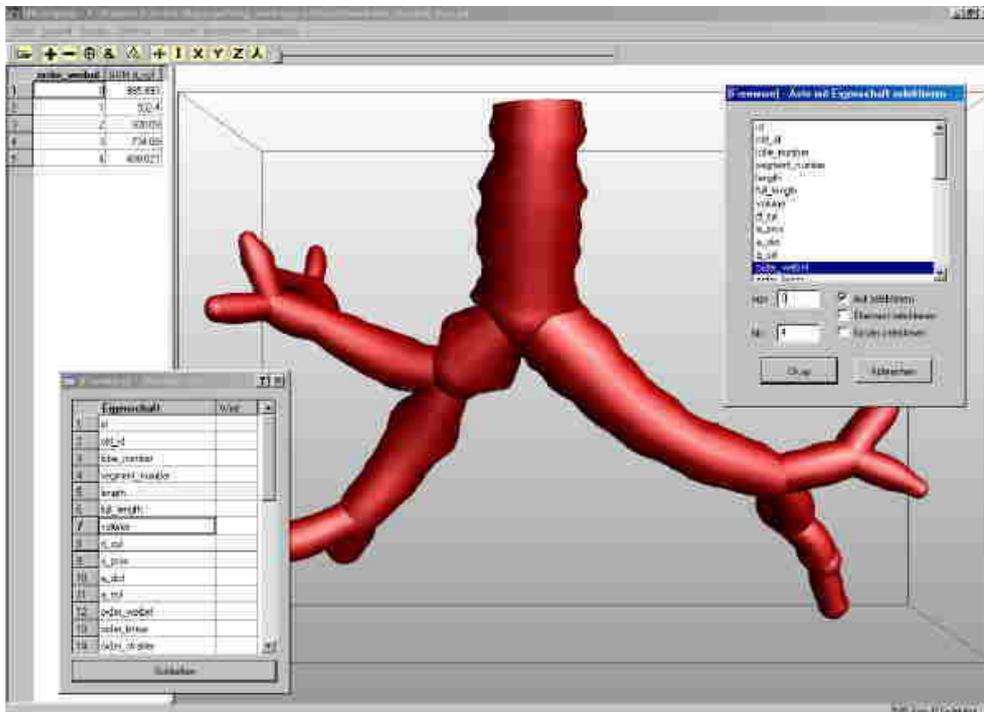


Abbildung 4.6: Screenshot der Applikation, Hauptfenster mit zwei Benutzer-Dialogen

Geräte verzichtet, da eine plattformunabhängige Unterstützung durch das Qt-Toolkit nicht direkt gegeben ist und alternative Treiberintegration den Rahmen des Projekts sprengen würde.

4.4.1 Navigation im Raum

Um dem Benutzer die Interaktion mit der Baumstruktur mithilfe einer herkömmlichen Computermaus zu ermöglichen, werden fünf verschiedene Bewegungsmodi zur Verfügung gestellt. In jedem dieser Modi wird ein unterschiedlicher Betrachtungsparameter durch Mausbewegung bei gedrückter Maustaste verändert. Es sind dies

- Translation
- Zoomen
- Rotation um die x/y/z-Achse des Objekts

Die Blickrichtung des Betrachters liegt fest in Richtung der negativen z-Achse. Die Bewegung des Betrachters ist im Grunde eine Bewegung aller dargestellten Objekte. Im Modus *Translation* verändert sich durch Mausbewegung in beide Richtungen die x- und y-Position des Betrachters; die z-Position wird durch *Zoomen* beeinflusst. Die in *Rotation* einzustellende Drehung des Baumes erfolgt derart, dass der aktuelle Baum um seinen Mittelpunkt rotiert wird. Dieses ist der Mittelpunkt der *Bounding Box* des Baumes, also des kleinsten quaderförmigen Bereiches parallel zu den Koordinatenachsen, der den gesamten Baum enthält. Die Bounding Box ist in verschiedenen Abbildungen wie zum Beispiel Abb. 4.8 zu erkennen. Die Rotationsachsen sind diejenigen eines Koordinatensystems, das mit dem Objekt rotiert wird; somit entspricht beispielsweise eine Rotation um die z-Achse immer einer Drehung um die gedachte „Körperachse“ der Lunge, wie sie im Menschen orientiert ist, unabhängig von bereits erfolgten Drehungen um andere Achsen. In anderen 3D-Viewern wird um das fixe Weltkoordinatensystem rotiert; eine Drehung um das Objektkoordinatensystem ist intuitiver.

4.4.2 Selektion von Ästen

Eine Möglichkeit zur Auswahl einzelner Segmente des Baumes ist, es dem Benutzer zu ermöglichen, sie direkt mit der Maus anzuklicken. Das Betätigen einer Maustaste im Grafikfenster löst ein Ereignis aus, das die zweidimensionalen Fensterkoordinaten der Mausposition enthält. Es gilt nun, dasjenige dreidimensionale Objekt zu identifizieren, das an dieser Stelle des Fensters sichtbar ist. Dies ist keine triviale Aufgabe, da man es ja mit einer perspektivischen Projektion des 3D-Objekts auf das Darstellungsfenster zu tun hat. Eine eindeutige Zuordnung eines Projektionspunktes zu einem Punkt im dreidimensionalen Raum ist nicht mehr möglich, vielmehr werden alle Punkte auf einer Raumgeraden auf diesen einen Projektionspunkt abgebildet. Das dem Betrachter naheliegende Objekt, das auf dieser Geraden liegt, ist sichtbar, da es die dahinterliegenden verdeckt. Dieses soll demnach ausgewählt werden. In der Praxis wird man nicht

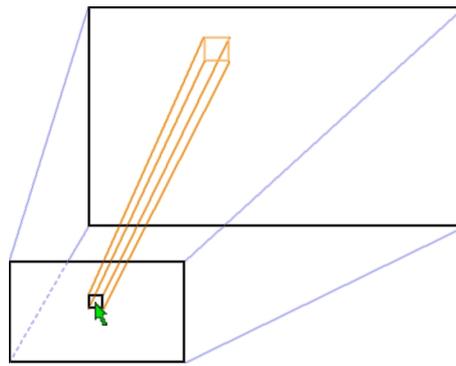


Abbildung 4.7: Selektionsproblematik bei perspektivischer Projektion. Der blau umgrenzte Pyramidenstumpf ist das gesamte dargestellte Raumvolumen. Bei einem Klick auf die angegedeutete Position in der 2D-Projektion muss der orange begrenzte Bereich auf Objekte getestet werden.

genau einen Punkt (Pixel) wählen, sondern einen kleinen quadratischen Bereich um die Mausposition herum, um eine gewisse Toleranz bei der Objektauswahl zu erlauben. Die zu untersuchende Raumgerade wird dann zu einem Volumen, das bei perspektivischer Projektion ein Pyramidenstumpf ist. Abb. 4.7 versucht, dies zu verdeutlichen; für Details zu verschiedenen Projektionen ist die Lektüre des OpenGL-Programming Guide [WND97] zu empfehlen.

Glücklicherweise bietet OpenGL mit dem sogenannten Selektions- und Rückkopplungsmodus einen komfortable Weg an, die angerissenen Operationen durchzuführen. Sobald der Benutzer eine Position anklickt, wird dieser Modus aktiviert und die Szene erneut gerendert. Dabei spezifiziert man zunächst unverändert die aktuelle Projektion (inkl. Position, Blickwinkel) und lässt dann eine Hilfsfunktion, der die Mausposition und eine Toleranz übergeben wird, das darzustellende Volumen genau auf den beschriebenen eng begrenzten Pyramidenstumpf begrenzen. Diese einmalige Darstellung der Szene wird nicht auf dem Bildschirm ausgegeben. Vielmehr wird analysiert, welche Objekte dargestellt *würden*. Das Ergebnis ist eine Liste von Objektname und deren Entfernung zum Betrachter. Die Objektname entsprechen im Falle dieser Arbeit den IDs der Äste. Es genügt nun, das nächstliegende Objekt zu suchen, und der ausgewählte Ast ist bekannt.

Ein selektierter Ast wird sinnvollerweise in einer anderen Farbe dargestellt als die übrigen. Zur Verwaltung der Auswahlinformationen wurde der Weg gewählt, in der `Lung`-Klasse ein Feld von booleschen Werten anzulegen, sodass der Eintrag `SelectedBranches[i]` für den Ast mit der ID `i` angibt, ob er selektiert ist oder nicht. Da auch die Darstellung der Äste sowie die statistische Auswertung auf dieser nach ID geordneten linearen Darstellung beruht, ist dies die beste Möglichkeit.

Neben der Menge von *selektierten* Ästen gibt es noch genau einen *aktuellen* Ast, der mit einer anderen Maustaste gewählt wird. Dieser dient zwei verschiedenen Zwecken: einerseits können bei Bedarf detaillierte textuelle Informationen über ihn in einem separaten

s_1	s_2	Additiv $s_1 \vee s_2$	Subtraktiv $s_1 \wedge \neg s_2$	XOR $s_1 \oplus s_2$	UND $s_1 \wedge s_2$
0	0	0	0	0	0
0	1	1	0	1	0
1	0	1	1	1	0
1	1	1	0	0	1

Tabelle 4.2: Semantik der Selektionsmodi

Fenster angezeigt werden; andererseits lassen sich zwei weitere Selektionsmöglichkeiten mit ihm realisieren: der Befehl „Teilbaum ab aktuellem Ast“ selektiert den gesamten Unterbaum unter dem aktuellen Ast. Gleichsam lassen sich alle Äste, die auf dem Pfad von der Trachea bis zum aktuellen Ast liegen, selektieren. Realisiert wird dieses Verhalten durch die **Lung**-Klasse, die die Baumstruktur beginnend beim aktuellen Ast rekursiv durchläuft und alle Kinder respektive alle Elternäste selektiert. Eine letzte Möglichkeit zur Auswahl einer Teilmenge von Ästen ist die Spezifizierung einer Eigenschaft, die die Äste erfüllen sollen. Dies erfolgt durch Angabe eines Intervalls „von-bis“ und der gewünschten Eigenschaft. So lassen sich beispielsweise alle Äste selektieren, deren Durchmesser zwischen fünf und zehn Millimetern liegt. Alle definierten Eigenschaften, die für die Statistik nutzbar sind, können auch hier verwendet werden (vgl. Abschnitt 3.4.1).

Die drei genannten Alternativen bieten bereits einen sinnvollen Weg zur Auswahl von Ästen. Um die Flexibilität noch zu erweitern, ist es angebracht, Verknüpfungen verschiedener Auswahlen zuzulassen, sodass beispielsweise zuerst ein Teilbaum gewählt wird, etwa ein Lungensegment, und danach die Auswahl noch mit dem Durchmesserkriterium eingeschränkt wird. Die gewählte Lösung besteht in der Implementierung verschiedener *Selektionsmodi*. In den Modi „Additiv“ und „Subtraktiv“ werden der Auswahl Äste hinzugefügt beziehungsweise bereits selektierte Äste entfernt. Daneben existieren die Modi „XOR“ und „UND“, die die alte und neue Selektion mit den entsprechenden logischen Operatoren verknüpfen. Auch das Hinzufügen und Entfernen von Ästen kann als logische Verknüpfung angesehen werden. Beim Hinzufügen ist es der ODER-Operator: ein Ast ist selektiert, wenn er es bereits vorher war oder durch die neue Auswahl gewählt wurde. Tabelle 4.2 gibt einen Überblick über die vier Modi.

Dabei sei s_1 der alte Zustand eines Astes und s_2 der Zustand, der sich durch die neue Selektionsaktion ergeben würde. Die Verknüpfung ergibt den tatsächlichen Endzustand des Astes. In Abb. 4.9 ist ein schematisches Beispiel für einige sukzessive Selektionen zu sehen. Die Zahlen an den Ästen sind deren IDs, in Klammern steht die Weibel-Generation des Astes.

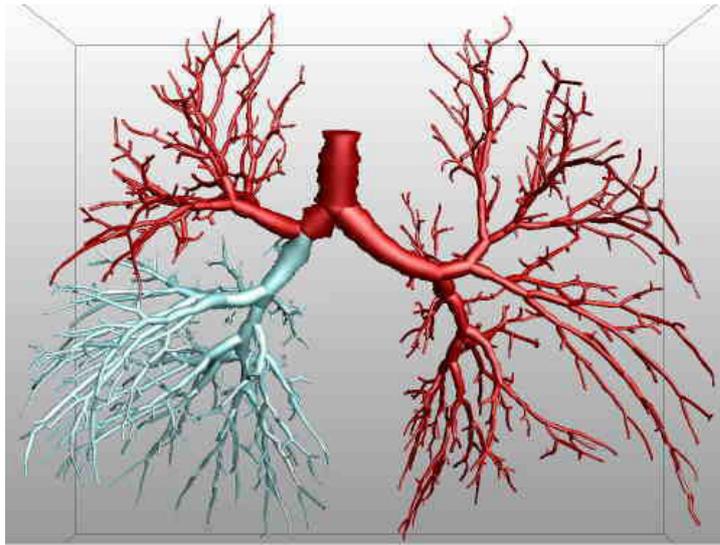


Abbildung 4.8: Screenshot der Applikation: rechter unterer Lungenlappen selektiert

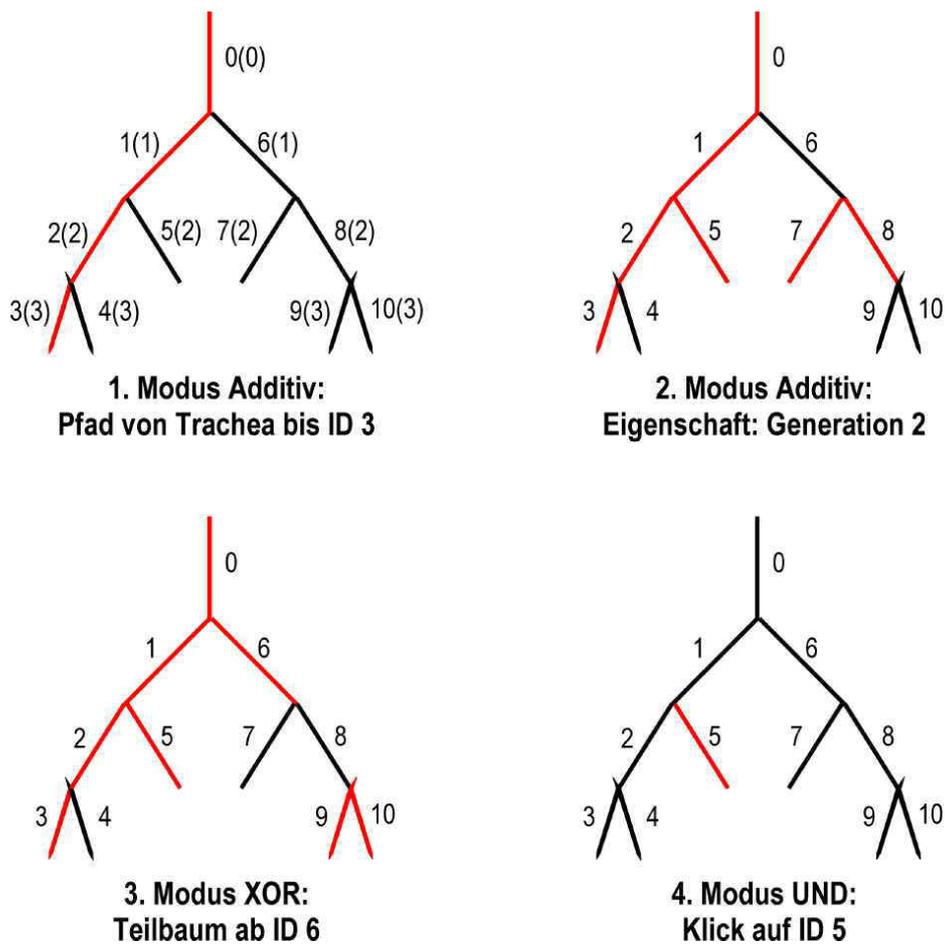


Abbildung 4.9: Beispielhafte Verknüpfungen von Astselektionen

4.5 Statistik

Die Implementierung der statistischen Erfassung morphologischer Parameter baut direkt auf der Selektionsmöglichkeit auf – alle weiteren Auswertungen beziehen sich immer nur auf die Teilmenge von Ästen, die der Benutzer gewählt hat. Die Anforderungen an die Applikation in Bezug auf die Datenerfassung wurden bereits in Abschnitt 3.4 definiert. Dem Benutzer wird diese Funktionalität in einem zentralen Dialogfenster zur Verfügung gestellt, das in Abb. 4.10(a) zu sehen ist. Aus einer Liste können alle gewünschten Asteigenschaften ausgewählt werden. Die beschriebene Gruppierung nach einem Attribut ist optional wählbar durch Angabe genau einer Gruppierungseigenschaft, der Intervallbreite und der Art der Zusammenfassung (Mittelwert oder Summe). Nach Bestätigung all dieser Angaben erzeugt das Programm automatisch eine tabellarische Ausgabe im Hauptfenster der Applikation, die in Abb. 4.10(b) illustriert ist. Dabei ist die `Lung`-Klasse verantwortlich für das Erfassen der Daten. Eine entsprechende Methode nimmt die gewählten Parameter entgegen und liefert eine Tabellenstruktur, die dem `QTable`-Widget des GUI (vgl. Abb. 4.5) übergeben wird. Die Details dieser Datenstrukturen sind nicht relevant; einige Punkte bezüglich der Asteigenschaften bedürfen jedoch einer Erklärung:

Binäre Eigenschaften

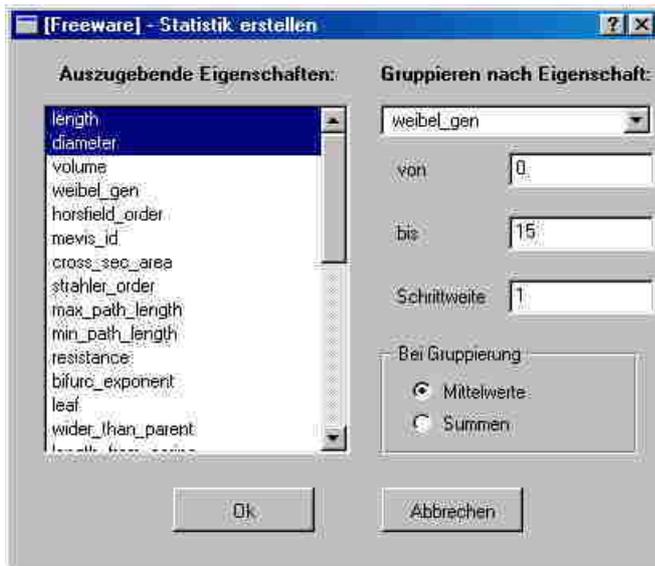
Alle Asteigenschaften liefern positive Fließkommazahlen. Auch Attribute wie „ist-Blatt“, die eine Ja/Nein-Rückgabe besitzen, verletzen dieses Schema nicht; vielmehr wird der Wert 1.0 für Ja, 0.0 für Nein zurückgegeben.

Ungültige Eigenschaften

Gewisse Eigenschaften sind nicht für jeden Ast definiert. Beispielsweise ist der Bifurkationsexponent nur für Äste mit genau zwei Töchtern definiert, deren Durchmesser kleiner ist als der des Elternastes. Ähnliches gilt für die Flächenasymmetrie. In dieser Arbeit wird der Rückgabewert -1.0 für ungültige Eigenschaften definiert. Dies ist glücklicherweise problemlos machbar, da alle Attribute positive Werte haben und so keine Kollisionen entstehen können. Ungültige Werte entstehen auch dann, wenn bei Gruppierung einem Intervall kein Ast zugeordnet wird, wenn beispielsweise bei der Gruppierung nach Durchmessern in den Intervallen $\{[0; 5), [5; 10), [10; 15)\}$ keine Äste vorhanden sind, deren Durchmesser zwischen fünf und zehn Millimetern liegt.

Kapselung der Eigenschafts-Methoden

Jeder definierten Eigenschaft entspricht eine Methode der Klasse `Branch`, etwa `getDiameter()` für den Durchmesser oder `isLeaf()` für die Angabe, ob der Ast ein Blatt ist. Eine Liste aller verfügbaren Eigenschaften muss an verschiedenen Stellen in der Applikation zugänglich sein, vor allem in den beiden Listenfeldern des Statistik-Dialogs. Dort müssen dem Nutzer lesbare Zeichenketten angeboten werden, und die ausgewählten Listeneinträge müssen in der Applikation den entsprechenden `Branch`-Methoden zugeordnet werden. Um diese Schnittstelle flexibel zu halten und eine möglichst einfache programmtechnische Erweiterung zu ermöglichen, wurde eine weitere Indirektion eingeführt: die allgemeine Methode `Branch::getProperty(string prop)`. Diese hat



(a) Dialogfenster für die statistische Auswertung

	weibel_gen	MW length	MW diameter
1	0	29.9091	18.3403
2	1	29.87	11.7382
3	2	18.8266	7.97168
4	3	12.4571	6.07774
5	4	8.88547	4.46971
6	5	10.2125	3.32383
7	6	9.30644	2.50378
8	7	8.26785	2.0285
9	8	7.98058	1.71699
10	9	6.95959	1.53302
11	10	6.98288	1.44966
12	11	7.05248	1.38414
13	12	6.9116	1.40852
14	13	8.06639	1.32612
15	14	6.06188	1.26995
16	15	10.0007	1.26327

(b) Ergebnistabelle

Abbildung 4.10: Programmelemente zur Erfassung statistischer Größen

keine andere Funktion als die übergebene Zeichenkette (z.B. 'diameter') zu überprüfen und das Ergebnis des entsprechenden Methodenaufrufs zurückzugeben. Diese String-Vergleiche führen mithin zu einem gewissen Overhead bei jeder Eigenschaftsabfrage, jedoch bietet die Methode eine flexible zentrale Schnittstelle zu allen Attributen. Eine global verfügbare Liste aller definierten Eigenschafts-Zeichenketten kann nun ohne weiteres für das Erstellen der Dialogelemente herangezogen werden. Neu hinzu kommende Eigenschaften werden integriert, indem lediglich in dieser Liste ein neuer Eintrag definiert und `getProperty()` um diesen erweitert wird. Änderungen an der Statistik-Funktion oder dem GUI sind nicht notwendig.

4.6 Schnittstelle zum Simulationsmodul

Die numerische Simulation operiert auf der selben Datenstruktur des Baumes wie die hier vorgestellte Visualisierung. Alle Erweiterungen des Modells wurden mit der Maßgabe spezifiziert, diese Kompatibilität nicht zu brechen. Daher gehört es zu den Aufgaben der Applikation, Simulationsergebnisse zu visualisieren. Dies können Konzentrationsangaben für jeden Ast über die Zeit oder auch Geschwindigkeitsverteilungen sein. Das Simulationsprogramm exportiert eine Vielzahl von Ergebnissen im Textformat. Für die Visualisierung sind jene Ergebnisse relevant, die jedem Ast (identifiziert durch seine ID) zu einem oder mehreren diskreten Zeitpunkten einen numerischen Wert zuordnen. Diese Werte sollen farbkodiert dargestellt werden, wie in Abschnitt 3.3 beschrieben. Die erzeugte Textdatei hat nun folgendes Format:

```
ResultsByBranch
Branches: 35
```

```

Acini: 18
GenerationsAc: 7
TimeStep: 0.2
Range: 2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 [...]
0.2 0.174672 0.111163 0.0942413 0.072655 0.0496471 0.0590254 [...]
0.4 0.20894 0.206259 0.205391 0.204059 0.20174 0.202879 [...]
0.6 0.209993 0.209975 0.209965 0.209928 0.209615 0.209734 [...]
0.8 0.209999 0.209997 0.209992 0.209967 0.209714 0.209798 [...]
1.0 0.209999 0.209997 0.209991 0.209966 0.209713 0.209785 [...]
1.2 0.209999 0.209997 0.209989 0.20996 0.20967 0.20974 [...]
1.4 0.209999 0.209996 0.209985 0.209946 0.20957 0.209649 [...]
1.6 0.209999 0.209995 0.209978 0.209919 0.20935 0.209457 [...]
1.8 0.209998 0.209991 0.20996 0.209848 0.208733 0.208911 [...]
2.0 0.209997 0.209986 0.209913 0.209626 0.204749 0.204306 [...]

```

Im Kopf steht nach einem Erkennungsstring (`ResultsByBranch`) die Anzahl der Äste. Diese muss identisch sein mit der Astanzahl des visualisierten Baums – eine weitere Überprüfung, ob die Simulationsergebnisse tatsächlich auf dem visualisierten Baum beruhen, gibt es nicht. Denkbar wäre es, einen einzigartigen Identifizierungscode für jeden Baum zu generieren und in allen Dateien anzugeben; bisher obliegt eine korrekte Nutzung jedoch dem Benutzer. Die Angaben `Acini` und `GenerationsAc` beziehen sich auf die respiratorischen Einheiten distal zu den Blättern des Baumes, die in der Simulation ebenfalls betrachtet werden, für die Visualisierung zurzeit aber irrelevant sind.

Die eigentlichen Daten liegen in Form einer umfangreichen Matrix vor, die in den Spalten die IDs der Äste (beginnend bei Null) und in den Zeilen die diskreten Zeitschritte speichert. Die Zeitzählung beginnt bei Null Zeiteinheiten und geht in Intervallen der Größe `TimeStep` bis `Range` Zeiteinheiten vor. Die aktuelle Zeit wird zu Beginn jeder Zeile noch einmal angegeben, dann folgen jeweils `Branches` relevante Ergebniswerte. Danach stehen Daten zu den erwähnten respiratorischen Einheiten.

Das oben angegebene Beispiel zeigt einen sehr kleinen Baum mit 35 Ästen und einer groben Zeitdiskretisierung von 0,2 Zeiteinheiten (sprich Sekunden) im Bereich von 0-2 Sekunden. Demnach sind $35 \cdot 11 = 385$ Fließkommazahlen zu speichern. Üblich sind größere Bäume mit > 1000 Ästen. Für eine ansprechende Animation des Atemvorgangs sind überdies weit mehr Zeitschritte über mehrere Atemzyklen (die üblicherweise vier Sekunden dauern) notwendig, etwa 0,005 sec Intervalle für insgesamt 16 sec Atemzeit. Somit wächst das Datenaufkommen auf eine Größenordnung von mehreren MByte. Trotzdem werden die Daten jeweils komplett eingelesen und intern in einem zweidimensionalen Feld gespeichert; ein hinreichend großer (virtueller) Arbeitsspeicher wird für die Verarbeitung großer Datenmengen vorausgesetzt.

Interaktionsmöglichkeiten

Dem Benutzer steht über einen Menüeintrag die Möglichkeit offen, eine Textdatei im oben beschriebenen Format einzulesen. Die grafische Darstellung eines geladenen Baumes ist nun in zwei Modi möglich: im herkömmlichen Modus sind Astselektionen durch farbliche Hervorhebung sichtbar, im neuen Modus werden die Äste entsprechend den eingelesenen numerischen Daten eingefärbt. Wahlweise kann eine textuelle Darstellung

des aktuell visualisierten Zeitpunkts eingeblendet werden. Mittels eines Schiebereglers, der vom Qt-Toolkit zur Verfügung gestellt wird (`QSlider`), kann der aktuelle Zeitschritt gewählt werden.

Weiterhin gibt es die Option, sämtliche Zeitschritte in Folge als Bilddateien auf den Datenträger zu schreiben. Hierfür steht nur das Format „Windows-BMP“ zur Verfügung. Die Bilddateien enthalten einen vom Benutzer vorgegebenen Namen mit den Suffixes „0001.BMP“, „0002.BMP“ usw.. In einem externen Programm können aus diesen Einzelbildern Animationen hergestellt werden, die einen guten visuellen Eindruck des Einwaschvorgangs von eingeatmenten Gasen und der Ausbildung eines dynamischen Gleichgewichtes liefern.

Neben einer visuellen Darstellung ist eine quantitative Betrachtung der Konzentrationsverteilungen von großer Bedeutung. Hierzu könnten die bei der Simulation erzeugten Textdateien direkt in einem Statistikprogramm weiterverarbeitet werden. Die Fülle der Daten – Konzentrationswerte für alle Zeitschritte über sämtliche Äste des Baumes – macht allerdings eine zielgenaue Analyse schwierig. Beispielsweise kann es erforderlich sein, die Konzentrationsverteilung auf bestimmten Pfaden von der Wurzel des Baumes zu Blättern in verschiedenen Segmenten zu betrachten. Man könnte nun mit der hier entwickelten Applikation die IDs der Äste entlang eines Pfads ermitteln und in einem zweiten Schritt die entsprechenden Spalten in der Ergebnisdatei manuell extrahieren. Einfacher wäre es jedoch, diesen Prozess in der Applikation selber durchzuführen. Leider ist der Umgang mit den zeitveränderlichen Ergebnisdaten nicht mit dem selben Mechanismus handhabbar, mit dem bereits die statischen Parameter der Äste wie Länge, Durchmesser etc. erfasst werden können. Stattdessen wird die Option angeboten, eine Teilmenge von Ästen zu selektieren (vgl. Abschnitt 4.4.2), die eingelesenen Ergebnisdaten zu filtern und als neue Textdatei auszugeben. Dabei werden tatsächlich nur noch die Spalten der selektierten Äste ausgegeben. Die ID jedes Astes steht zu Beginn jeder Spalte, sodass eine eindeutige Zuordnung möglich ist, beispielsweise zur Koppelung der dynamischen Daten mit strukturellen Parametern, die auf dem Wege der bereits beschriebenen Statistikfunktion erfasst wurden. Dieser Mechanismus stellt ein ausreichendes Mittel dar, mit den Datenmengen der Simulation umzugehen.

5 Ergebnisse und Diskussion

Die in dieser Arbeit entworfene Applikation erlaubt es, die aus radiologischen Datensätzen extrahierten morphologischen und strukturellen Informationen zu visualisieren und quantitativ zu erfassen. Es wurden zwei Modelle entwickelt, welche verschiedene Abstraktionsgrade der vorhandenen Daten repräsentieren. Das erweiterte geometrische Modell baut auf dem Basismodell auf, verwendet die gleichen Datenstrukturen und bietet die gleichen Benutzerschnittstellen. Zusammen mit den zur Verfügung gestellten Parametern zur Detailreduktion ist es möglich, den Grad an Detailtreue an die Anforderungen des Benutzers anzupassen. Dies hat Auswirkungen auf die Visualisierung, die morphologische Analyse und auch auf eine nachfolgende numerische Simulation. Es sollen nun einige Ergebnisse anhand von exemplarischen Datensätzen aufgeführt werden, um qualitative und quantitative Unterschiede der Modellierungsarten bezüglich Darstellung und Struktur aufzuzeigen. Zum Abschluss stellt sich die Frage, inwiefern das entwickelte Visualisierungs- und Analysemodell auch in anderen Einsatzgebieten Verwendung finden könnte.

5.1 Quantitativer Längenvergleich

Wie in Kapitel 3 beschrieben, lassen sich die zur Verfügung stehenden Daten auf verschiedene Arten in eine Baumgeometrie umsetzen. Im Basismodell wird jeder Ast durch ein geradliniges Segment beschrieben, im erweiterten Modell wird die Mittelachse durch eine Vielzahl von Teilsegmenten modelliert. Das Basismodell kontrahiert daher die Astlängen; wie stark diese Kontraktion ist, hängt von der Grobstruktur des Astes ab. In realen Datensätzen finden sich Beispiele von stark gekrümmten Ästen, deren direkte Entfernung zwischen Anfangs- und Endpunkt enorm von der tatsächlichen Astlänge abweichen kann. In Abb. 5.2(a) ist solch ein „Worst-Case“ abgebildet, der im Basismodell eine Länge von 4 mm aufweist, im erweiterten Modell jedoch 9,4 mm lang ist¹. Im optimalen Fall wie in Abb. 5.2(b) ist ein Ast nahezu gerade; beim abgebildeten Ast weichen die Längen um lediglich 0,2% ab.

Auch innerhalb des erweiterten Modells ist die Länge eines Astes nicht völlig klar vorherzusehen, da verschieden starke Glättungen der Mittelachse deren Feinstruktur verändern. Um einen Eindruck der Varianzen zu vermitteln, wurden die Längen aller 1431 Äste eines Datensatzes des Ausgussmodells von Abb. 2.1(a) im Basismodell und in verschieden stark geglätteten Versionen des erweiterten Modells untersucht. Die Längenmessung erfolgte mithilfe der bereits entwickelten grafischen Schnittstellen zur Statistik. Die Längen der Trachea², des rechten und linken Hauptbronchus' (RHB/LHB) und zwei weiteren Ästen sind exemplarisch in Tab. 5.1 aufgezeigt.

¹Bei einfach geglätteter Mittelachse.

²Man beachte, dass die Trachea durch den CT-Scan nicht vollständig erfasst wurde, daher ist die Länge des Astes nicht mit der anatomischen Dimension von ca. 12 cm vergleichbar.

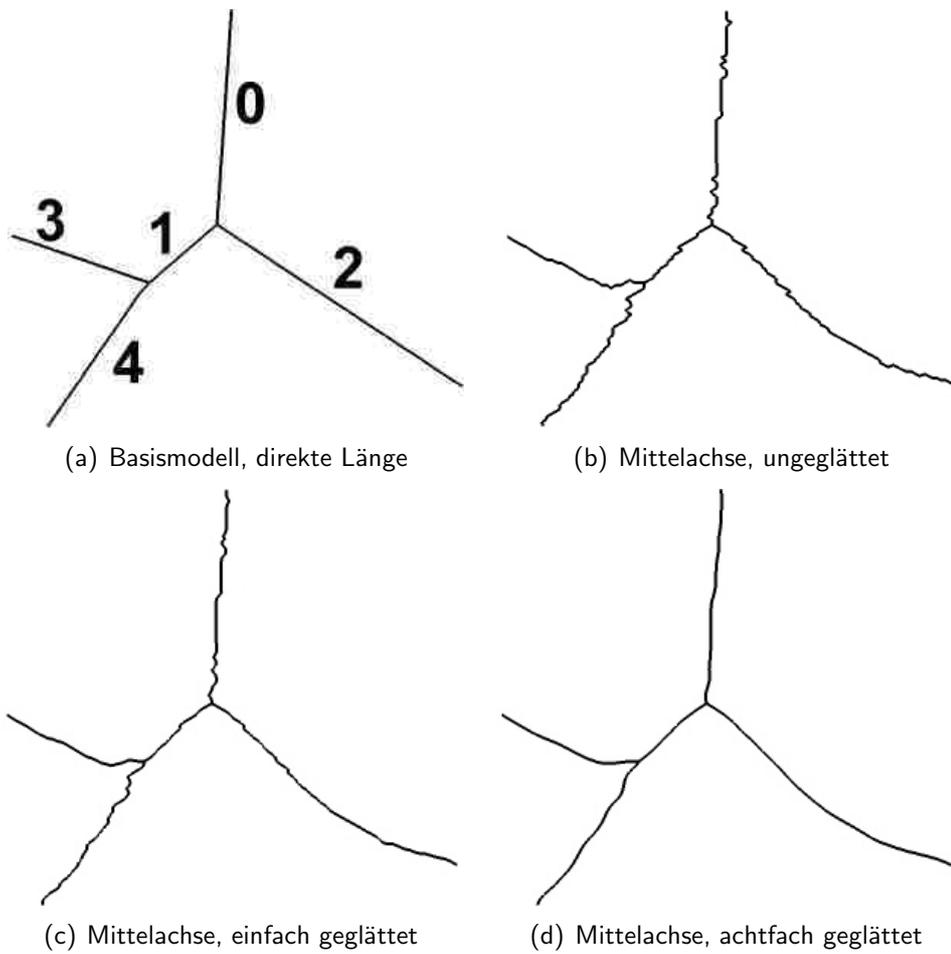


Abbildung 5.1: Längenunterschiede durch Varianzen der Feinstruktur

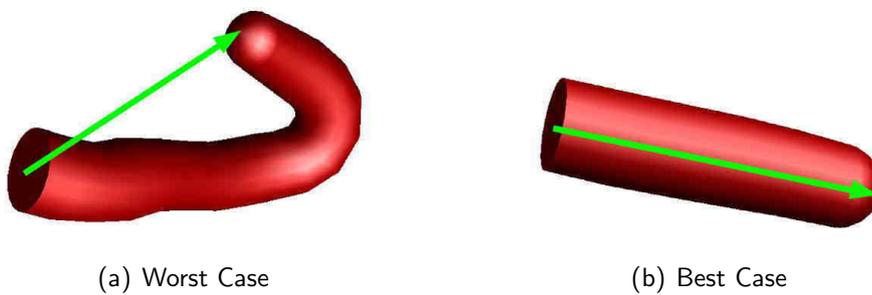


Abbildung 5.2: Längenunterschiede auf Grund der Grobstruktur

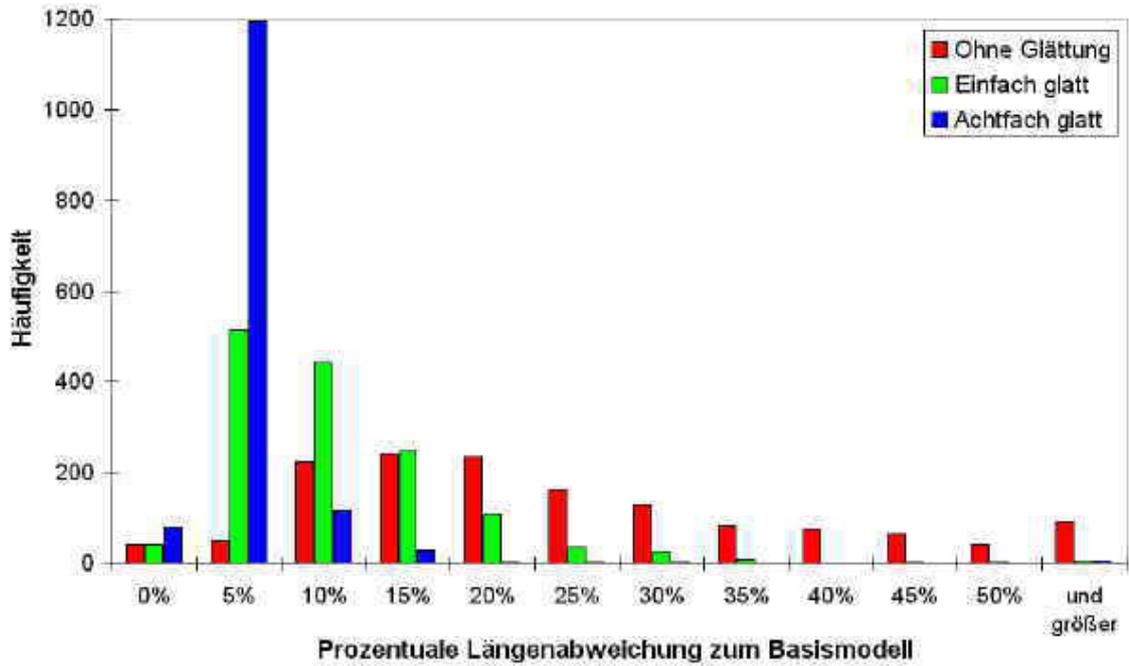


Abbildung 5.3: Längenvergleich Basismodell vs. erweitertes Modell (N=1431) bei verschiedenen starker Glättung der Mittelachsen

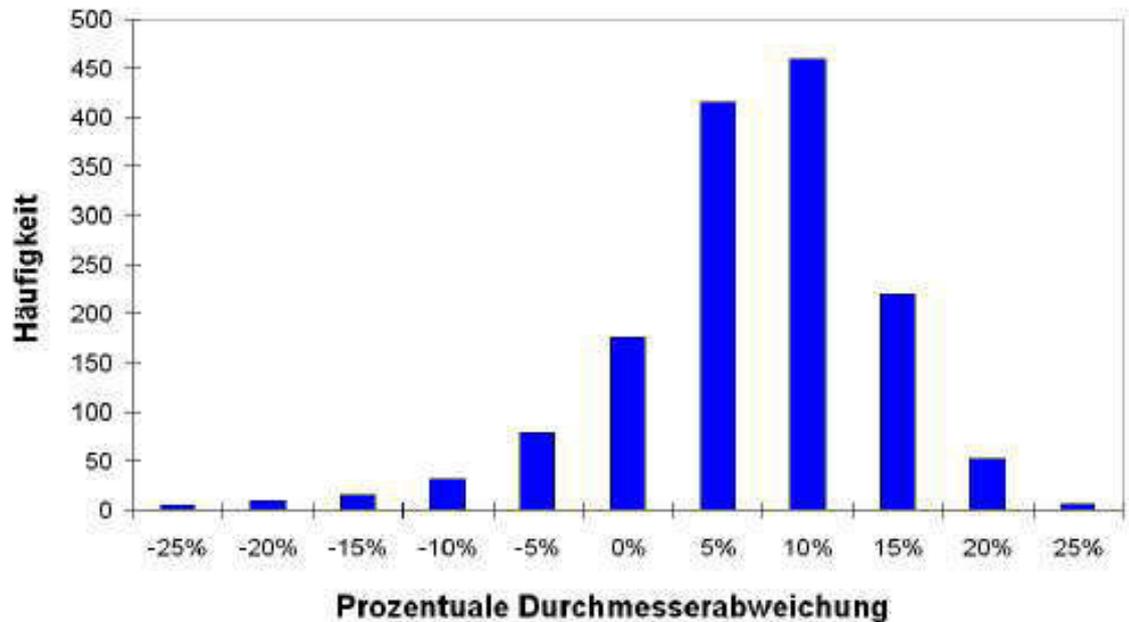


Abbildung 5.4: Durchmesservergleich Basismodell vs. erweitertes Modell (N=1431)

Länge[mm] Ast-ID \	Mittelachse ungeglättet	Mittelachse einfach glatt	Mittelachse achtfach glatt	Direkte Länge
0 (Trachea)	42.0	33.6	30.3	29.9
1 (RHB)	16.4	14.2	13.5	13.4
2 (LHB)	57.4	49.6	47.3	46.3
3	26.2	24.4	23.5	22.9
4	38.6	31.8	29.2	28.7

Tabelle 5.1: Exemplarischer Längenvergleich

Die fünf exemplarischen Äste weichen in ihrer Grobstruktur nicht stark voneinander ab. Dennoch erkennt man, dass vor allem die ungeglättete Mittelachse große Abweichungen zu den restlichen Längen aufweist. Dies ist darauf zurückzuführen (vgl. [Sel99]), dass die Punkte auf der Mittelachse diskrete Voxelkoordinaten sind; durch die Skelettierung können beinahe gerade Äste eine „Zickzack“-Mittelachse erhalten. Im schlechtesten Fall wird daher die Länge um den Faktor $\sqrt{2}$ überschätzt. Im vorliegenden Fall werden die fünf Äste im Vergleich zur einfach geglätteten Mittelachse im Mittel um den Faktor 1,17 verlängert. Eine einfache Glättung der Skelettlinie ist daher unabdingbar. Durch weitere Glättungsoperationen nähert sich die Mittelachse immer mehr der geradlinigen Entfernung des Basismodells an. Die „tatsächliche“ Astlänge wird zwischen den beiden Extremen liegen³.

Im Histogramm von Abb. 5.3⁴ ist die Verteilung der relativen Verlängerung im Vergleich zum Basismodell für alle 1431 Äste aufgezeigt. Bei einer starken Glättung der Mittelachse werden die Abweichungen in der Feinstruktur weitestgehend unterdrückt, und in der Tat weichen bei achtfacher Glättung 97,2% aller Äste nur um maximal zehn Prozent von der Länge im Basismodell ab. Vier der 1431 Äste sind noch um mehr als 50% verlängert. Das sind Äste ähnlich dem „Worst-Case“ von Abb. 5.2(a).

Die Längenkontraktion bei stark gekrümmten Ästen im Basismodell ist nicht zu vermeiden. Im erweiterten Modell spielt auch die Güte der Visualisierung eine Rolle bei der Entscheidung für die Stärke der Glättungsoperation: eine einfache Glättung ist in jedem Fall vorzunehmen, eine stärkere Glättung erscheint nach einer qualitativen Betrachtung der Mittelachsen von Tab. 5.1 durchaus angebracht.

5.2 Quantitativer Durchmesservergleich

Basismodell und erweitertes Modell unterscheiden sich auch hinsichtlich der mittleren Astdurchmesser. Im Basismodell wird der Durchmesser des Zylinders aus der Astlänge und dem Gesamtvolumen berechnet. Im erweiterten Modell werden direkt die Durch-

³Die Länge natürlicher, nicht-euklidischer Objekte kann nicht einfach definiert werden, vgl. Mandelbrot [Man67].

⁴Die Klasseneinteilung der Histogramme ist wie folgt: sind z.B. die Klassen ...15%,20%,25%... gegeben, so fallen Werte $15% < x \leq 20%$ in die Klasse 20%.

messerinformationen entlang der Mittelachse verwendet, die auf andere Art und Weise ermittelt wurden (vgl. [Sel99]). Dadurch unterscheidet sich der mittlere Durchmesser eines unterteilten Bronchus im erweiterten Modell vom Durchmesser des zylindrischen Astes im Basismodell. Diese Variationen sind im Folgenden für den gleichen Datensatz wie im vorherigen Abschnitt deskriptiv angegeben. Der Datensatz enthält 1431 Bronchi, also ebenso viele Äste im Basismodell. Im erweiterten Datensatz sind insgesamt 18157 (unterteilte) Äste vorhanden.

Einem Ast im Basismodell entsprechen demnach eine Anzahl n von Ästen im erweiterten Modell. Für die Untersuchung wurden die Mittelwerte der Durchmesser dieser n Äste mit dem entsprechenden Astdurchmesser im Basismodell verglichen. Die Verteilung der prozentualen Abweichung ist im Histogramm von Abb. 5.4 dargestellt. Eine positive Abweichung bedeutet hier, dass der Ast im Basismodell dicker ist als der entsprechende erweiterte Ast. Dies trifft für 78% aller Äste zu, 20% haben im erweiterten Modell einen größeren Durchmesser. Für zwei Prozent ist keine Änderung des Durchmessers zu beobachten. Im Mittel sind die Äste im Basismodell um 4,1% dicker als im erweiterten Modell. Eine Korrelation der Durchmesservarianzen mit Astlänge, Durchmesser oder Lage im Baum konnte nicht gezeigt werden. Eine weitergehende Diskussion dieser Unterschiede wäre dennoch von Interesse, da die Modellierung der Durchmesser von hoher Bedeutung für die Flussaufteilung bei der Simulation ist. Dies würde eine genaue Betrachtung der Methoden zur Erstellung der Datensätze beinhalten, die jedoch in dieser Arbeit als gegeben vorausgesetzt werden.

5.3 Performance-Betrachtungen

Die Anzahl der Dreiecke, die für die Baumvisualisierung erzeugt werden, lässt sich wie in Abschnitt 4.2 beschrieben durch zwei Parameter variieren: Die Anzahl der Polygonpunkte pro Kreisfläche auf Generation Null, und die Abnahme dieser Anzahl pro Teilungsgeneration. Hier soll exemplarisch an einem komplexen Beispieldatensatz im erweiterten Modell die Darstellungsgeschwindigkeit in Abhängigkeit dieser Parameter untersucht werden. Die geeignete Maßzahl dafür ist die benötigte Zeit für ein Teilbild beziehungsweise die Teilbilder pro Sekunde (*frames per second, fps*), die bei ständiger Aktualisierung der Darstellung erreicht werden können. Bei der Messung dieser Werte wurde mit der C-Standardfunktion `clock()` die Zeit gemessen, die für die Darstellung von 50 Teilbildern nötig ist. Während der Messung wurde die Baumstruktur nur in den hinteren Puffer gerendert und nicht auf dem Bildschirm visualisiert. Dies ist sinnvoll, um Beeinflussungen durch die Bildwiederholfrequenz des Monitors auszuschalten: da eine Darstellung nur zum Bildwechsel (der etwa mit 80Hz erfolgt) möglich ist, wären die Frameraten an diesen Takt gebunden (bei 80Hz z.B. 80/1,80/2,80/3 usw.) – vergleiche hierzu [WND97].

Die Anzahl der modellierten Dreiecke lässt sich für die beiden Elemente Konussegment und Halbkugeln wie folgt angeben: Bei N Polygonpunkten besitzt ein Kegelstumpf N Seitenflächen mit jeweils zwei Dreiecken. Die Halbkugeln werden konstant mit fünf Segmenten modelliert, die aufeinander gestapelt sind. Vier davon sind Kegelstümpfe,

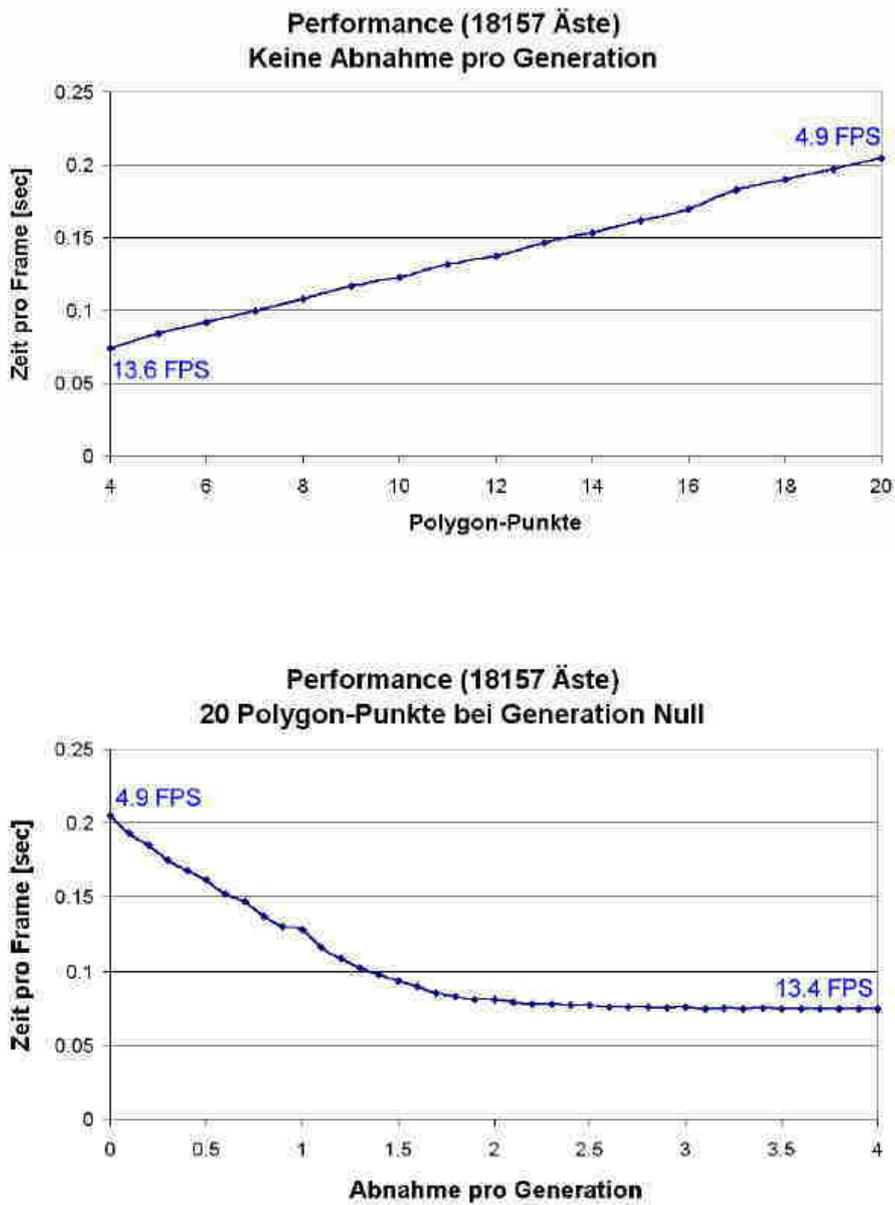


Abbildung 5.5: Performance der Visualisierung in Abhängigkeit des Detailgrades

das letzte eine Pyramide, die pro Seitenfläche lediglich ein Dreieck benötigt. Halbkugeln werden an Bifurkationen und Blättern modelliert, und jeder (Teil-)Ast wird durch genau ein Konussegment repräsentiert. Seien also n die Gesamtzahl der Äste in einem Baum, b die Zahl der Bifurkationen und l die Zahl der Blätter, dann ergibt sich die Gesamtzahl der Dreiecke bei konstant N Polygonpunkten zu

$$T = n \cdot 2N + (b + l) \cdot 9N$$

Da $b + l \leq n$, steigt die Anzahl der Dreiecke in jedem Fall linear mit der Gesamtzahl der Äste ($T(n) = O(n)$). Im Fall einer stetigen Abnahme der Polygonpunkte N pro Teilungsgeneration muss man die Anzahl der Äste pro Generation in Betracht ziehen. Eine Formel hierfür anzugeben ist wenig sinnvoll, jedoch ist zu beachten, dass die Anzahl der Äste in sukzessiven Teilungsgenerationen stark anwächst. Dieser exponentielle Zuwachs kann durch die begrenzte lineare Abnahme lediglich abgemildert werden.

Der Beispieldatensatz besteht aus 18157 Teilästen, davon 729 Blätter, und 728 Bifurkationen. Er wurde auf einem handelsüblichen PC (AMD Duron, 1200 MHz) und einem Consumer-3D-Grafikadapter (Nvidia Geforce 3Ti200) unter Linux visualisiert. In einem Testlauf wurde die Anzahl der Polygonpunkte ohne Abnahme pro Generation betrachtet, begonnen bei 20 (exzellente Qualität) bis hinunter zu vier. Die Gesamtzahl der Dreiecke variiert daher von ca. 1 Mio. (988.540) bis ca. 200.000 (197.708). Die für ein Frame benötigten Darstellungszeiten sind in Abb. 5.5 oben zu sehen⁵. Bei 20 Polygonpunkten wird eine Framerate von 4,9 erreicht, bei vier Polygonpunkten sind es 13,6 fps. Die Zunahme der Visualisierungsgeschwindigkeit ist linear.

Im zweiten Testlauf wurden verschiedene Abnahmefaktoren pro Generation bei konstant 20 Polygonpunkten am Wurzelast getestet. Die Ergebnisse zeigt Abb. 5.5 unten. Man erkennt auch hier eine lineare Zunahme der Geschwindigkeit bei stärkerer Abnahme (bis etwa 2 Polygonpunkte pro Generation). Danach ist keine merkliche Beschleunigung mehr zu erwarten, da der Großteil der Äste bereits mit der minimalen Polygonzahl von vier dargestellt wird. Trotzdem werden die oberen Äste noch detailliert dargestellt. Aufgrund dieser Daten und der qualitativen Betrachtung des Datensatzes bildet eine Abnahme von 1,3 pro Generation, bei 20 Polygonpunkten auf Generation Null, einen guten Standardwert, der auch bei komplexen Baumgeometrien eine befriedigende Framerate gewährleistet.

5.4 Erweiterungen, weitere Anwendungsbereiche

Die Applikation und die Datenstrukturen, die für diese Arbeit Verwendung fanden, wurden in Hinblick auf eine Verwendung innerhalb des DFG-Lungenprojekts konzipiert. Geometrisches Modell und morphologische Parameter wurden abgestimmt auf den Umgang mit Baumstrukturen, die einen Bronchialbaum repräsentieren, und spezielle Schnittstellen zur Integration von Ergebnissen der numerischen Simulation wurden

⁵Eine Darstellung der Frameraten, also des Kehrwerts der Zeiten, ist intuitiver, würde jedoch die Linearität der Geschwindigkeitszunahme nicht widerspiegeln.

implementiert. Baumartige Strukturen findet man jedoch auch in anderen – vor allem medizinischen – Bereichen wie der Leber [Sel99] oder dem Blutkreislauf. Für diese können ganz ähnliche Graphenstrukturen generiert werden wie diejenigen, welche die Basis für diese Arbeit bildeten. Allerdings sind die Strukturen in dieser Arbeit einigen Restriktionen unterworfen, da der Bronchialbaum in der Tat einen strikten Binärbaum darstellt. In anderen Gebieten sind oft mehrere disjunkte Teilbäume oder Multifurkationen vorhanden, die im entwickelten Modell nicht vorkommen dürfen. Diese Beschränkungen sind integraler Bestandteil der entworfenen Datenstrukturen und daher nicht ohne weiteres aufzuheben. Auch das Visualisierungsmodell ist direkt auf die Erfordernisse des Lungenprojekts zugeschnitten, namentlich die Visualisierung dynamischer Daten mit hoher Ortsauflösung. Anderen Eigenschaften, wie der exakten Modellierung im Bereich von Verzweigungspunkten, wurde eine geringere Priorität zugewiesen. Für diagnostische und therapeutische Anwendungen im klinischen Bereich eignen sich andere, oftmals aufwendigere Darstellungsmodelle besser [FKFW02, HPSP01, Zim98]. Die morphologische Analyse ist jedoch eine Stärke des hier entwickelten Systems, ebenso die Möglichkeit, auch im erweiterten Modell jedes Teilsegment individuell ansteuern zu können (vgl. Abschnitt 3.2.3), um dynamische Daten zu visualisieren.

Bei der praktischen Arbeit mit der Applikation offenbaren sich eine Reihe möglicher Erweiterungen. Hierbei steht vor allem eine erhöhte Interaktion mit der Baumstruktur im Vordergrund, beispielsweise Möglichkeiten zur Veränderung der Geometrie. Die zur Verfügung stehenden Basisdaten sind oftmals nicht optimal – irrelevante Seitenäste können auftreten, die Durchmesserinformationen für das erweiterte Modell können verzerrt sein (vgl. rechter Hauptbronchus in Abb. 4.6), Äste sind möglicherweise nicht komplett erfasst. Hier bieten sich interaktive Editieroptionen für manuelle oder semi-automatische Änderungen an. Im Zuge dessen wäre auch eine ausgereifere Navigation im Raum vorteilhaft; die implementierten Steuerungsoptionen offenbaren Schwächen, wenn es etwa darum geht, einem Pfad im Baum zu folgen oder wahlfrei um bestimmte Punkte zu rotieren.

Abschließend wäre eine Integration der Programmmodule in ein größeres Bildverarbeitungssystem wünschenswert, etwa in das ILAB4-System von MeVis, mit dessen Hilfe ebenfalls alle vorgelagerten Bearbeitungsschritte vollzogen werden (Segmentierung, Mittelachsentransformation, Erstellung der Graphenstruktur). Die spezifischen Vorteile der Applikation könnten so zum Beispiel gemeinsam mit dem Volumenrendering anderer Komponenten genutzt werden, um etwa respiratorische Einheiten an den Blättern der Baumstruktur zu visualisieren. Derartige hybride Ansätze vereinen die Vorteile beider Methoden und würden das hier entwickelte Modell auch in Bezug auf die Visualisierung von Simulationsergebnissen verbessern können.

6 Zusammenfassung/Summary

Die interaktive Visualisierung von tubulären Verzweigungsstrukturen wird in zahlreichen medizinischen Anwendungen gefordert, zum Beispiel im Bereich der Diagnose, der Planung von operativen Eingriffen oder der radiologischen und anatomischen Ausbildung. Moderne bildgebende Verfahren wie die hochauflösende Computertomografie (HRCT) erlauben die nichtinvasive Erforschung komplexer Geometrien wie Blutgefäßsystemen oder hochstrukturierten Organen, etwa Leber und Lunge. Diese Arbeit befasst sich mit der Repräsentation des Bronchialbaums der Lunge *in silico*, mit einem Schwerpunkt auf der Erfassung morphologischer Parameter und der Visualisierung dynamischer Daten aus numerischen Simulationen.

HRCT-Scans von präparierten Ausgussmodellen oder *in vivo* Lungen liefern dreidimensionale Daten mit einer Auflösung im Millimeterbereich. Durch eine Reihe bildverarbeitender Techniken werden diese Daten in eine Graphenstruktur überführt, welche die Geometrie des Verzweigungsbaums repräsentiert. Diese Datenstruktur erlaubt eine automatisierte Erfassung morphometrischer Daten, z.B. mittlere Längen und Durchmesser in einer gegebenen Tiefe des Baumes. Außerdem ist eine numerische Simulation des Atmungsvorgangs auf Grundlage der Graphenstruktur möglich.

Auf Basis dieser Daten wurde in der vorliegenden Arbeit eine Software-Applikation entworfen, die es erlaubt, Bronchialbäume als 3D-Oberflächenmodelle in verschiedenen Detail- und Abstrahierungsgraden darzustellen. Im Basismodell wird jeder Bronchus durch ein einzelnes konisches Segment approximiert; in einem erweiterten Modell wird eine hohe räumliche Auflösung erreicht, indem alle verfügbaren Daten zum Astdurchmesser verwendet werden und jeder einzelne Bronchus durch eine Vielzahl verketteter konischer Segmente modelliert wird (Die Datenstrukturen der beiden Modelle sind identisch). In allen Fällen kann jedes Segment individuell identifiziert und visualisiert werden; dies ermöglicht die Darstellung farbkodierter Ergebnisse aus der numerischen Simulation sowie die interaktive Selektion beliebiger Teile des Baums. Morphologische Parameter aller Segmente können extrahiert, gruppiert und zur weiteren Analyse exportiert werden.

Die Applikation wurde in C++ und OpenGL implementiert und ist unter Windows und auf Unix-Systemen lauffähig. Sie wurde im Hinblick auf eine Echtzeitdarstellung auf nicht-spezialisierter Hardware entworfen, wie man sie in heutigen PCs vorfindet.

Three-Dimensional Visualisation and Morphological Analysis of the Tubular Structure of Lung

Interactive visualization of tubular branching structures is a demand in numerous medical applications, such as in diagnosis, pre-operative planning or radiological and anatomical education. Modern imaging techniques like high resolution computed tomography (HRCT) enable non-invasive exploration of complex branching geometries such as blood vessels networks or highly structured organs like liver and lung. This thesis focuses on the representation of the bronchial tree of lung *in silico*, with an emphasis on the assessment of 3D morphometric data and the visualization of dynamic data originating from numerical simulation.

HRCT scans of either prepared airway casts or *in vivo* lungs reveal three-dimensional data at (sub-)millimeter resolution. An imaging processing pipeline transforms this data into a graph structure along with quantitative data, which describes the geometry of the branching pattern. This data structure is suitable for an automated assessment of 3D morphometric data, such as mean lengths and diameters of branches in any given depth of the tree. Additionally, such a representation also allows for numerical simulations of the breathing process.

Given this data, a graphical software application was designed that permits to present the tree structure as a 3D surface model in different degrees of detail and abstraction. In the base model, each bronchus is approximated by a single conical element; in an extended model, a high spatial resolution is achieved by utilizing all given diameter information and modeling a single bronchus by a number of small successive conical segments (The data structures for these two models are identical). In each model, every segment may be addressed and visualized individually, allowing the illustration of color-coded results from numerical simulation and interactive selection of arbitrary parts of the tree. Morphological parameters from all segments may be extracted, grouped and exported for further analysis.

The application was implemented in C++ and OpenGL and is portable to Windows and common Unix systems. It is designed to provide real-time interaction on non-specialized graphics hardware that is found in today's consumers PCs.

Abbildungsverzeichnis

Abbildungen

1.1	Schema des Datenflusses im Projekt „Computerlunge“	2
2.1	Bronchialbaum	5
2.2	Vergleich von regulärer und irregulärer Verzweigungsstruktur	6
2.3	Alveolen	7
2.4	Segmentierung und Skelettierung von Bronchialsegmenten	10
2.5	Skelettierungen von 2D-Rohrverzweigungen	11
2.6	Transformation eines skelettierten Baums in eine Graphenstruktur	12
3.1	Virtuelle Bronchial-Endoskopie	15
3.2	Oberflächendarstellungen einer Bronchialbaum-Bifurkation	16
3.3	Vergleich linienhafter und voluminöser Darstellung	17
3.4	Beispiele für Mevis-Graphen	19
3.5	Erweiterte geometrische Informationen im Mevis-Graphen	21
3.6	Vergleich krummliniger und zylindrischer Äste	22
3.7	Mögliche Darstellungen der Baumstruktur im Basismodell	23
3.8	Erweitertes Modell	25
3.9	Bestimmung von Normalenvektoren der Kreisebenen	27
3.10	Rotation einer Kreisfläche	27
3.11	Lagekorrektur von Kreisflächen	27
3.12	Korrektur des Kreisflächenradius	28
3.13	Obere Atemwege im erweiterten Modell	30
3.14	Farbfunktion Temperaturskala	31
3.15	Vergleich verschiedener Maßzahlen im Basis- und im erweiterten Modell	34
4.1	Klassendiagramm <code>MevisFile</code> und <code>Lung</code>	40
4.2	Transformation Mevis-Graph zu Lung-Struktur	41
4.3	Modellierung der grafischen Grundelemente in OpenGL	45
4.4	Bronchialbaum mit umhüllenden Parenchym	47

4.5	Klassendiagramm zur GUI	50
4.6	Screenshot der Applikation	50
4.7	Selektionsproblematik bei perspektivischer Projektion	52
4.8	Screenshot der Applikation: rechter unterer Lungenlappen selektiert . .	54
4.9	Beispielhafte Verknüpfungen von Astselektionen	54
4.10	Programmelemente zur Erfassung statistischer Größen	56
5.1	Längenunterschiede durch Varianzen der Feinstruktur	60
5.2	Längenunterschiede auf Grund der Grobstruktur	60
5.3	Quantitativer Längenvergleich	61
5.4	Quantitativer Durchmesservergleich	61
5.5	Performance der Visualisierung	64

Tabellen

3.1	Durchmesser pro Weibel-Generation	36
4.1	Binärformat zur Speicherung der Lung-Struktur	44
4.2	Semantik der Selektionsmodi	53
5.1	Exemplarischer Längenvergleich	62

Literaturverzeichnis

- [BEP⁺] BENNETT, Stephen ; ELDRIDGE, M. ; PUENTE, C. ; RIEDL, R. ; NELSON, T. ; GOETZMANN, B. ; MILSTEIN, J. ; SINGHAL, S. ; HORSFIELD, K. ; WOLDENBERG, M. *Origin of Fractal Branching Complexity in the Lung*. <http://hemodynamics.ucdavis.edu/fractal>
- [CH94] CLARK, John ; HOLTON, Derek A.: *Graphentheorie – Grundlagen und Anwendungen*. Spektrum Akademischer Verlag Heidelberg, 1994. – ISBN 3-86025-331-X
- [Dom99] DOMKE, Christian: *Visualisierung der linienhaften Topologie und Funktion von Nervenzellen*, Fachhochschule Gießen/Friedberg, Diplomarbeit, Februar 1999
- [FKFW02] FELKEL, Petr ; KANITSAR, Armin ; FUHRMANN, Anton L. ; WEGENKITTL, Rainer: Surface Models of Tube Trees / VRVis Research Center, Vienna, Austria. 2002. – Forschungsbericht. http://www.vrvis.at/TR/2002/TR_VRVis_2002_008_Full.pdf
- [Hab91] HABERÄCKER, Peter: *Digitale Bildverarbeitung*. Hanser, München, 1991
- [HC67] HORSFIELD, K. ; CUMMING, G.: Angles of branching and diameters of branches in the human bronchial tree. In: *J. Appl. Physiol.* 29 (1967), S. 245–259
- [HC68] HORSFIELD, K. ; CUMMING, G.: Morphology of the bronchial tree in man. In: *J. Appl. Physiol.* 24 (1968), S. 373–383
- [Her01] HEROLD, Helmut: *Das Qt-Buch – Portable GUI-Programmierung unter Linux/UNIX/Windows*. SuSE Press, 2001. – ISBN 3-934678-76-9
- [HPSP01] HAHN, Horst K. ; PREIM, Bernhard ; SELLE, Dirk ; PEITGEN, Heinz-Otto: Visualization and Interaction Techniques for the Exploration of Vascular Structures. In: *Proceedings of IEEE Visualization: Conference held October 21 through 26, San Diego, California, USA*, 2001, S. 395–402
- [HT01] HOWATSON-TAWHAI, Merryn: *An anatomically based mathematical model of the human lungs, applied to gas mixing and water vapour and heat transport*, University of Auckland, Diss., February 2001
- [JDRH95] JUSKIW, Scott ; DURDLE, Nelson G. ; RASO, V. J. ; HILL, Doug L.: Interactive rendering of volumetric data sets. In: *Comput. & Graphics* 19 (1995), Nr. 5, S. 685–693
- [KHM⁺02] KIRALY, Atilla ; HIGGINS, William E. ; MCLENNAN, Geoffrey ; HOFFMAN, Eric A. ; REINHARDT, Joseph M.: 3D human airway segmentation for

- clinical virtual bronchoscopy. In: *Acad. Radiol.* 9 (2002), Nr. 10, S. 1153–1168
- [KPT⁺02] KITAOKA, Hiroko ; PARK, Yongsup ; TSCHIRREN, Juerg ; REINHARDT, Joseph M. ; SONKA, Milan ; MCLENNAN, Geoffrey ; HOFFMAN, Eric A.: Automated nomenclature labeling of the bronchial tree in 3D-CT lung images. In: DOHI, Takeyoshi (Hrsg.) ; KIKINIS, Ron (Hrsg.): *Lecture Notes in Computer Science* Bd. 2489. Utrecht : Springer-Verlag, October 2002, S. 1–11
- [Les03] LESK, Arthur M.: *Bioinformatik: eine Einführung*. Spektrum Akademischer Verlag Heidelberg, 2003. – ISBN 3–8274–1371–0
- [MAMC91] MERCER, Robert R. ; ANJILVEL, S. ; MILLER, Frederick J. ; CRAPO, James D.: Inhomogeneity of ventilatory unit volume and its effects on reactive gas uptake. In: *J. Appl. Physiol.* 70 (1991), Nr. 5, S. 2193–2205
- [Man67] MANDELBROT, Benoit: How Long is the Coast of Great Britain, Statistical Self Similarity and Fractional Dimension. In: *Science* 155 (1967), S. 636–638
- [Man83] MANDELBROT, Benoit: *The fractal geometry of nature*. New York: Freeman, 1983. – ISBN 0–7167–1186–9
- [Mev] MEVIS. *Homepage von Mevis, dem Zentrum für medizinische Diagnose-systeme und Visualisierung*. Website: <http://www.mevis.de/>
- [Sel99] SELLE, Dirk: *Analyse von Gefäßstrukturen in medizinischen Schichtdatensätze für die computergestützte Operationsplanung*, Universität Bremen, Fachbereich Mathematik und Informatik, Diss., September 1999
- [ST90] SCHMIDT, Robert F. (Hrsg.) ; THEWS, Gerhard (Hrsg.): *Physiologie des Menschen*. Springer-Lehrbuch, 1990. – ISBN 3–540–52643–9
- [SWHB95] SMIT, J. ; WESSELS, H. J. ; VAN DER HORST, A. ; BENTUM, M. J.: On the design of a real-time volume rendering engine. In: *Comput. & Graphics* 19 (1995), Nr. 2, S. 297–300
- [Tro] TROLLTECH. *Homepage von Trolltech*. Website: <http://www.trolltech.com>
- [Vos] DE VOS, Louis. *Biodic*. Website: <http://www.ulb.ac.be/sciences/biodic/>
- [Wei63] WEIBEL, E. R.: *Morphometry of the human lung*. Springer-Verlag, Berlin, 1963
- [Wei84] WEIBEL, E. R.: *The pathway for oxygen: Structure and function in the mammalian respiratory system*. Harvard Univeristy Press, Cambridge, Mass., 1984

- [WND97] WOO, Mason ; NEIDER, Jackie ; DAVIS, Tom: *The OpenGL Programming Guide*. Addison-Wesley Longman, Massachusetts, USA, 1997
- [YHJ⁺] YOU, Suya ; HONG, Lichan ; JUNYAPRASERT, Kittiboon ; KAUFMAN, Arie ; MURAKI, Shigeru ; ZHOU, Yong: Interactive Volume Rendering for Virtual Colonoscopy / Center for Visual Computing and Department of Computer Science, State University of New York at Stony Brook. – Forschungsbericht
- [Zim98] ZIMMERMANN, Alexander: 3D-Rekonstruktion von Blutgefäßen mit polynomialen und rationalen Splinefunktionen aus segmentierten MRA Datensätzen. In: LEHMANN, Thomas (Hrsg.) ; METZLER, Volker (Hrsg.) ; SPITZER, Klaus (Hrsg.) ; TOLXDORFF, Thomas (Hrsg.): *Bildverarbeitung für die Medizin 1998, Algorithmen, Systeme, Anwendungen, Proceedings des Workshops am 26. und 27. März 1998*. Universitätsklinikum der RWTH Aachen : Informatik aktuell, Springer, 1998

Alle angegebenen WWW-Links wurden zuletzt am 01.12.2003 auf ihre Gültigkeit überprüft.