Die approbierte Originalversion dieser Diplom-/Masterarbeit ist an der Hauptbibliothek der Technischen Universität Wien aufgestellt (http://www.ub.tuwien.ac.at).

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology (http://www.ub.tuwien.ac.at/englweb/).

Unterschrift des Betreuers

Technische Universität Wien

DIPLOMARBEIT

Segmentierung und Visualisierung von MR-Daten zur Modellierung des arteriellen Blutflusses

ausgeführt am Institut für

Analysis und Scientific Computing

der Technischen Universität Wien

unter Anleitung von

Ao. Univ-Prof. Dipl-Ing. Dr. Felix Breitenecker

durch

Johannes Daniel Leitner

Blumauergasse 1c/3

1020 Wien

30.August 2004

_____

Unterschrift

Vorwort

Ich möchte mich sehr herzlich bei Prof. Felix Breitenecker bedanken, der nicht zuletzt auf der Toskanareise mein Interesse für Simulation geweckt hat.

Großer Dank geht auch an Sigi Wassertheurer vom *ARC* Seibersdorf für die hervorragende Betreuung in den letzten zwölf Monaten. Bedanken möchte ich mich auch bei der gesamten *MEDSIM* Gruppe für die Idee für diese Arbeit, das freundliche Arbeitsklima und die vielen wertvollen Anregungen.

Zu guter Letzt möchte ich mich bei meiner Familie für die große Unterstützung in den letzten Jahren bedanken und die mir damit das Studium ermöglicht haben. Vielen Dank auch meinen Freunden die mit Korrekturlesen, Photoshop-kenntnissen und Hendl Braten diese Arbeit verbessert und erleichtert haben.

Daniel Leitner

Zusammenfassung

Das Ziel der Arbeit ist es, ein Programm zu entwickeln, das aus Magnetresonanz-Bildern eines Patienten Informationen über das Gefäßsystem gewinnt. Diese Informationen sollen als *XML* Datei exportiert werden und dienen als Eingangsdaten die für dynamische Simulation des arteriellen Blutflusses.

Die Simulation des Herzkreislaufsystems dient dazu, dessen Erkrankungen, wie Arteriosklerose, ohne Patienteneinwirkung zu untersuchen, oder Auswirkungen von chirurgischen Eingriffen wie das Legen von Shunts besser vorhersagen zu können. Das Kreislaufsystem wird dazu üblicherweise mit Kompartments beschrieben. Will man lokale Aussagen treffen, muss in einem graphentheoretischen Modell die Blutflussdynamik in den Arterien mit partiellen Differentialgleichungen berechnet werden.

Das hier vorgestellte Programm erzeugt solche graphentheoretische Modelle in Echtzeit aus Volumsdaten, die aus tomographischen Aufnahmen von Patienten gewonnen werden und bietet die Möglichkeit, das Gefäßsystem interaktiv zu bearbeiten. Zur visuellen Kontrolle werden Methoden bereitgestellt, das Gefäßsystems innerhalb der Volumsdaten zu visualisieren.

Die Arbeit beginnt mit einem Überblick über die Physiologie und über grundlegenden Methoden der Modellbildung und Simulation, um dann näher auf die Herzkreislaufsimulation einzugehen. Es wird dabei versucht, aus tomographische Aufnahmen von Patienten Informationen über das Gefäßsystem zu gewinnen. Es werden daher Methoden zur Exploration des Gefäßsystems in den Volumsdaten vorgestellt. Die gängigsten Verfahren der Visualisierung, vor allem der direkten Volumsvisualisierung, werden dabei näher erklärt. Im folgendem wird die Segmentierung der Querschnittsbilder beschrieben und ein Algorithmus zur zuverlässigen Gefäßverfolgung vorgestellt. Das letzte Kapitel beschreibt die Umsetzung des Algorithmus in der Programmiersprache Java und die Benutzeroberfläche des Programms.

Abstract

The aim of this work is to develop software to analyse magnet resonant pictures and to gain information about the cardiovascular system. The information shall be exported to *XML* files and the data is used as an input for dynamic simulation of the cardiovascular system.

The simulation of the cardiovascular system enables to examine diseases like arteriosclerosis virtually without a patient, and to predict the consequences of surgical operations. The cardiovascular system is normally described with compartments. If local results are important the blood circulation must be calculated with partial differential equations within a graph-theoretic model of the vessels.

The program developed in this work creates such a graph-theoretic model in real time from volume data, which are gained from tomographical images from patients, and enables to adapt the model interactively. In order to give a visual control methods to visualize the vessel system within the volume data are explained.

The work starts with an overview about physiology and basic methods of modelling and simulation to explain the simulation of the cardiovascular system. Information about the vessel system shall be extracted from tomographical images from patients. Therefore methods for exploring the vessel system within the volume data are illustrated. The most common techniques in visualisation especially direct volume visualisation are explained more precisely. The work continues with the segmentation of the tomographical images and a new algorithm for reliable vessel tracking is introduced. The last chapter describes the implementation of the algorithm in java and the graphical user interface of the program.

Inhaltsverzeichnis

Kapitel 1 Einleitung 1

Kapitel 2 Herzkreislaufsimulation 4

2.1 Physiologie 4

2.1.1 Kreislauf 4

2.1.2 Herz 6

2.1.3 Blutgefäße 7

2.1.4 Kontrollmechanismen 8

2.1.5 Erkrankungen des Herzkreislaufsystems 9

2.2 Modellbildung und Simulation 10

2.2.1 Überblick 10

2.2.2 Mathematische Beschreibungformen 13

2.3 Modellbildung unter Berücksichtungen physiologischer Vorgaben 16

2.4 Dynamisches Modell 17

2.4.1 Überblick 17

2.4.2 Mathematische Beschreibung 18

2.4.3 Visualisierung der Ergebnisse des dynamischen Modells 19

2.5 Das Kompartmentmodell von Möller 20

2.5.1 Überblick 20

2.5.2 Mathematische Beschreibung 21

2.5.3 Visualisierung der Ergebnisse 22

2.6 Kopplung mit dem Kompartment Modell 23

2.7 Zusammenfassung 23

Kapitel 3 Visualisierung 25

3.1 Überblick 25

3.1.1 Das MRT und CT Verfahren 26

3.1.2 Aufbereitung der Daten 27

3.1.3 Daten auf die Geometrie abbilden 29

3.1.4 Das Rendern der Geometrie 31

3.2 Direkte Volumsvisualisierung 31

3.2.1 Slicing 32

3.2.2 Raycasting 32

3.2.3 Shear-Warp Factorization....33
3.2.4 First Hit....35
3.2.5 Maximum Intensity Projection (MIP)....36
3.2.6 Das Emissions– und Absorptionsmodell....37
3.3 Visualisierung als Werkzeug der Segmentierung....40
Kapitel 4 Segmentierung....41
4.1 Allgemeine Methoden....41
4.1.1 Bereich ausschneiden....41
4.1.2 Helligkeit und Kontrast....42
4.1.3 Binär entscheiden....43
4.1.4 Filter....44
4.1.5 Region Growing....45
4.1.6 Konkave Bereiche ausfüllen....45
4.1.7 Der Watershed Algorithmus....46
4.2 Segmentierung von Arterien....47
4.2.1 Annahmen....47
4.2.2 Aufbereitung der Daten....49
4.2.3 Finden der Ellipsen....50
4.2.4 Arterienbaum finden....52
Kapitel 5 Implementierung....61
5.1 Wahl der Programmiersprache....61
5.2 Programmbeschreibung....62
5.2.1 Öffnen der Querschnittbilder....62
5.2.2 Blickrichtung, Bereich und Kontrast wählen....64
5.2.3 Segmentierung....66
5.2.4 Das Verbinden der Ellipsen....68
Kapitel 6 Ausblicke....72

Kapitel 1
Einleitung

Simulation und Visualisierung sind zwei Gebiete, die seit langem aufeinander angewiesen sind und die gemeinsam interessante Methoden, Werkzeuge und Anwendungen hervorgebracht haben. Speziell in der Medizin entstanden durch die rasante Entwicklung der tomograhphischen Instrumente in den letzten Jahren neue Möglichkeiten der Visualisierung. Die kontrastierte Computertomographie (*CT*) sowie die Magnetresonanztomographie (*MRT*) liefern Bilder mit einer räumlichen Auflösung im Millimeterbereich, auch dreidimensionale Ultraschall-Applikationen werden in den nächsten Jahren diese Qualität erreichen. Durch die hohe Auflösung ist erstmals eine Exploration des Gefäßsystems innerhalb der Volumsdaten möglich. Erkenntnisse, die daraus über das Gefäßsystem gewonnen werden, können für die Simulation des Herzkreislaufsystems verwendet werden. Dadurch können die Auswirkung von Gefäßerkrankungen oder von chirurgischen Eingriffen simuliert werden.

Die Simulation des Herzkreislaufsystems ist aus verschiedenen Gründen eine schwierige Aufgabe. Die Druckverteilung in den Gefäßen hängt sowohl von verschiedenen Parametern, als auch von topologischen Eigenschaften der Arterien ab.
Das dynamische Herzkreislaufmodell [5] beschreibt das Gefäßsystem als Netzwerk von Rohren, die mit hydraulischen Knoten verbunden sind. Mit dynamisch ist gemeint, dass der Druck über einen Herzschlag berechnet wird. Das Modell braucht dabei Informationen über die Konnektivität des Gefäßsystems

sowie die Durchmesser und Längen der Arterien. Diese Daten können aus den tomographischen Aufnahmen eines Patienten gewonnen werden. Das dynamische System kann damit lokale Aussagen über die Eigenschaften bestimmter Arterien treffen.

Ein anderer Ansatz ist das Herzkreislaufsystem mit Kompartments zu simulieren. Das System wird dazu in homogene Einheiten geteilt, und die Wechselwirkung zwischen diesen Einheiten wird mathematisch beschrieben. Das Kompartmentmodell eignet sich gut um Belastungen des Kreislaufs zu simulieren, es können jedoch keine lokalen Aussagen getroffen werden, da das gesamte systemische Hochdrucksystem in einem Kompartment dargestellt wird.

Man kann beide Ansätze kombinieren, um ein gekoppeltes System zu erhalten, das in [1, Kapitel 6] beschrieben ist.

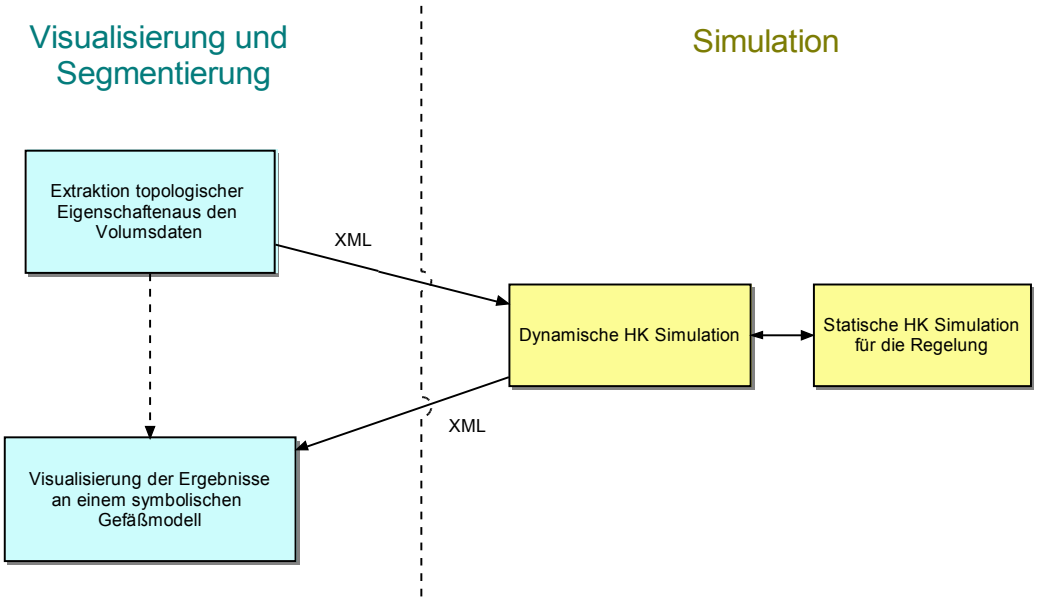


Abbildung 1.1: Zusammenhang zwischen Visualisierung und HK Simulation

Die Arbeit beschäftigt sich zum einen mit Methoden, um topologische Eigenschaften der Gefäßsysteme aus hochauflösenden Volumsdaten zu extrahieren, also mit der Segmentierung und der Analyse der Volumsdaten. Zum anderen wird auf die Visualisierung der Volumsdaten eingegangen. Dabei werden die üblichen Verfahren vorgestellt, aber auch darauf eingegangen, wie aus den extrahierten topologischen Eigenschaften ein idealisiertes Gefäßsystem auf der Basis von Oberflächenprimitiven gewonnen wird, in dem Ergebnisse der

Simulation visualisiert werden können.

Ziel ist es, ein Programm zu entwickeln, das eine Segmentierung, Skelettierung, Radius- und Längenbestimmung, sowie eine Analyse der Konnektivität vornimmt. Die so gewonnen Daten sollen mit *XML* exportiert werden und als Eingang für das bestehende Programm *Cardiosim*, das in [3] entwickelt wurde, dienen. Es muss dabei die Möglichkeit geben, alle Daten interaktiv zu ändern. Um visuelle Kontrolle über die Daten zu haben, ist es notwendig, das Gefäßsystem idealisiert auf der Basis von Oberflächen darzustellen. An diesem Arterienbaum können alle Änderungen vor dem Exportieren vorgenommen werden. Das Programm *Cardiosim* exportiert die Ergebnisse wiederum mit *XML*. Werte wie Druck, Fluss oder Radiusveränderungen können so im idealisierten Arterienbaum veranschaulicht werden.

Kapitel 2
Herzkreislaufsimulation

In diesem Kapitel soll ein Überblick über die Herzkreislaufsimulation gegeben werden. Dazu wird zunächst die Physiologie mit besonderer Berücksichtigung auf das Hochdrucksystem besprochen. Weiters wird eine Einführung über allgemeine mathematische Modellbildung und Simulation gegeben. Anschließend wird näher auf die Herzkreislaufsimulation eingegangen. Es wird dabei die dynamische Simulation, die ein graphentheoretisches Modell des Gefäßsystems benötigt, und das Kompartmentmodell von Möller erklärt und die Kopplung beider Systeme vorgestellt.
Besonderes Augenmerk wird dabei auf die Eingangs- und Ausgangsvariablen der Systeme gelegt, und es soll geprüft werden, ob für die Modelle relevante Parameter aus den tomographischen Aufnahmen eines Patienten extrahiert werden können.

2.1 Physiologie

2.1.1 Kreislauf

Das Herzkreislauf-System des Menschen hat die Aufgabe, die Körperzellen mit Sauerstoff, Nährstoffen und Hormonen zu versorgen. Außerdem werden Abfallstoffe wie Kohlendioxid und Stoffwechselprodukte abtransportiert. Zusätzlich hält das Herzkreislaufsystem die Körpertemperatur stabil. Das

geschieht, indem Hitze aus dem Körperinneren an die Oberfläche transportiert wird.

Der menschliche Kreislauf ist ein Hoch-Niederdrucksystem. Vom Herzen weg wird Blut in die Arterien gepumpt. Von dort gelangt es zuerst in die Arteriolen, kleinste Arterien mit einem Durchmesser weniger von als 0.5 mm, bis hin zu den Kapillaren, in denen der Austausch stattfindet. Daraufhin fließt das Blut in das Niederdrucksystem, in das venöse System. Das venöse System transportiert das Blut zurück zum Herzen.

Das Kreislaufsystem wird in den pulmonären Kreislauf, den Lungenkreislauf, und in den systemischen Kreislauf unterteilt. Im Lungenkreislauf wird das sauerstoffarme Blut durch die Arterien in die Kapillaren der Lunge gepumpt. Dort wird das Blut vom Kohlendioxid befreit und mit Sauerstoff angereichert. Das sauerstoffreiche Blut fließt nun durch die Venen des Lungenkreislaufes zurück zum Herzen. Das Herz pumpt es in den Systemkreislauf. Im Systemkreislauf gibt das Blut den Sauerstoff in den Kapillaren ab, nimmt wiederum Kohlendioxid auf und gelangt durch die Venen zurück zum Herzen, wo es anschließend wieder in den Lungenkreislauf fließt.

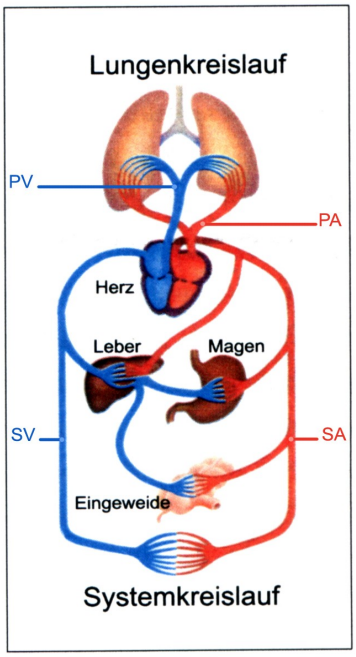


Abbildung 2.1: Der Blutkreislauf

Das Kreislaufsystem hat ein Blutvolumen von ungefähr fünf Litern. Wie inTabelle 2.1 ersichtlich befindet sich davon der Großteil in den Venen und nur ein kleiner Teil im arteriellen System.

Der Druck (*P*) ist die Kraft pro Flächeninhalt, er wird in der Physiologie in mmHg (Millimeter Quecksilbersäule) gemessen. Der Druck ist in den großen Arterien um die 100 mmHg, nimmt in der Peripherie aber immer mehr ab. In den Kapillaren hat er nur noch 20 mmHg, im rechten Atrium ist er fast 0 mmHg. Die Werte beziehen sich auf den über einen Herzschlag gemittelten Druck.

	Druck in mmHg	Volumen in Liter
Arteriell systemisch (AS)	100	1
Venös systemisch (VS)	2	3.5
Arteriell pulmonär (AP)	15	0.1
Venös pulmonär (VP)	5	0.4

Tabelle 2.1: Druck im arteriellen und venösen System

Der Kreislauf ist ein komplexes, stark verzweigtes Netzwerk mit einer Gesamtlänge von 96.500 Kilometern. Nimmt man die Kapillaren dazu, hat das Kreislaufsystem eine Oberfläche von 1000 m².

2.1.2 Herz

Das Herz hat die Aufgabe, das Blut durch das Kreislaufsystem zu pumpen. Es ist in zwei Hälften unterteilt, die durch eine Scheidewand, das Septum, voneinander getrennt sind. Beide Hälften bestehen aus zwei Kammern, dem muskelstärkeren Ventrikel, sowie der muskelschwächeren Vorkammer, dem Atrium.

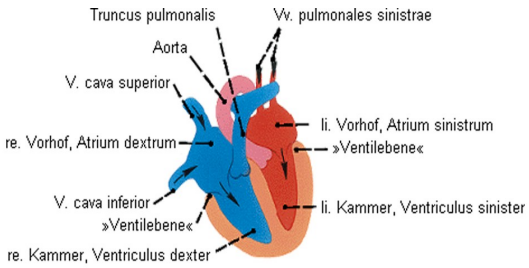


Abbildung 2.2: Das Herz

Das Herz arbeitet in der Art einer Druck- und Saugpumpe. Der Sinusknoten steuert dabei die gleichzeitige Kontraktion der Herzkammern. Während der Ausdehnung, der Diastole, sind die Herzkammern entspannt. Blut fließt dabei von den beiden Vorkammern in die Ventrikel. Die Herzklappen verhindern dabei, dass Blut von der Aorta oder den Lungenarterien in die Ventrikel zurückfließt. Während der Diastole ist der Druck am niedrigsten. Bei der Kontraktion des Herzens, der Systole, ziehen sich die Ventrikel zusammen und pumpen dadurch Blut in die Arterien. Gleichzeitig erweitern sich die Vorkammern und saugen Blut

aus den Venen an. Klappen zwischen Vorkammern und Ventrikeln verhindern das Zurückfließen des Blutes. Während der Systole ist der Druck am höchsten, wie in Abbildung 2.3 ersichtlich ist.

Die Herzfrequenz (*HF*) ist die Anzahl der Herzschläge pro Minute und beträgt in etwa 68. Ein Herzschlag dauert daher um die 0.8 Sekunden, 0.3 Sekunden entfallen dabei auf die Systole, die Diastole benötigt die restlichen 0.5 Sekunden.

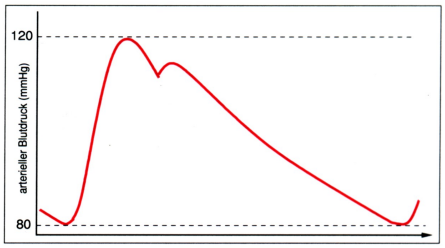


Abbildung 2.3: Druckverlauf in der Aorta über einen Herzzyklus

Das Schlagvolumen (*SV*) ist die Menge Blut, die während einer Systole in das arterielle System gepumpt wird.

Der Fluss (*Q*) ist die Blutmenge in Liter pro Minute, die einen festen Punkt im Kreislauf passiert. Normalerweise betrachtet man das Blutvolumen, das in einer Minute aus dem Herzen gepumpt wird, also ist

$$Q = SV * HF$$

Neben den oben eingeführten Größen besitzen die Blutgefäße noch zwei weitere charakterisierende Eigenschaften, den Widerstand der dem Blutfluss durch das Gefäß entgegengesetzt wird, und die Compliance, das ist die Dehnungsfähigkeit des Gefäßes in Reaktion auf steigenden Druck.

2.1.3 Blutgefäße

Speziell in der dynamischen Simulation des Herzkreislaufsystems spielen die Eigenschaften der Blutgefäße eine wesentliche Rolle. Von besonderem Interesse sind dabei die größeren Gefäße, also die Aorta und die Arterien. Wenn man zu kleine Gefäße berücksichtigt, wird das graphentheoretische Modell des Gefäßsystems sehr groß und damit das System zu komplex.

	Aorta	Arterie	Arteriole	Venole	Vene	V. cava
Wandstärke	2,5 *mm*	1 *mm*	20 μm	10 μm	0,5 *mm*	1,5 *mm*
Innendurchmesser	25 *mm*	4 *mm*	40 μm	60 μm	5 *mm*	30 *mm*

Tabelle 2.2: Übersicht der Wandstärken und Durchmesser der Blutgefäße

Widerstand und Compliance unterscheiden sich stark in den einzelnen Gefäßen. Nahe dem Herzen ist die Compliance hoch, da die Gefäße die durch den Herzschlag entstehenden Stöße auffangen müssen. Am deutlichsten zeigt sich dieses Verhalten am Aortenbogen. Dieser dehnt sich bei jedem Herzschlag aus, dämpft so den Stoß und sorgt für einen gleichmäßigen Blutstrom. Dieses Verhalten wird Windkesseleffekt genannt.
Der Widerstand in den Arteriolen ist von großer Bedeutung, denn durch einen hohen Anteil an glatter Muskulatur in diesen Gefäßen ist es ihnen möglich, große Querschnittsveränderungen durchzuführen und damit ihren Widerstand zu steuern. So kann die Durchblutung einzelner Organe auf den wechselnden Bedarf angepasst werden. Der Gesamteinfluss der Arteriolen wird peripherer Widerstand genannt.

2.1.4 Kontrollmechanismen

Es gibt verschiedene Kontrollmechanismen, die es dem Herzkreislaufsystem ermöglichen, einen adäquaten Blutfluss sicherzustellen. Die Blutmenge, die einzelne Körperteile benötigen, ist dabei wechselnd. Beispielsweise muss bei körperlicher Anstrengung der erhöhte Sauerstoffbedarf der Muskeln gedeckt werden.
Die verschiedenen Mechanismen wirken dabei sehr unterschiedlich. Einige wirken für Sekunden oder Minuten, andere wirken über Tage und Wochen.

Zuerst soll die Autoregulation des Kreislaufsystems erwähnt werden. Sie basiert auf der Abhängigkeit des Flusses vom venösen Druck. Der Fluss durch beide Herzkammern ist gleich, denn wäre er ungleich, käme es zu einer Verschiebung des Blutvolumens. Käme es zu einer solchen Verschiebung, würde sich der Druck

im systemischen oder pulmonären Kreislauf erhöhen, der venöse Druck würde steigen und das würde zu einem erhöhten Fluss in den Kreislauf mit weniger Blutvolumen führen. Daher wird das Gleichgewicht automatisch wieder hergestellt.

Für schnelle Druckänderungen verfügt der Kreislauf über sogenannte Bararezeptoren, das sind Nervenenden, die sich in im Aortenbogen sowie in den Halsschlagadern befinden. Sie reagieren auf Schwankungen des systemischen arteriellen Drucks. Sie senden Impulse an das Nervenzentrum im Rückenmark, die Medulla. Sie steuert das parasympathische Nervensystem (*PNS*) und das sympathische Nervensystem (*SNS*).

Das *PNS* wirkt dabei hemmend, indem es die Herzfrequenz senkt.

Das *SNS* wirkt anregend. Es erhöht die Herzfrequenz und das Schlagvolumen, indem es den venösen Druck erhöht, weiters wird der systemische Widerstand größer. Durch diese Maßnahmen erhöht sich der Druck im systemischen arteriellen Kreislauf.

Die Bararezeptoren passen sich über Stunden an erhöhten Blutdruck an, haben also keine Wirkung als Langzeit-Regelung. Langzeit-Regelung basiert auf einer Reihe von Faktoren wie hormonellen, metabolischen und chemischen Einflüssen. Die Langzeit-Regelung spielt für die kommenden Modelle wenig Rolle und kann aus diesem Grund vernachlässigt werden. Es gibt eine Vielzahl von Regelungsmechanismen des Herzkreislaufes, die teilweise sehr komplex sind und deren Zusammenwirken teilweise noch nicht vollständig erforscht ist.

2.1.5 Erkrankungen des Herzkreislaufsystems

Erkrankungen des Herzkreislaufsystems sind die häufigste Todesursache in den Industrienationen. Es ist daher von großem Interesse, das Gefäßsystem untersuchen zu können und die Folgen von chirurgischen Eingriffen

vorherzusagen. Durch Simulation ist es möglich, virtuell Bypässe und Shunts zu legen und damit deren Auswirkungen auf den Kreislauf besser einschätzen zu können. Im folgenden werden zuerst allgemein die Methoden der Modellbildung und Simulation erklärt. Diese werden anschließend auf das Herzkreislaufsystem angewandt.

2.2 Modellbildung und Simulation

2.2.1 Überblick

Simulation ist die Nachbildung eines dynamischen Prozesses in einem Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind. Ein Sektor der Realität, ein System, wird in einem Modell nachgebildet und in einer geeigneten Programmiersprache realisiert.
Ein System ist eine Zusammenfassung von Objekten, die durch Abhängigkeiten oder Wechselwirkung miteinander verbunden sind.
Unter einem Modell versteht man die Beschreibung eines konkreten Systems. Bei einem mathematischen Modell ist das die formelmäßige Beschreibung des Systems.
Simulation im engeren Sinn ist das Experimentieren mit einem Modell, das meist als Computerprogramm implementiert ist.
Ziel des Experimentierens ist es, Aufschlüsse über Eigenschaften des Systems wie

- Voraussagen über das zukünftige Verhalten des Systems,
- das Finden oder die Analyse von Systemeigenschaften oder
- technische Entwürfe

zu erhalten.

Die Erstellung und die Simulation eines Modells erfolgt in mehreren Schritten:

(1) Problemformulierung
(2) Modellbildung
(3) Experimente am Modell

(4) Interpretation der Ergebnisse

Zunächst muss eine Abgrenzung des Systems gegen die Umwelt erfolgen. Dabei werden jene Merkmale im System berücksichtigt, die notwendig und hinreichend sind, die Aufgabe zu beschreiben. Diese Grenzziehung ist natürlich nicht einfach. Zieht man zu viele Merkmale in Betracht, wird das System sehr komplex und eine Beschreibung ist schwer oder gar nicht möglich. Betrachtet man jedoch zu wenige Merkmale, können wesentliche Systemeigenschaften verloren gehen, und es ist nicht möglich, Aufschlüsse über das System zu erhalten.
Die Komplexität eines Systems hängt also von der Komplexität des zu beschreibenden Systems ab und vom Detaillierungsgrad, der durch die Zielsetzung festgelegt ist.

Die Wechselwirkung zwischen System und Umwelt wird durch Eingänge $u(t)$ in das System und Ausgänge $v(t)$ aus dem System beschrieben.
Eingänge sind beispielsweise Erregung, Störung oder eine Steuerung. Ausgänge könnten Reaktionen des Systems, eine Messung oder Regelgrößen sein.
Man unterscheidet weiters zwischen:

- Exogenen Variablen: sie beschreiben die Umwelt des Systems .
- Endogene Variablen: sie beschreiben den Systemzustand .
- Parameter: sie sind konstante oder von der Zeit abhängige Größen, die das System quantitativ charakterisieren.

Systeme, die nicht von der Zeit abhängen, bezeichnet man als statische Systeme, zeitabhängige Systeme werden dynamische Systeme genannt.
In dynamischen Systemen wird weiters unterschieden, ob die Veränderungen zu einzelnen Zeitpunkten oder kontinuierlich mit der Zeit erfolgen. Ersteres nennt man ein diskretes System, im Gegensatz dazu spricht man von einem kontinuierlichen System. In der Praxis kommt es oft vor, dass beides gemischt wird, also Teile eines diskreten Systems kontinuierlich sind. Solche Systeme werden hybride Systeme genannt.

In einem mathematischen Modell erfolgt die Modellbildung, indem man versucht,

entweder die Struktur des Modells wiederzugeben, oder das Verhalten des Modells nachzubilden. Je nachdem bezeichnet man das Modell als Strukturmodell oder Verhaltensmodell. In Strukturmodellen ist also die Wirkungsstruktur des Modells bekannt und basiert meistens auf physikalischen Gesetzen. Strukturmodelle werden also durch Deduktion gewonnen. Ein typisches Strukturmodell ist beispielsweise das Modell eines elektrischen Netzwerks. In Verhaltensmodellen ist keine Wirkungsstruktur bekannt. Hauptaugenmerk liegt hier auf der Verhaltensübereinstimmung. Die Herleitung von Verhaltensmodellen erfolgt durch Induktion. Annhand der Eingänge und Ausgänge werden Annahmen über die Modellstruktur gemacht, Gesetzmäßigkeiten sind wenige oder keine bekannt. In Verhaltensmodellen ist es daher wichtig, dass eine große Menge an Messdaten vorhanden ist. Populationsmodelle sind beispielsweise typische Verhaltensmodelle. In der Praxis treten meist Mischformen von Verhaltens- und Strukturmodellen auf.

Meist sind in einem Modell nicht alle Parameter bekannt. Unbekannte Parameter müssen daher geschätzt werden. Diesen Vorgang nennt man Identifikation. Man vergleicht dabei die Systemantworten von einer großen Anzahl von Erregungen mit den entsprechenden Modellantworten. Die Parameter werden nun so gesetzt, dass eine größtmögliche Übereinstimmung erzielt wird.
Will man die Güte eines Modells testen, geht man ähnlich vor. Zu Erregungen werden Paare von Systemantwort und Modellantwort verglichen. Diesen Vorgang nennt man Validierung.
Die Identifikation führt zu einer Parameteränderung. Ist jedoch die Güte des Modells in der Validierung nicht hinreichend, muss das Modell geändert werden. Sowohl bei der Identifikation als auch bei der Validierung braucht man eine große Anzahl von Testdaten. Es muss darauf geachtet werden, dass die Identifikation und die Validierung mit verschiedenen Paaren von Erregung und Systemantwort durchgeführt wird, ansonsten wird das Modell während der Validierung zu optimistisch bewertet, da die Parameter während der Identifikation ja mit den gleichen Daten angepasst wurden.

Neben der Validierung gibt es noch die Empfindlichkeitsanalyse, um die Güte

eines Modells zu testen. Bei der Empfindlichkeitsanalyse variiert man Größen, bei denen bekanntermaßen Unsicherheiten bestehen. Meist sind das geschätzte oder gemessene Parameter. Führen kleine Änderungen dieser Größen zu kleinen Änderungen der Systemantwort, spricht man von Robustheit des Systems.

2.2.2 Mathematische Beschreibungformen

Ein kontinuierliches dynamisches mathematisches Modell kann durch seine Systemgleichungen beschrieben werden.

$$\dot{x}(t) = f(x(t), u(t), t) \qquad (Zustandsgleichung)$$
$$v(t) = g(x(t), u(t), t) \qquad (Verhaltensgleichung)$$

Die Zustandsgleichung ist dabei ein System von *n* Differentialgleichungen erster Ordnung. Die Verhaltensgleichung ist ein algebraischer Ausdruck.

- $u(t)$ bezeichnet den Eingangsvektor.
- $v(t)$ ist der Ausgangsvektor.
- $x(t)$ nennt man den Zustandsvektor. Er liefert eine vollständige Beschreibung des Systemzustands. Die Anzahl seiner Komponenten ist die Dimension *n* des Systems.

In linearen System gilt das Superpositionsprinzip. Jedes System bildet aus *n* Eingängen u_i die Ausgänge v_i aus einer Linearkombination der Eingänge. Das heißt die Eingangsgrößen werden um beliebige Faktoren $\alpha_i \in \mathbb{R}$ verstärkt und aufsummiert.

Unter einer Steuerung versteht man, wenn ein bestimmter Ausgang $\bar{v}$ als Systemantwort gewünscht wird, und daraus der Eingang $\bar{u}$ durch vorhergehende Rechnungen und Messungen bestimmt wird.

Eine Regelung wird meist dann verwendet, wenn Modellungenauigkeiten kompensiert werden sollen. Dazu wird ein tatsächlicher Ausgang v , der

sogenannte Istwert mit einem gewünschten Ausgang $\bar{v}$, dem Sollwert, verglichen. Die sogenannte Regelabweichung $\bar{v}-v$ wird von einem Regler verarbeitet, der so einen neuen Eingang $\bar{u}$ erhält, in dem die Regelabweichung vermindert oder beseitigt wurde.

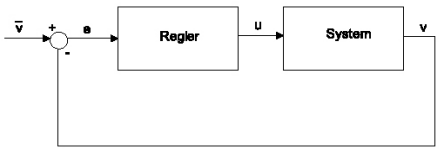


Abbildung 2.4: Blockschaltbild eines Regelkreises

Unter Rückkopplung versteht man den Vorgang, wenn der Systemzustand sich selbst beeinflusst. Solche Systeme unterliegen einer gewissen Eigendynamik. Hat ein System eine postive Rückkoplung, wirkt es selbstverstärkend. Eine negative Rückkoplung hat meist eine Stabilisierung zur Folge. Das System schwingt in einen Gleichgewichtszustand ein.

In Verhaltensmodellen wird versucht, einen Zusammenhang zwischen den Eingängen und den Systemantworten herzustellen. Gesucht ist also eine Funktion *g,* die $v(t)=g(u(t))$ leistet. Häufig ist es möglich, die Funktion *g* mit ihrer sogenannten Übertragungsfunktion zu beschreiben. Dazu werden die Laplacetransformierten $U(s)$ und $V(s)$ des Systemeingangs $u(t)$ und der Systemantwort $v(t)$ betrachtet.
Die Laplacetransformierten $U(s)$ und $V(s)$ sind

$$U(s)=Z[u(t)](s)=\int_0^\infty \mathrm{e}^{-st}u(t)\,dt$$

$$V(s)=Z[v(t)](s)=\int_0^\infty \mathrm{e}^{-st}v(t)\,dt$$

Bei der Laplacetransformation handelt es sich um eine Transformation vom Raum der reelen Funktionen, dem sogenannten Zeitbereich, in den Raum der komplexen Funktionen.
Durch den Differentiationsansatz

$$Z[\dot{u}(t)](s)=sU(s)-u_+(0)$$

und dem Integralansatz

$$Z[\int_0^t u(\tau)d\tau](s)=\frac{1}{s}U(s)$$

ist ersichtlich, dass die Differentialgleichung in eine algebraische Gleichung übergeführt wird.

Setzt man die Transformierten $U(s)$ und $V(s)$ in Zusammenhang, sodass $V(s)=G(s)U(s)$ gilt, nennt man $G(s)$ die Übertragungsfunktion des Systems.

Abhängig von der Gestalt von $G(s)$ können Rückschlüsse auf die Beziehung zwischen $u(t)$ und $v(t)$ gezogen werden.

- Hat $G(s)$ die Gestalt einer rationalen Funktion, wird der Zusammenhang zwischen $u(t)$ und $v(t)$ in Form einer linearen Differentialgleichung mit konstanten Koeffizienten beschrieben.
- Hat $G(s)$ die Gestalt $\frac{1}{Ts+1}$, handelt es sich um ein Verzögerungsglied erster Ordnung. Das heißt, dass der Ausgang dem Eingang mit exponentieller Verzögerung folgt. T beschreibt dabei die Verzögerung. Ist T klein, so ist die Verzögerung gering. Zum Zeitpunkt T hat der Ausgang 63% seines asymptotischen Endwerts erreicht.

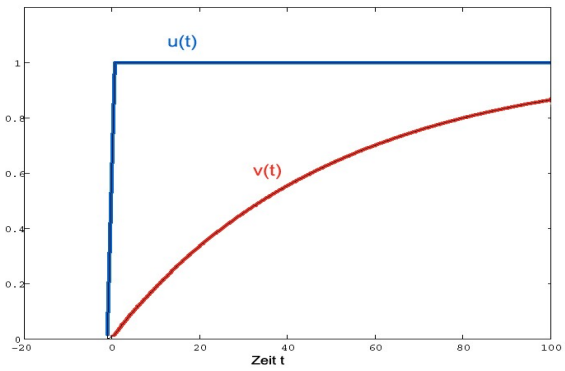


Abbildung 2.5: Verzögerungsglied erster Ordnung mit T = 50

- Ist $G(s)$ konstant, heißt das, der Eingang wird um diesen Faktor verstärkt.

Von der Übertragungsfunktion $G(s)$ kann man also auf das Systemverhalten zurückschließen. Will man einen expliziten Zusammenhang zwischen $u(t)$

und $v(t)$, muss $G(s)$ zurücktransformiert werden.
Häufig ist das jedoch nicht nötig. Die Darstellung eines Systems mit einem sogenannten Blockschaltbild beschreibt das System lediglich mit der Übertragungsfunktion.

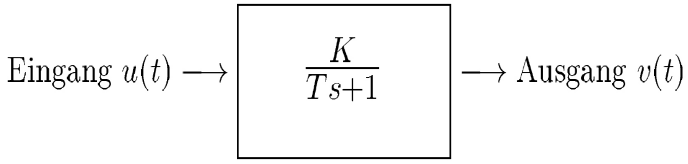


Abbildung 2.6: Blockschaltbild eines Systems

2.3 Modellbildung unter Berücksichtungen physiologischer Vorgaben

Bezugnehmend auf die oben vorgestellten Modellierungsansätze, wird im folgenden auf ausgewählte, besonderes häufig in der mathematischen Physiolgie verwendete, Ansätze eingegangen.

- Dynamische Modellierung mittels Differentialgleichungen
- Kompartement Modellierung
- Deren Kombination

Die aufgezählten Methoden haben Vor- und Nachteile. Deswegen ist es wichtig, deren Einsatz sorgfältig gegeneinander abzuwägen. Die folgenden Fallbeispiele sollen dem Leser einen ersten Eindruck vermitteln.

2.4 Dynamisches Modell

2.4.1 Überblick

Das dynamische Modell berücksichtigt die Fluss- und Druckwellen, die durch die stoßweise Entladung der Ventrikel entstehen. Wegen der unterschiedlichen Compliance der Gefäße haben die Druck- und Strömungskurven eine charakteristische Form. Die Pulsation wird in Richtung der Peripherie durch die Windkesselfunktion der elastischen Gefäße immer mehr gedämpft. Durch reflektierte Wellen kommt es zu einem kleinen Rückfluss in den linken Vorhof.
Das dynamische Modell wird durch ein Netzwerk von Rohren modelliert. Dabei wird jedes Segment des Arterienbaums zwischen zwei Verzweigungspunkten durch ein Rohr und die Verzweigung durch hydraulische Knoten dargestellt, wie in [4] genauer beschrieben ist.

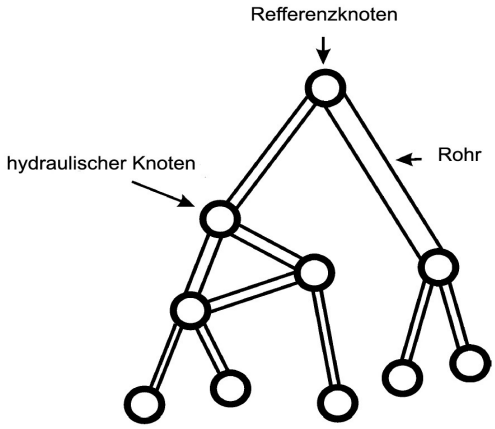


Abbildung 2.7: Der Arterienbaum als Netzwerk

Die berücksichtigten Eigenschaften sind Länge und Durchmesser der Arterien, aber auch die Dehnbarkeit und Dicke der Arterienwände. Das Blut wird vereinfacht als Newtonsche Flüssigkeit angenommen. Die Arterien werden als Rohre dargestellt, sind also zylindrisch dicht und hydraulisch glatt. Die Elastizität wird als linear angenommen, und es wird nur die radiale Ausdehnung berücksichtig, die weitaus stärker ist als die longitudinale. Dichte und

Reibungswiderstand der Gefäßwände werden vereinfacht als konstant betrachtet.
Der Arterienbaum ist sehr stark verzweigt. Bei kleineren Arterien ist es nicht mehr möglich, Eigenschaften wie Länge, Durchmesser und Compliance zu messen, außerdem wird das Modell mit zunehmender Größe zu komplex. Aus diesem Grund wird der Arterienbaum ab einer gewissen Tiefe einfach abgeschnitten.
Es ist möglich, einige der verwendeten Größen aus tomographischen Aufnahmen eines Patienten zu gewinnen. So kann der gesamte Arterienbaum mit Länge und Durchmesser der Arterien aus den Querschnittsbildern aufgebaut werden. Je nach Qualität der Bilder können Arterien bis zu wenigen Millimetern Durchmesser erkannt und deren Länge und Radius vermessen werden. Dieses graphentheoretische Modell des Gefäßsystems eines Patienten kann auf diese Weise für die Simulation verwendet werden.

2.4.2 Mathematische Beschreibung

Die verwendeten Gleichungen, um das Netzwerk von Rohren zu beschreiben, sind die Gleichung für Massenbilanz, die Gleichung für Momentbilanz, die das Kräftegleichgewicht an den Gefäßwänden zurückgibt, und die Zustandsgleichung für achsensymmetrische Röhren, diese folgt aus dem Hookschen Gesetz.
In [5] werden diese Gleichungen näher beschrieben und hergeleitet. Wegen der Annahme der Symmetrie können die Gleichungen auf eine Raumdimension reduziert werden [3]. So ergibt sich ein System partieller Differentialgleichungen.

$$A_t + Q_x = 0$$
$$Q_t + \frac{\partial}{\partial x}(\alpha Q^{2A} + p) = K\frac{Q}{A}$$

Größe	Beschreibung
A	Die Querschnittsfläche kann aus tomographischen Aufnahmen ermittelt werden
Q	Fluss
K	Wandreibungskoeffizient
α	Ein Koeffizient zur Berücksichtigung des nicht flachen Geschwindigkeitsprofil.

Tabelle 2.3: Beschreibung der Größen im dynamischen Modell

Das partielle Differentialgleichungssystem kann nicht explizit gelöst werden. Für die Berechnung werden numerische Methoden herangezogen.

2.4.3 Visualisierung der Ergebnisse des dynamischen Modells

Will man die Ergebnisse des dynamischen Systems visualisieren, ist zu beachten, dass es sich dabei um Informationsvisualisierung handelt, es muss klar sein, dass es sich nur um einen idealisierten Arterienbaum handelt. Da die Ergebnisse aus einem eindimensionalen Modell kommen, ist die Lage der Rohre nicht von Bedeutung, lediglich wie diese miteinander verbunden sind.

Druck oder Fluss können in einem idealisierten Arterienbaum farblich eingezeichnet werden. Der Fluss kann aber nicht nur farblich dargestellt werden, sondern kann mit animierten Partikeln, oder mit Glyphen, die auf texturierte Rohre gezeichnet werden, veranschaulicht werden. In den Glyphen könnte wiederum der Druck farblich aufgetragen werden.

Die Ausdehnung der Arterien bei erhöhtem Druck könnte zur Veranschaulichung prozentuell verstärkt werden. So könnte man beispielsweise in einer Animation sehen, wie sich eine Druckwelle im Netzwerk ausbreitet. Wichtig dabei ist, gut zu dokumentieren, wie sehr die Ausdehnung vergrößert wurde.

2.5 Das Kompartmentmodell von Möller

2.5.1 Überblick

Das Modell nach Möller beschreibt die Regelung des arteriellen systemischen Blutdrucks im menschlichen Körper. Das Modell wurde ursprünglich in [1] vorgestellt. Die Arbeit wurde in [3] ausführlich erläutert und erweitert, außerdem wurde das Modell in [2] neu validiert.

Das ungeregelte Modell von Möller beschreibt ein statisches Modell für den Blutkreislauf. Mit statisch ist gemeint, dass mit über einen Herzzyklus gemittelten Werten des Blutdrucks, peripheren Widerstand und äußeren Einflüssen gerechnet wird. Die durch die stoßweise Entladung der Ventrikel auftretenden Fluss- und Druckwellen werden im Modell also nicht berücksichtigt.

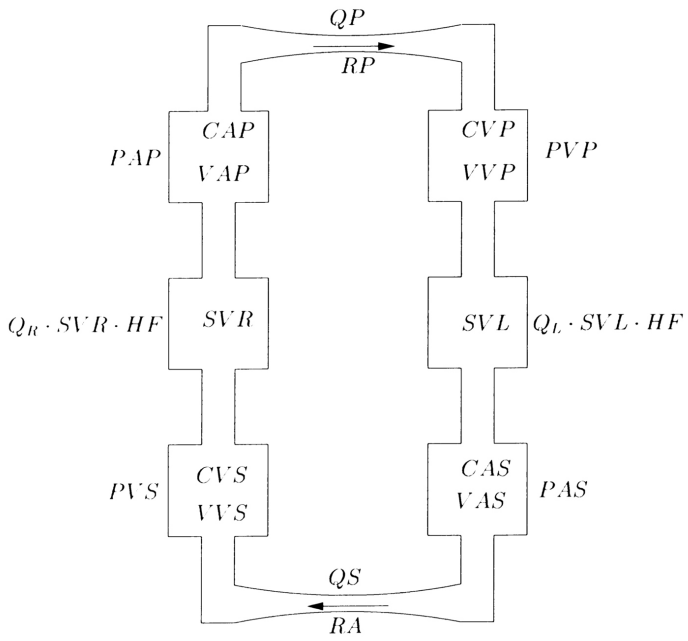


Abbildung 2.8: Schema des ungeregelten Kompartmentmodells nach Möller

Der Blutkreislauf wird als einfaches mechanisches zwei Pumpen- Modell nachgebildet. Blut wird dabei als inkompressibel und von konstanter Dichte angenommen. Neben den zwei Herzkammern gibt es vier weitere Subsysteme,

sogenannte Kompartments. Ein Paar beschreibt den pulmonären Kreislauf, unterteilt in venös und arteriell, das andere Paar den systemischen Kreislauf, wieder geteilt in venös und arteriell.

Das ungeregelte Modell vernachlässigt die barorezeptorische Kurzzeitregulation. Im geregelten Modell wird diese berücksichtigt, es kann damit also simuliert werden, wie das Kreislaufsystem auf Belastungen reagiert. Auf das geregelte Modell soll hier nicht weiter eingegangen werden. Es sei auf [2, Seite 30-40] sowie [3, Seite 29-24] verwiesen.

2.5.2 Mathematische Beschreibung

Das Kompartmentmodell hat als wesentliche beschreibende Größen Volumina und Flüsse. Das Modell stellt einen Zusammenhang zwischen Drücken und Flüssen der einzelnen Kompartments, peripheren Widerstand und Compliance her.
Das unregulierte Herzkreislaufmodell wird durch das folgende lineare inhomogene Differentialgleichungssystem beschrieben. Hergeleitet wird dieses in [1, Seite 46-58].

$$\begin{pmatrix} P\dot{A}S \\ P\dot{V}S \\ P\dot{A}P \\ P\dot{V}P \end{pmatrix} = \begin{pmatrix} -\frac{1}{CAS\,RA} & \frac{1}{CAS\,RA} & 0 & 0 \\ \frac{1}{CVS\,RA} & -\frac{1}{CVS\,RA} & 0 & 0 \\ 0 & 0 & -\frac{1}{CAP\,RP} & \frac{1}{CAP\,RP} \\ 0 & 0 & \frac{1}{CVP\,RP} & -\frac{1}{CVP\,RP} \end{pmatrix} \begin{pmatrix} PAS \\ PVS \\ PAP \\ PVP \end{pmatrix} + \begin{pmatrix} \frac{1}{CAS}Q_L \\ -\frac{1}{CVS}Q_R \\ \frac{1}{CAP}Q_R \\ -\frac{1}{CVP}Q_L \end{pmatrix}$$

Größe	Beschreibung
PAS, PVP, PVS, PAP	Druck des venösen bzw. arteriellen Teils des System- bzw. Lungenkreislaufs
CAS, CVS, CAP, CVP	Als konstant angenommene Dehnbarkeiten
RA, RP	Peripherer Widerstand des System- bzw. Lungenkreislaufs
QL, QR	Blutfluss aus dem linken bzw. rechten Ventrikel

Tabelle 2.4: Beschreibung der Größen im ungeregelten stationären Modell

2.5.3 Visualisierung der Ergebnisse

Normalerweise werden die Druckkurven gemeinsam mit Messdaten in einen Graphen gezeichnet, was eine schnelle visuelle Überprüfung der Güte des Modells ermöglicht. Abbildung 2.9 zeigt den arteriellen systemischen Druck des geregelten Modells in einem Ergometertest. Die rote Linie stellt dabei die Ausgabe des Modells dar, die blauen Punkte sind die gemessenen Werte. Ein großer Nachteil des Kompartmentmodells ist, dass keine lokalen Aussagen getroffen werden können. Es kann lediglich eine Druckkurve über den gesamten arteriellen systemischen Druck ausgegeben werden, jedoch können in diesen Modell keine Aussagen über die Druckkurven der einzelnen Arterien gemacht werden.

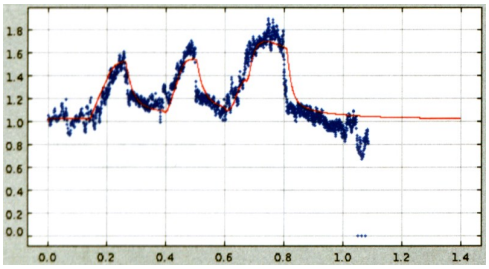


Abbildung 2.9: Der mittlere arterielle Blutdruck im Ergometertest

Zum Visualisieren der Ergebnisse könnte eine schematische Darstellung wie Abbildung 2.8, in der die sechs Kompartments erkennbar sind, verwendet werden. Dabei kann der Druck der einzelnen Kompartments farblich dargestellt und

animiert werden. Einfache Vorgänge, wie die schon erwähnte Autoregulation, könnten so anschaulich als kurzes Video oder sogar interaktiv erklärt werden.

2.6 Kopplung mit dem Kompartment Modell

Das dynamische System ist ein nicht geregeltes Modell. Die Idee liegt daher nahe, das dynamische Modell mit dem geregelten Kompartment Modell zu koppeln.
Die Ergebnisse der Regelung werden als Eingangsgröße für das dynamische System verwendet. Sie sind abhängig von Belastung, Herzfrequenz und peripheren Widerstand. Durch Rückkopplung werden diese Eingangsgrößen angepasst.
Das Ergebnis des dynamischen Modells ist eine Druckkurve. Über diese wird über einen Herzzyklus gemittelt, das Ergebnis dient wiederum als Eingang für die Regelung.

2.7 Zusammenfassung

Das Kapitel hat gezeigt welche Verfahren in der Herzkreislaufsimulation Verwendung finden. Dabei wurden die notwendigen Größen vorgestellt und darauf eingegangen welche davon aus tomographischen Aufnahmen gewonnen werden können.
Im folgenden Kapitel wird erklärt, wie aus den Querschnittsbilder Volumsdaten gewonnen werden, und es werden Verfahren zur Exploration dieser Daten vorgestellt. Dabei ist die direkte Volumsvisualisierung von besonderer Bedeutung, weil damit das Gefäßsystem in Echtzeit dargestellt werden kann, und es damit möglich ist das Gefäßsystem zu untersuchen.
Im weiteren wird auf die Segmentierung dieser Daten eingegangen. Das erfolgt in zwei Schritten. Zuerst werden die Arterien in den Volumsdaten erkannt, daraufhin wird mit einem Algorithmus zur Gefäßverfolgung ein graphentheoretisch Modell

des Gefäßsystems erzeugt, welches dann für die Herzkreislaufsimulation verwendet wird.

Kapitel 3
Visualisierung

Der Sinn jeder Visualisierung ist es Einblick, in die zugrunde liegenden Daten zu erhalten. In diesem Fall will man dadurch einen Eindruck erlangen, wie gut die Struktur der Arterien erkennbar ist, also welches Ergebnis man erwarten kann.
Im folgendem wird zuerst Visualisierungs-Pipeline betrachtet. Es wird besprochen, in welcher Form die Daten vorliegen, wie die Qualität der Daten verbessert werden kann und wie die Daten auf eine Geometrie abgebildet und gerendert werden. Danach wird näher auf die direkte Volumsvisualisierung eingegangen.

3.1 Überblick

Zuerst sollen die einzelnen Schritte der Visualisierungs-Pipeline in der Volumensvisualsierung genauer erklärt werden.

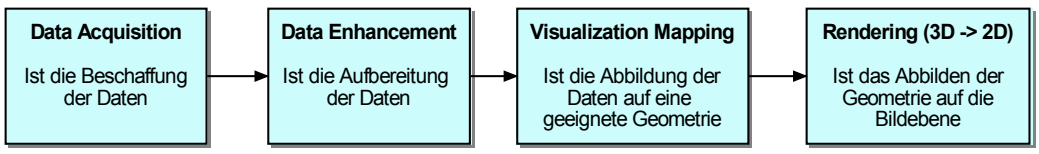


Abbildung 3.1: Die Visualisierungs-Pipeline

3.1.1 Das MRT und CT Verfahren

Die in dieser Arbeit verwendeten Daten sind Querschnittbilder. Die üblichsten Verfahren, Querschnittsbilder verschiedener Körperabschnitte zu erhalten, sind die Magnet-Resonanz-Tomographie (*MRT*) und die Computer-Tomographie (*CT*).

MRT ist eine diagnostische Technik zur Darstellung der inneren Organe und Gewebe mit Hilfe von Magnetfeldern und Radiowellen. *MRT* liefert genaue und differenzierte Darstellungen aller Körpergewebe, vor allem nicht-knöcherner Strukturen, wie Weichteile, Organe, Gelenkknorpel, Meniskus und Gehirn. Strukturen, die einen geringen Wassergehalt haben, wie beispielsweise Knochen, oder luftreiche Regionen wie die Lunge, können dagegen mit der *MRT* nicht gut dargestellt werden.

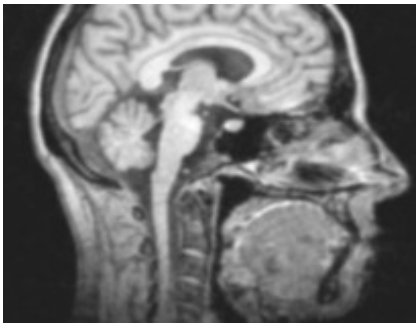


Abbildung 3.2: MRT Querschnitt eines Kopfes

Bei *CT* handelt es sich um ein spezielles Röntgenverfahren, das Querschnittsbilder verschiedener Körperabschnitte anfertigt. *CT* liefert einen besseren Kontrast als *MRT,* und auch Knochen werden gut dargestellt.

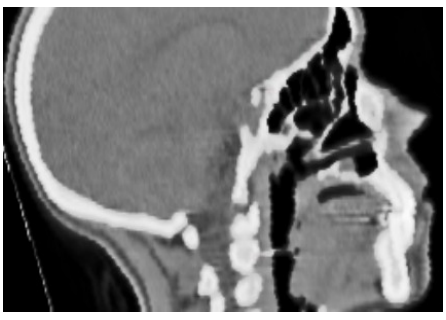


Abbildung 3.3: CT Querschnitt eines Kopfes

Beide Verfahren liefern Bilder mit einer Auflösung bis zu 512*512 Pixel. Bei dieser Auflösung hat eine Arterie im Querschnitt zwischen 45 und einem Pixel. Jeder Pixel hat dabei eine Intensität, die mit 16 Bit beschrieben wird. Um einen ganzen Menschen einzuscannen, benötigt man um die 1500 Querschnitte. Ein solcher Datensatz benötigt in etwa 750 MB.

Will man Information über die Struktur der Arterien erhalten, ist *MRT* zu bevorzugen. Denn im *CT* liegen die Dichtewerte von Knochen und Arterien sehr nahe beieinander. Ohne händische Nachbearbeitung der Bilder ist es kaum möglich, die Arterien zu segmentieren, da Knochen unweigerlich als Arterien erkannt werden und damit mißinterpretiert werden. Aus diesem Grund werden in dieser Arbeit ausschließlich MR-Bilder verwendet.

3.1.2 Aufbereitung der Daten

Das Aufbereiten der Daten ist sowohl für die Visualisierung also auch für das Segmentieren von Bedeutung. An dieser Stelle soll nur ein kurzer Überblick gegeben werden, in Kaptiel 3.1 wird konkret erklärt, wie die MR-Bilder für die Segmentierung modifiziert werden.

Die Querschnittbilder werden in einem regulären Gitter angeordnet. Dieses besteht aus Zellen, deren Ecken, die sogenannte Voxel, die Intensitäten der Pixel repräsentieren. Die Größe des Gitters muss sich nicht unbedingt nach der Anzahl der Bilder und der Anzahl der Pixel in den Bildern richten. Es können durch Interpolation weitere Voxel hinzugefügt werden. Das Umrechnen von einen Gitter auf ein beliebiges anderes Gitter nennt man allgemein Resampling, siehe dazu Kapitel 4.2.2.

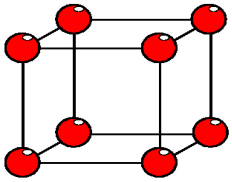


Abbildung 3.4: Die Eckpunkte einer Zelle sind Voxel

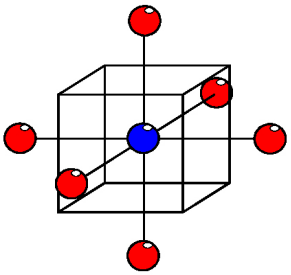


Abbildung 3.5: Die Voxel innerhalb einer Zelle erhält man durch Interpolaiton

Die Punkte in den Zellen werden entweder durch das Nearest Neighbour Verfahren oder durch trilineare Interpolation ermittelt.

NearestNeighbour

$$f(x,y,z)=S(rnd(x),rnd(y),rnd(z))$$

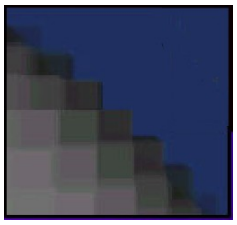


Abbildung 3.6: Nearest Neighbour

Trilineare Interpolation

$$f(x,y,z)=(x-1)(y-1)(z-1)S(0,0,0)$$
$$x(y-1)(z-1)S(1,0,0)+(x-1)y(z-1)S(0,1,0)+(x-1)(y-1)zS(0,0,1)+$$
$$(x-1)yzS(0,1,1)+x(y-1)zS(1,0,1)+xy(z-1)S(1,1,0)+$$
$$xyzS(1,1,1)$$

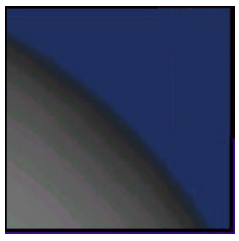


Abbildung 3.7: Trilineare Interpolation

Die Qualität der so erhaltenen Volumsdaten muss jedoch noch verbessert werden. In den Querschnittbildern treten folgende Probleme auf:

- Oft unterliegen die Bilder einem Rauschen. Solche Effekte lassen sich

mit Hilfe von Faltung mit Filterfunktionen unterdrücken, wie in Kapitel 4.1.4 beschrieben wird. Im allgemeinen wird dadurch das Bild weicher, Details können jedoch verlorengehen.

- Ein weiteres Problem ist, dass die Bilder häufig verschieden stark belichtet sind. Helligkeit und Kontrast der Bilder müssen angepasst werden.
- Die Datensätze sind häufig sehr groß und die Visualisierung daher sehr rechenintensiv. Es muss daher der Bereich ausgewählt werden, der tatsächlich relevant ist. Solche einfachen Auswahlverfahren erhöhen die Geschwindigkeit der Visualisierung enorm.

Unter das Aufbereiten der Daten fällt weiters auch das Berechnen von Werten, die später in der Visualierungs-Pipeline benötigt werden. Darunter fällt beispielsweise das Berechnen von Gradienten. Den Gradient benötigt man für die Berechnung von realistischer Beleuchtung. Das verstärkt den dreidimensionalen Eindruck der Ergebnisbilder, was speziell bei der Visualisierung von feinen Strukturen wie Gefäßsystemen wichtig ist.

Gradient

$$\nabla f(x,y,z)=\left(\frac{1}{2}(f(x+1,y,z)-f(x-1,y,z)),\frac{1}{2}(f(x,y+1,z)-f(x,y-1,z)),\frac{1}{2}(f(x,y,z+1)-f(x,y,z-1))\right)$$

Ist der Datensatz freilich sehr groß, kann es ein Problem sein, die Gradienten zu speichern. Denn speichert man zu jedem Voxel $|\nabla f(\underline{x})|$, braucht man genauso viel Speicher dafür wie für den Datensatz selbst. Berechnet man hingegen die Gradienten erst beim Rendern, muss ein großer Geschwindigkeitsverlust in Kauf genommen werden.

3.1.3 Daten auf die Geometrie abbilden

Moderne Graphikkarten können eine riesige Anzahl von Dreiecken darstellen. Ein Ansatz ist daher, aus den Daten Isoflächen zu berechnen. Der Schwellwert ist dabei die Intensität des Organs, die dargestellt werden sollen. Die Isofläche wird dann durch Dreiecke angenähert, die im letzten Schritt gezeichnet werden. Ein

einfacher, schneller Algorithmus, der das leistet, ist der Marching Cubes Algorithmus, siehe dazu [9].

In dieser Arbeit wird näher auf die direkte Volumsvisualisierung (*DVR*) eingegangen. *DVR* zeichnet sich dadurch aus, dass das Volumen direkt gerendert wird, ohne dass es vorher auf eine andere Geometrie abgebildet wird. Vor dem Rendern wird lediglich jedem Voxel mit einer Transferfunktion eine Farbe und eine Transparenz zugewiesen.

$$TF: \; f(\underline{x}) \rightarrow (r, g, b, \alpha)$$

Transferfunktionen haben eine zentrale Bedeutung. Schon mit einfachsten Methoden wie dem Slicing, wie in Kapitel 3.2.1 beschrieben, kann mit guten Transferfunktionen viel Information aus den Bildern gewonnen werden. Die Wahl einer guten Transferfunktion ist jedoch nicht leicht. In der Praxis werden Transferfunktionen durch Probieren gewonnen, oder wie in [10] beschrieben semiautomatisch erzeugt.

Wurden die Gradienten des Volumens berechnet, kann auch eine zweidimensionale Transferfunktion verwendet werden, die $|\nabla f(\underline{x})|$ berücksichtigt. Auf diese Weise kann man Bereichen, die starken Änderungen unterliegen, also wo $|\nabla f(\underline{x})|$ groß ist, weniger Transparenz zuordnen und sie dadurch besser sichtbar machen. Bereiche, in denen weniger Änderungen stattfinden, die daher weniger interessant sind, kann man eine höhere Transparenz zuteilen.

$$TF: \; (f(\underline{x}), |\nabla f(\underline{x})|) \rightarrow (r, g, b, \alpha)$$

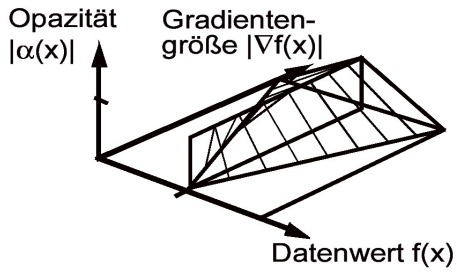


Abbildung 3.8: Zweidimensionale Transferfunktion

3.1.4 Das Rendern der Geometrie

In diesem Schritt muss die dreidimensionale Geometrie auf die Bildebene abgebildet werden. Es soll hier auf *DVR* Verfahren eingegangen werden, das heißt, es liegen die Daten in einem regulären Gitter vor, jedem Voxel wird mit einer Transferfunktion Farbe und Transparenz zugewiesen.

Man unterscheidet zwischen zwei Ansätzen, dem *Object Order* und dem *Image Order Approach*.
Beim *Object Order Aproach* wird jeder Voxel auf die Bildebene abgebildet. Der Aufwand dieser Verfahren ist demnach abhängig von der Anzahl der Voxel. Ein typisches Beispiel ist das Shear Warp Verfahren, das in Kapitel 3.2.3 beschrieben wird, oder das Splattering Verfahren.
Im Gegensatz dazu wird im *Image Order Approach* jeder Pixel der Bildebene berechnet. Der Aufwand hängt daher von der Größe des Bildes ab und der Zeit, die die Berechnung eines Pixels braucht. Raycasting, auf das in Kaptiel 3.2.2 näher eingegangen wird, und Raytracing sind Methoden dieser Art.

3.2 Direkte Volumsvisualisierung

Unter direkter Volumsvisualisierung versteht man, dass die Voxel, die in einem regulären Gitter vorliegen, direkt gerendert werden, ohne vorher auf eine andere Geometrie abgebildet zu werden. Im folgendem wird kurz auf *Slicing*

eingegangen, dem einfachsten Verfahren die Volumsdaten zu untersuchen. Danach wird Raycasting und Shear Warp Factorization besprochen und als Anwendung die häufigsten Techniken wie First Hit Rendering, Maximum Intensity Projection, sowie das Emissions- und Absorptionsmodell vorgestellt.

3.2.1 Slicing

Slicing ist eine sehr einfache Methode, um einen Einblick in die Volumsdaten zu erhalten. Durch die Volumsdaten wird dabei orthogonal zu einer Hauptachse eine Ebene gelegt. Diese Ebene kann man entlang der Achse durch das Volumen bewegen. Normalerweise wird das Bild dabei in Graustufen dargestellt, es kann aber auch eine Transferfunktion verwendet werden. Werden mehrer Ebenen verwendet, spricht man auch von Multiplanar Reconstruction (*MPR*).
Oft will man nur einen bestimmten Bereich von Intensitäten genauer untersuchen. Das erreicht man durch eine Technik, die Windowing genannt wird. Man verwendet dabei eine Transferfunktion, die allen Intensitäten, die kleiner als eine untere Schranke sind, die Farbe schwarz zuweist. Allen Intensitäten, die größer als eine obere Schranke sind, wird die Farbe weiß zugewiesen. Zwischen unterer Schranke und oberer Schranke wird linear interpoliert. Durch diese Methode wird nur der Bereich, der von Interesse ist, in Graustufen dargestellt.

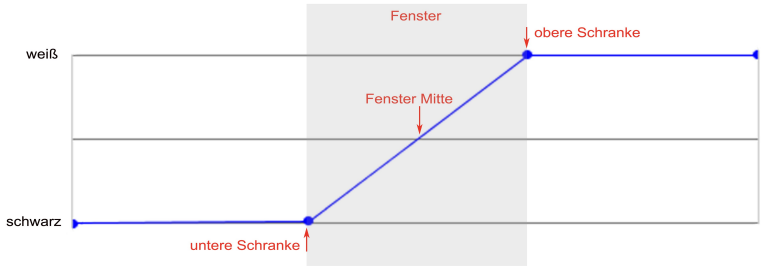


Abbildung 3.9: Windowing

Durch Slicing kann man die gegeben Daten sehr genau untersuchen. Das Problem ist jedoch, dass immer nur eine Ebene sichtbar ist. In großen Datensätzen müsste man hunderte Ebenen betrachten, bis man einen hinreichenden Einblick in die Daten erhält.

3.2.2 Raycasting

Für jeden Pixel der Bildebene wird ein Strahl durch das Volumen geschickt. Der Startpunkt und die Richtung der Strahlen hängen dabei vom Stand- und Blickpunkt des Betrachters ab und davon, ob Parallelprojektion oder perspektivische Projektion verwendet wird.
Der Farbwert des Pixels wird aus allen Punkten, die der Strahl passiert, berechnet. Es gibt verschiedene Ansätze, welche Punkte am Strahl zur Berechnung des Pixels dienen:

1. Der Voxelwert wird in regelmäßigen Abständen gemessen
2. Alle getroffen Voxel werden genau einmal verwendet
3. Die Strecke, die der Strahl im Voxel zurücklegt, wird berücksichtigt

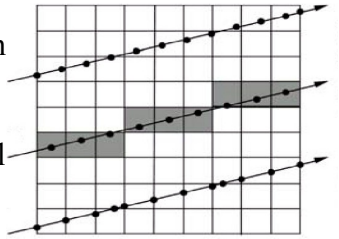


Abbildung 3.10: Verschieden Auswahlverfahren

Es wird weiters unterschieden, in welcher Richtung sich der Strahl bewegt. Der Strahl kann entweder von der Bildebene weggeschickt werden, oder sich zur Bildebene hinbewegen. Der einzige Unterschied besteht in der Reihenfolge, in denen die Voxelwerte gemessen werden. Für die meisten Verfahren liefern beide Methoden dieselben Ergebnisse.
Wie aus den gemessenen Voxelwerten und der Transferfunktion ein Pixelwert berechnet wird, ist vom gewünschten Verfahren abhängig, diese werden in den Kaptieln 3.2.4 – 3.2.6 behandelt.

3.2.3 Shear-Warp Factorization

Wenn die Strahlen orthogonal zu einer Hauptebene auf das Volumen treffen, kann jeder Voxel genau einmal zur Berechnung verwendet werden. Dabei ist keine Interpolation im Datenraum notwendig. Die Anzahl der Pixel stimmt mit der Anzahl der Voxel in der Hauptebene überein.

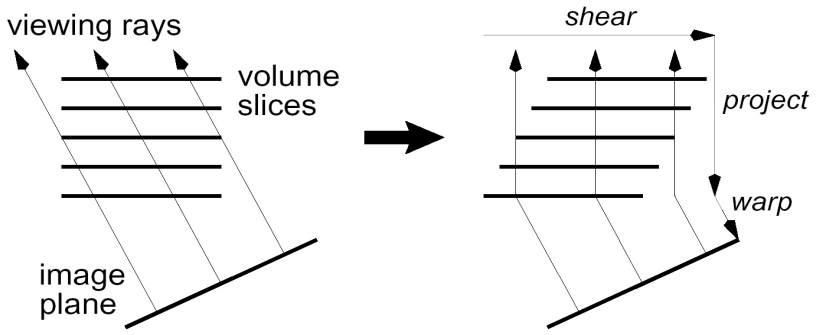


Abbildung 3.11: Die Strahlen treffen orthogonal auf die Querschnitte

Die Idee von Shear- Warp ist es nun, dass die Voxelebenen, aus denen das Objekt besteht, so lange in X-Y Richtung geschert werden, bis die Sichtstrahlen parallel zur dritten Koordinatenachse verlaufen. Das so berechnete Bild ist verzerrt, erst durch eine Transformation im Bildraum erhält man das fertige Bild.

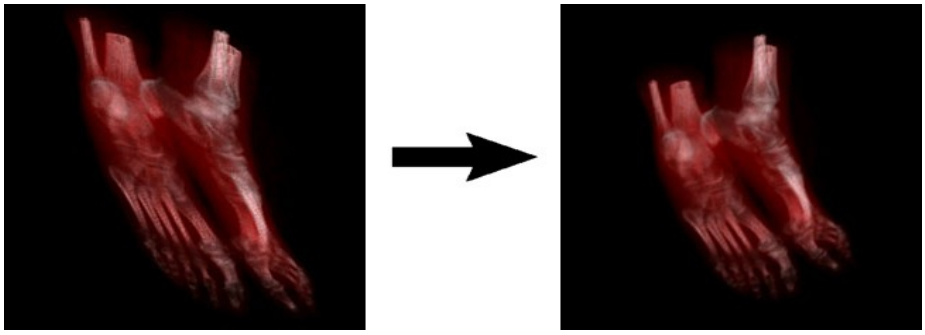


Abbildung 3.12: Das gescherte Bild wird durch Warp - Transformation entzerrt

Der Vorteil dieses Verfahren besteht darin, dass es sehr schnell ist, da jeder Voxel des Volumens genau einmal verwendet wird. Liegen die Voxelebenen lauflängencodiert vor, kann eine weitere Geschwindigkeitsverbesserung erzielt werden. Vollkommen transparente Voxel werden dabei zusammengefasst und können bei der Berechnung der Pixelfarbe einfach übersprungen werden.

Das Verfahren ist schneller als Ray Casting, die Qualität ist jedoch niedriger, denn die Auflösung des Bildes hängt von der Anzahl der Voxel im Volumen ab. Will man das Bild vergrößern, geschieht das durch bilineare Interpolation im Bildraum. Im Gegensatz dazu wird beim Raycasting trilineare Interpolation im Volumen verwendet, was speziell bei kleineren Datensätzen bessere Ergebnisse liefert. Das Shear- Warp Verfahren wurde in [11] vorgestellt.

3.2.4 First Hit

Mit dem First Hit Verfahren kann man sehr gut erkennen, wie genau sich der Arterienbaum segmentieren lässt. Weiters kann das Ergebnis der Segmentierung visuell überprüft werden, indem die visualisierten Gefäße mit dem graphentheoretischen Röhrenmodell verglichen wird.

Abbildung 3.13: First Hit Projektion von Arterien

Im First Hit Verfahren werden die Strahlen von der Bildebene weg durch das Volumen geschickt. Der erste Voxelwert, der größer als ein fix vorgegebener Grenzwert ist, dient zur Berechnung des Pixels.

Das einfachste Verfahren, dem Pixel eine Farbe zuzuordnen, ist, die Intensität des Pixels von der Entfernung des Voxels zur Bildebene abhängig zu machen. Dieses Verfahren nennt man z-Shading. Es entsteht jedoch häufig ein falscher räumlicher Eindruck, da keine Gradienten verwendet werden.

Ein komplexeres Modell, das den Gradienten des Voxels einbezieht, ist das Phong- Modell. Der Voxel wird dabei von einer parallelen Lichtquelle beleuchtet. Die Farbe des Voxels ergibt sich aus:

$$c_\lambda(x_i) = c_{p,\lambda} k_{\alpha,\lambda} + \frac{c_{p,\lambda}}{k_1 + k_2 d(x_i)} \left(k_{\delta,\lambda} (N(x_i).L) + k_{\sigma,\lambda} \left(N(x_i). \frac{V+L}{|V+L|} \right)^n \right)$$

Größe	**Beschreibung**
$c_\lambda(x_i)$	Die Farbkomponente für $\lambda = r, g, b$ an der Stelle x_i
$c_{p,\lambda}$	Die Farbkomponenten der parallelen Lichtquelle
$k_{\alpha,\lambda}, k_{\delta,\lambda}, k_{\sigma,\lambda}$	Die ambiente α, difusse δ und spekuläre σ Reflektion der einzelnen Farbkomponenten
n	Der Exponent, um den Glanzpunkt zu approximieren
k_1, k_2	Das sogenannte depth-cueing wird mit diesen Konstanten linear approximiert.
$d(x_i)$	Entfernung zu Bildebene
$N(x_i)$	Normalisierter Gradient an der Stelle x_i
L	Normalvektor in Richtung Lichtquelle
V	Normalvektor in Richtung Betrachter

Tabelle 3.1: Die verwendeten Größen im Phong Shading

3.2.5 Maximum Intensity Projection (MIP)

In diesem Verfahren wird zur Berechnung des Pixelwerts der größte Voxelwert entlang eines Strahls verwendet. Der Farbwert des Voxels ergibt sich ausschließlich aus der Transferfunktion.

Will man Bereiche mit niedriger Intensität sehen, die normalerweise nicht gezeichnet werden, kann man dazu das α der Transferfunktion verwenden. Es wird dann nicht der größte Voxelwert entlang des Strahls, sondern der größte α Wert verwendet. Auf diese Weise lassen sich mit *MIP* auch weiche Strukturen veranschaulichen.

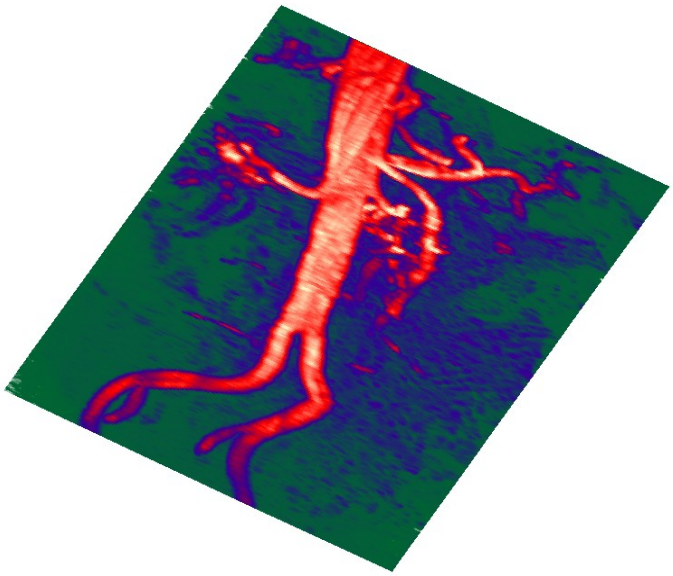


Abbildung 3.14: MIP eines Arterienbaums

Das Verfahren eignet sich sehr gut, um Helligkeit und Kontrast im Volumen zu veranschaulichen. Stammen die Volumsdaten aus MR- oder CT- Bildern, sind die einzelnen Ebenen häufig verschieden belichtet. In MIP Bildern können Helligkeit und Kontrast in ausgewählten Bereichen angepasst werden.

3.2.6 Das Emissions– und Absorptionsmodell

Bei den vorangegangenen Verfahren hat die Transparenz keine Rolle gespielt. Bei dem kommenden Verfahren werden alle Voxel mit ihrer zugeordneten Transparenz entlang eines Strahls verwendet. Beim Emissionsverfahren werden die Voxel als Leuchtpartikel angesehen und geben je nach ihrer Transparenz Helligkeit ab. Im Absorptionsverfahren verdunkeln die Voxel abhängig von der Transparenz die Helligkeit. Werden Emissions- und Absorptionsmodell kombiniert, ergeben sich mit einer geeigneten Transferfunktion realistische halbtransparente Darstellungen von Volumsdaten.

Im kontinuierlichen Emissionsmodell wird zunächst überlegt, wie viel Licht auf einem Abschnitt des Strahls hinzugefügt wird.
Der Ansatz

$$\frac{dI}{dt} = g(t)$$

führt durch Integration zu

$$I(s) = I_0 + \int_0^s g(t)\,dt$$

Dabei bezeichnet *I* die Lichtintensität. Die Funktion $g(t)$ wird *Glow Factor* genannt und im diskreten Emissionsmodell von der Transparenz der Voxel abhängig gemacht. Das diskrete Emissionsmodell ergibt sich aus dem Ansatz

$$out_{i+1} = out_i + c_i \quad ,$$

wobei c_i den Beitrag des Voxels auf dem entsprechenden Stück des Strahls bezeichnet. Es führt zu der expliziten Darstellung:

$$pixel = hintergrund + \sum_{i=0}^{N} c_i$$

Abbildung 3.15: Wolke aus Leuchtpartikeln

Im kontinuierlichen Absorptionsmodell wird betrachtet, wie viel Licht auf einem Abschnitt des Strahls absorbiert wird.

$$\frac{dI}{dt} = -\tau(t)\,I(t)$$

führt duch Integration zu

$$I(s) = I_0\, e^{-\int_0^s \tau(t)\,dt}$$

Die Funktion $\tau(t)$ wird Auslöschungskoeffizient genannt und hängt wiederum von der Transparenz der Voxel ab.

Das diskrete Absorptionsmodell ist daher durch

$$out_{i+1} = out_i(1 - \alpha_i)$$

gegeben, wobei die α_i Opazität des Voxels bezeichnet. Die explizite Lösung ist

$$pixel = (hintergrund) \prod_{i=0}^{N} (1 - \alpha_i)$$

Abbildung 3.16: Dunkle Wolke

Kombiniert man die beiden diskreten Modelle, erhält man aus dem Ansatz

$$out_{i+1} = out_i (1 - \alpha_i) + c_i \alpha_i$$

das Ergebnis

$$pixel = (hintergrund) \prod_{i=0}^{N} (1 - \alpha_i) + \sum_{i=0}^{N} \left(c_i \alpha_i \prod_{j=i}^{N} (1 - \alpha_j) \right)$$

Abbildung 3.17: Luminiszierende Wolke

Die Pixelwerte können noch wie in Kapitel 3.2.4 beschrieben einem Phong-Shading unterzogen werden. Um bessere Qualität zu erzielen, kann zusätzlich der Schatten des Volumens auf sich selbst berücksichtigt werden. Solche Effekte können mit einfacher oder mehrfacher Streuung erzielt werden. Verschiedene relativ rechenintensiven Methoden, die das leisten, werden in [12] erklärt.

3.3 Visualisierung als Werkzeug der Segmentierung

Die in diesen Kapitel vorgestellten Verfahren ermöglichen die Exploration der Gefäßsysteme in den Volumsdaten, die aus tomographischen Aufnahmen erzeugt werden. Im folgenden Kapitel sollen Möglichkeiten aufgezeigt werden, die Arterien automatisch zu erkennen und daraus ein graphentheoretisches Modell des Arterienbaumes zu erzeugen. Das geschieht in mehreren Schritten. Die Visualisierung der Daten dient dazu als Werkzeug und ermöglicht eine Kontrolle der Ergebnisse der Segmentierung in jedem Teilschritt.

Dazu wird vor allem

- *Slicing* zum navigieren in den Volumsdaten,
- *MIP* um Helligkeit und Kontrast anzupassen,
- *First Hit* um die Qualität der Segmentierung im voraus einschätzen zu können und zur visuellen Kontrolle des Endergebnisses

verwendet.

Kapitel 4
Segmentierung

Segmentierung bedeutet, dass in den Querschnittbildern Organe und Gefäße erkannt und klassifiziert werden. Dabei wird in den Bilder jeder Pixel oder in Volumsdaten jeder Voxel einem Organ oder Gefäß zugeordnet. Durch dieses so genannte Labeling können in der Visualisierung für verschiedene Organe verschiedene Transferfunktionen verwendet werden. Organe, die nicht von Interesse sind, können ausgeblendet werden. Ebenso ist es möglich, einzelne Bereiche genauer zu untersuchen. In dieser Arbeit wird versucht, durch Segmentierung geometrische Informationen über die Arterien zu erhalten. Diese Daten sollen für die dynamische Herzkreislaufsimulation verwendet werden.

4.1 Allgemeine Methoden

In diesem Kapitel wird auf allgemeine Methoden eingegangen, die zur Segmentierung verwendet werden, ohne dass spezielle Eigenschaften des Organs, das segmentiert werden soll, berücksichtigt werden. Die Methoden stammen aus der Bildverarbeitung, siehe [14], und dienen aus Ausgangspunkt für die Verfahren zur Segmentierung von Arterien.

4.1.1 Bereich ausschneiden

Zuerst muss der Bereich ausgewählt werden, in dem sich das Gefäß befindet.

Dadurch werden die Volumsdaten kleiner, und die Gefahr sinkt, dass falsche Gefäße erkannt werden. Bezeichnet R den interessanten Bereich, liefert die folgende Filterfunktion das gewünschte.

$$I_0(x) = \begin{cases} I(x) & wenn \quad x \in R \\ 0 & sonst \end{cases}$$

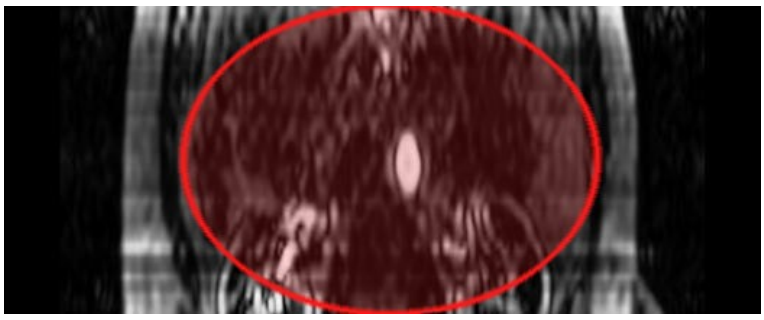


Abbildung 4.1: Der für die Segmentierung interessante Bereich muss ausgeschnitten werden

4.1.2 Helligkeit und Kontrast

Durch das Ändern der Helligkeit ändern sich alle Intensitäten im selben Maße. Die Intensitäten werden um eine fixe Größe verschoben. Im Gegensatz dazu werden beim Einstellen des Kontrasts die Unterschiede zwischen hellen und dunklen Pixeln verstärkt. Definiert man

$$\theta = (I_1(x) - \frac{1}{2})\pi$$

wobei $I_1(x)$ die Intensität and der Stelle *x* bezeichnet, dann berechnet sich die neue Intensität aus

$$I_k(x) = I_1(x) + 1 \text{over} 2 \, sign\left(\frac{\sin(\theta)+1}{2} - I_1(x)\right)$$

Diese Werte werden eingeschränkt auf das Intervall von 0 bis 1.

$$I_2(x) = \begin{cases} 1 & wenn \quad I_k(x) > 1 \\ 0 & wenn \quad I_k(x) < 1 \\ I_k(x) & sonst \end{cases}$$

Ist *sign* dabei ein positiver Wert, wird das Bild geschärft, ist *sign* negativ, wird das Bild weicher.

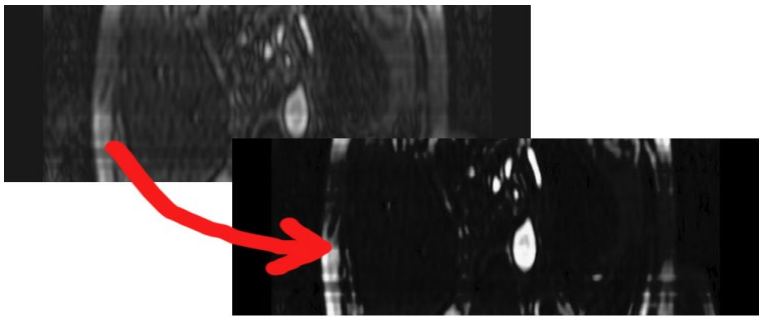


Abbildung 4.2: Durch stärkeren Kontrast sind die Arterien besser erkennbar.

In MR – Bildern sind die Querschnitte im oberen und unteren Bereich meist zu dunkel und auch der Kontrast ist zu schwach. Normalerweise sollte man zuerst die Helligkeit einstellen, schwarze Flächen erscheinen dann grau. Danach sollte der Kontrast eingestellt werden, um die grauen Flächen wieder abzudunkeln.

4.1.3 Binär entscheiden

Im nächsten Schritt wird binär entschieden, ob der Pixel oder Voxel zum Gefäß gehört oder nicht. Dabei wird ein Intervall von Intensitäten angegeben. Liegt der Pixel oder Voxel im Intervall, bekommt er den Wert 1, wenn er zum Gefäß gehört, sonst den Wert 0.

$$I_3(x) = \begin{cases} 1 & wenn \;\; I_2(x) \in [I_{min}, I_{max}] \\ 0 & sonst \end{cases}$$

Das Volumen muss für diesen Filter nicht verändert werden, es muss nur eine Transferfunktion gewählt werden, die das leistet.

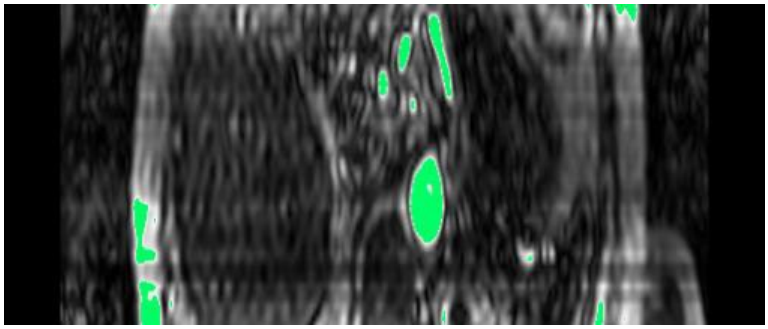


Abbildung 4.3: Es wird binär entschieden, ob ein Voxel eine Arterie ist

4.1.4 Filter

Der Sinn von Filtern ist, die Qualität der Volumsdaten zu verbessern. Oft unterliegen die Daten einem starken Rauschen. Es wird versucht, aus den gemessenen Daten die tatsächlichen Intensitäten zu rekonstruieren.

Ein statistischer Filter überprüft die 3 x 3 x 3 Nachbarschaft jedes Voxels. Unterscheiden sich zu viele Nachbarn im Wert von dem Voxel, wird der Wert des Voxels geändert. Sind also einzelne Voxel im Gefäß falsch erkannt, wird das korrigiert. So kann regelmäßiges Rauschen aus den Volumsdaten eliminiert werden.

$$n = \sum_{y \in N(x)} I_3(y) \neq I_3(x)$$

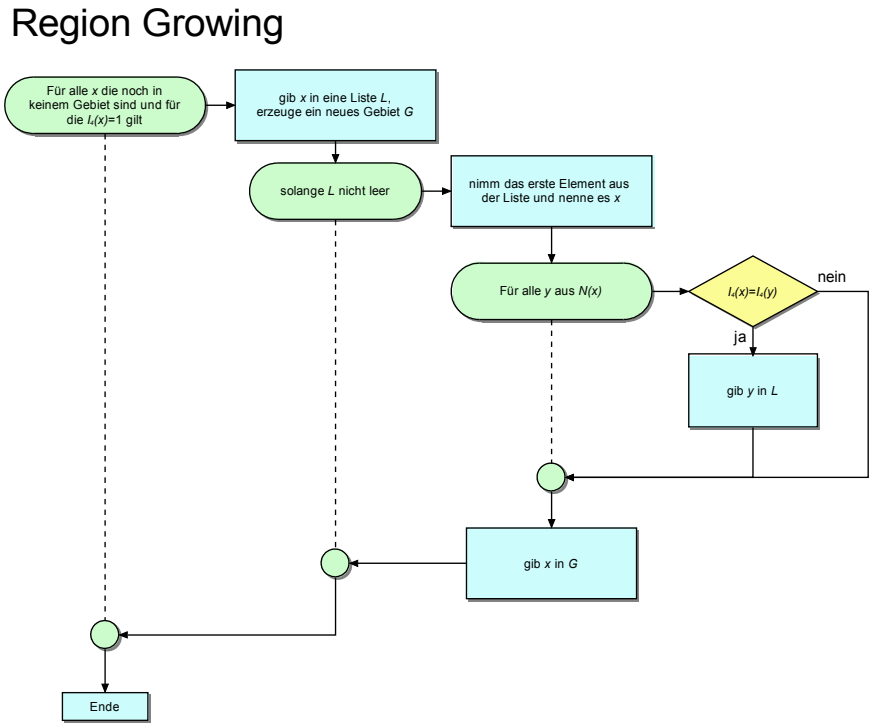
Der Wert n gibt daher an, wieviele Voxel aus der Nachbarschaft sich unterscheiden.

$$I_4(x) = \begin{cases} 1 & wenn \quad n \geq min \\ 0 & wenn \quad n \leq min \end{cases}$$

Der Wert *min* gibt an, wie viele Nachbarn sich von dem Voxel, der überprüft wird, unterscheiden dürfen. Normalerweise wird *min* auf 9 gesetzt.

4.1.5 Region Growing

Der Algorithmus ordnet den in den letzten Schritten klassifizierten Voxeln zusammenhängende Gebiete zu. Jeder Voxel wird dabei mit einer Nummer versehen, die angibt, zu welchem Gebiet er gehört.



Je nachdem, ob man in einem Bild oder in einem Volumen arbeitet, bedeutet *N(x)* die benachbarten vier Pixel im Bild oder die benachbarten sechs Voxel im Volumen.

4.1.6 Konkave Bereiche ausfüllen

Häufig werden von Gefäßen nur die Ränder richtig erkannt und im Inneren gibt es Löcher, die falsch erkannt werden. Es soll nun versucht werden, diese Löcher zu schließen.

Der folgende Operator füllt das Innere von konkaven Gebieten. Abhängig davon, wie oft er angewandt wird, können Löcher beliebiger Größe ausgefüllt werden.

Der Operator verwendet dazu die Definition von lokaler Konkavität. Für ein Bild wird die 3x3 Nachbarschaft von *x* mit 8 verschiedenen Konfigurationen C_i angegeben, wie in Abbildung 4.4 ersichtlich.

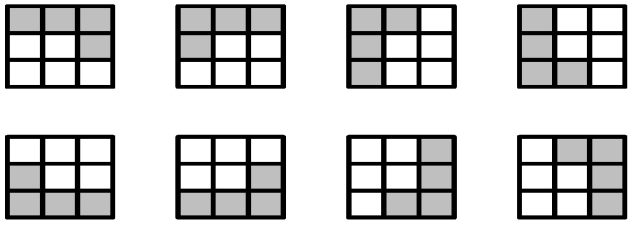


Abbildung 4.4: Die verschiedenen Konfigurationen lokaler Konkavität

Nun wird die Nachbarschaft *N(x)* mit den Konfigurationen C_i verglichen. Entspricht *N(x)* einem C_i , wird der Pixel gesetzt. Bei der Anwendung des Operators muss vorsichtig vorgegangen werden, wird er zu oft iteriert, wird das ganze Bild ausgefüllt.

$$I_5(x) = \begin{cases} 1 & wenn \;\; C_i(x) = N(x) \\ I_4(x) & sonst \end{cases}$$

Sollen in einem Volumen konkave Bereiche gefüllt werden, wird der Operator in jeder Ebene angewandt.

4.1.7 Der Watershed Algorithmus

Der Watershed Algorithmus ist ein häufig verwendeter Algorithmus in der Bildverarbeitung und Bilderkennung. Er wird benutzt, um Konturen zu erkennen, und wird oft zu Segmentierung einzelner Ebenen benutzt. Watershed wurde zuerst in [15] vorgestellt.

Ein Graustufenbild wird als topographisches Relief betrachtet. Die Intensitäten der Pixel werden dabei als Höhen im Relief interpretiert. Schwarze Bereiche werden so als Täler dargestellt, helle Bereiche als Berge.

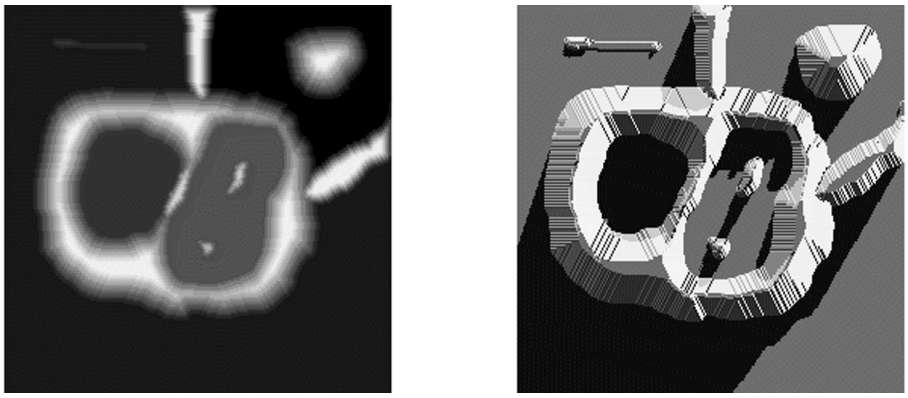


Abbildung 4.5: Das Graustufenbild wird als topographisches Relief interpretiert

Die Idee ist nun, die so erhaltene Fläche unter Wasser zu setzen. Dazu fließen von verschiedenen fixen Punkten verschiedene Flüssigkeiten auf die Fläche. Normalerweise sind diese Punkte die Minima der Fläche. Die Flüssigkeiten haben dabei die Eigenschaft, zusammenhängend und homogen zu sein. Jeder Pixel wird also höchstens von einer Flüssigkeit überdeckt. Die Flüssigkeit sammelt sich zunächst in den Tälern und erschließt mit steigendem Wasserstand immer größere Bereiche. Treffen zwei Flüssigkeiten aufeinander, entsteht eine Grenze, die man sich als hohe Mauer vorstellen kann. Die Flüssigkeiten können daher nicht übereinander fließen, auch wenn sie verschieden hohen Wasserstand haben. Ist die gesamte Fläche bedeckt, werden die Flüssigkeiten als Gebiete im ursprünglichen Bild interpretiert, die Grenzen stellen die Konturen des Bildes dar.

Der Algorithmus erzeugt häufig übersegmentierte Bilder. Um das zu vermeiden, sollte man die Bilder mit einem geeigneten Filter weich zeichnen. Eine andere Möglichkeit ist durch geschickte Wahl der Einflusspunkte die Anzahl der nötigen Flüssigkeiten klein zu halten. Weiters können auch im nachhinein Flüssigkeiten mit ähnlichem Wasserstand vereint werden.

4.2 Segmentierung von Arterien

In diesem Kapitel soll ein einfacher zuverlässiger Algorithmus zur Segmentierung

von Arterienbäumen vorgestellt werden. Dabei wird weniger Wert auf die Segmentierung der genauen Geometrie gelegt, sondern es sollen Eigenschaften, wie Länge, Radius und Konnektivität, die für die dynamische Herzkreislauf-simulation von Bedeutung sind, gewonnen werden. Die vorangegangen Verfahren werden dabei benützt.

4.2.1 Annahmen

Ziel ist es, aus den Volumsdaten ein graphentheoretisches Modell, welches den Arterienbaum repräsentiert, zu gewinnen, in dem mit dem dynamischen Kerzkreislaufmodell gerechnet werden kann. Dieses Ziel unterscheidet sich stark von der klassischen Segmentierung, da dieser Algorithmus weder an der Geometrie noch am Generieren von Flächen interessiert ist. Es wird vielmehr versucht, spezielle Informationen aus den Volumsdaten zu extrahieren, die für das dynamische Herz- Kreislauf- Modell benötigt werden. Das sind:

- Lage der Bifurkationen
- Länge der Arterie zwischen zwei Bifurkationen
- Gemittelte Dicke der Arterie zwischen zwei Bifurkationen
- Punkte der Mittellinie der Arterie zwischen zwei Bifurkationen

Die Mittelpunkte der Rohre werden für die Simulation nicht verwendet, man braucht sie jedoch, um den idealisierten Arterienbaum zu visualisieren und Ergebnisse der Simulation darin darzustellen. Verwendet man zu wenige Mittelpunkte, passiert es, dass die Länge der Arterie, mit der im Modell gerechnet wird, nichts mit der Länge der Arterie in der Visualisierung zu tun hat.

Bei der Segmentierung eines Arterienbaums spielen die Eigenschaften von Arterien eine wichtige Rolle:

- Die Arterien verzweigen sich nach unten hin immer stärker.
- Der Radius der Arterien wird kleiner, große Arterien können nicht

plötzlich enden.

- Arterien mit einem Durchmesser von weniger als 1 Millimeter sind für das dynamische Modell nicht von Bedeutung.
- Die meisten Arterien stehen senkrecht zur Bildebene, erscheinen daher als Ellipsen, tun sie das nicht werden sie im ersten Schritt dahingehend gedreht.

4.2.2 Aufbereitung der Daten

Die Daten liegen als MR Bilder vor und stellen Querschnittsbilder des gewünschten Bereichs dar. Häufig liegen nur wenige Bilder mit einem konstanten Abstand vor. Um die Bilder zu visualisieren und zu drehen, werden neue Schnitte zwischen den Bildern eingefügt. Diese werden durch lineare Interpolation erstellt. Der Stapel der Querschnittbilder wird nun als Volumen betrachtet.

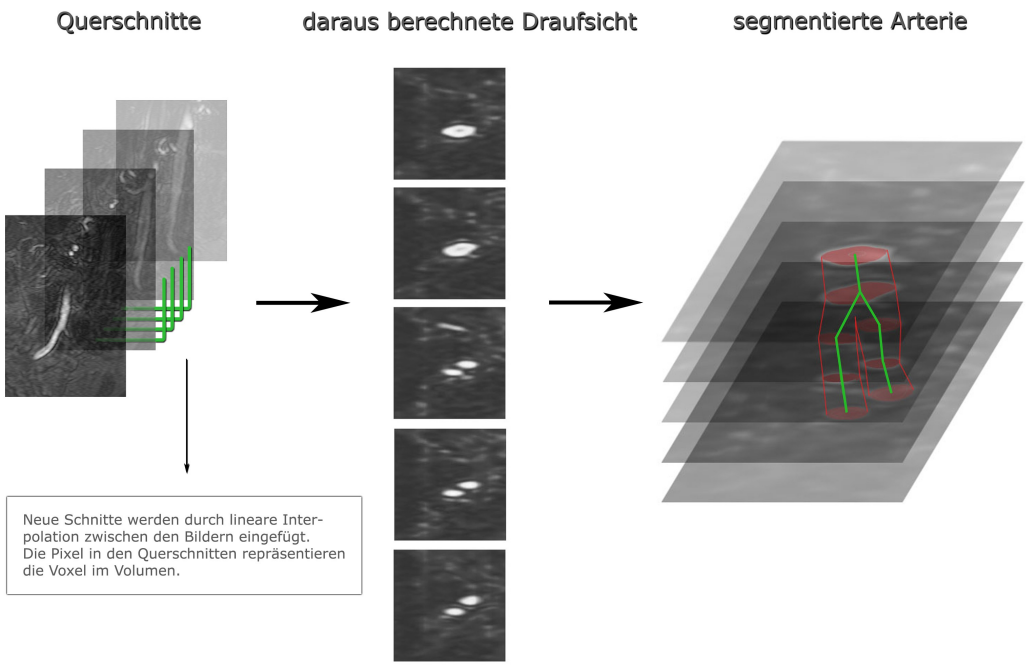


Abbildung 4.6: Das Umrechnen der Querschnittbilder

Das Volumen wird nun so gedreht, dass die meisten Arterien senkrecht zur Bildebene stehen. Das heißt, die meisten Arterien erscheinen als Ellipsen in den Ebenen.

Visualisiert man das erhaltene Volumen mit *MIP*, erkennt man, ob die Helligkeit und der Kontrast in den Querschnittsbildern in etwa gleich sind. Unterliegt das Volumen starken Schwankungen, müssen die Intensitäten des Volumens angepasst werden.

Durch eine First Hit Visualisierung kann eingeschätzt werden, wie gut eine Segmentierung möglich ist. Als Grenzwert dienen die Intensitäten, welche die Arterien im Volumen haben.

4.2.3 Finden der Ellipsen

Die Segmentierung der Arterien wird in zwei Arbeitsschritte geteilt. Als erstes wird in jeder Ebene versucht, die Arterien zu finden und als Ellipsen darzustellen. Im folgendem Kapitel wird behandelt, wie diese Ellipsen verbunden werden.

Das erkennen der Ellipsen in den Ebenen erfolgt in zwei Schritten.

- In jeder Ebene des Bildes wird wie in Kapitel 4.1.3 zuerst binär entschieden, ob die Voxel zur Arterie gehören oder nicht.
- Nun wird in jeder Ebene der in Kapitel 4.1.5 vorgestellte Region Growing Algorithmus angewandt. Für jedes gefundene Gebiet (G) werden zusätzlich der Mittelpunkt (m) und die Grenzen (R) gespeichert. Um eine Übersegmentierung zu vermeiden, werden zu kleine Gebiete, die also nur aus wenigen Voxeln bestehen, vernachlässigt.

Ein Problem ist, dass es sich nicht bei allen gefundenen Gebieten um Ellipsen handelt. Eine Vielzahl der Arterien steht zwar annähernd senkrecht auf das Bild, aber natürlich einige auch nicht. Speziell an Bifurkationen treten häufig Probleme auf, wie in Abbildung 4.7 ersichtlich ist. In Abbildung 4.8 handelt es sich bei (2) um eine Bifurkation, bei (1) hingegegen handelt es sich doch nur um eine stark gekrümmte Arterie. Im allgemeinen ist es schwierig, Schnitte von Bifurkationen von Schnitten durch stark gekrümmte Arterien zu unterscheiden.

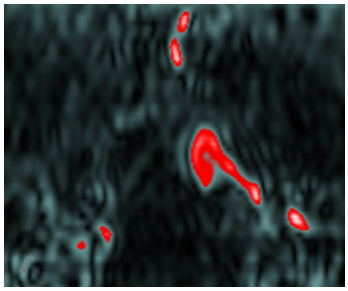


Abbildung 4.7: Bifurkation

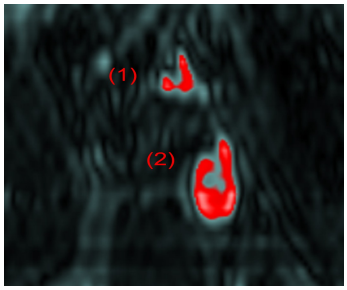


Abbildung 4.8

Um nun zu überprüfen, wie stark das Gebiet einer Ellipse ähnelt, werden alle Voxel innerhalb der Grenzen des Gebiets in die Ungleichung

$$e(v) := \left(\frac{x_1}{a}\right)^2 + \left(\frac{x_2}{b}\right)^2 \leq 1$$

mit

$$x_1 = (v_x - m_x) * \cos(\alpha) + (v_y - m_y) * \sin(\alpha)$$
$$x_2 = (v_x - m_x) * \sin(\alpha) - (v_y - m_y) * \cos(\alpha)$$

eingesetzt. Dabei bezeichnet *v* die Position des Voxels, *m* ist der Mittelpunkt der Ellipse, *a* und *b* die Hauptachsen und α der Drehungswinkel der Ellipse.

Wie stark das Gebiet (*G*) innerhalb der Grenzen (*R*) einer Ellipse ähnelt, berechnet sich aus

$$q(G, R) := \frac{|\{v \in G \ : \ e(v) \leq 1\}| + |\{v \in R \setminus G \ : \ e(v) > 1\}|}{|R|}$$

Das Ergebnis ist ein Wert im Intervall von 0 bis 1. Je stärker das gefundene Gebiet einer Ellipse ähnelt, desto größer ist der Wert. Ist *q(G,R)* größer als 0.7, handelt es sich üblicherweise um einen Schnitt durch ein Arterienstück, ist der Wert kleiner, handelt es sich um eine Bifurkation oder das Arterienstück liegt waagrecht zur Bildeben. Die Qualität der Ellipsen wird mitgespeichert und kann beim Verbinden der Ellipsen dazu verwendet werden, Bifurkationen besser zu erkennen.

Es soll über jedes gefundene Gebiet eine möglichst passende Ellipse gelegt

werden. Die Qualität q der Ellipse hängt von m, a, b, und α ab. Von diesen Werten ist nur der Mittelpunkt bekannt.

Man startet, indem man a als $R_x/2$ und b als $R_y/2$ annimmt. Die einzige unbekannte α wird nun durch ein einfaches Suchverfahren bestimmt, in dem q maximiert wird. Nun hält man α fest und verbessert q weiter, indem man zuerst a und dann b variiert.

Das Ausfüllen der konkaven Bereiche, wie im Kapitel 4.1.6 beschrieben, ist dafür meist nicht nötig. Denn wenn im Inneren des Arterienschnitts Teile nicht erkannt werden, wirkt sich das zwar auf q aus, jedoch wird die Ellipse immer noch richtig angenähert. In Abbildung 4.9 hat der Schnitt durch die Aorta mit q = 0.73 laut *q(G,R)* zwar am wenigsten Ähnlichkeit mit einer Ellipse, die Ellipse wurde jedoch richtig in den Schnitt gelegt, ohne die schlecht erkannten konkaven Bereiche auszufüllen.

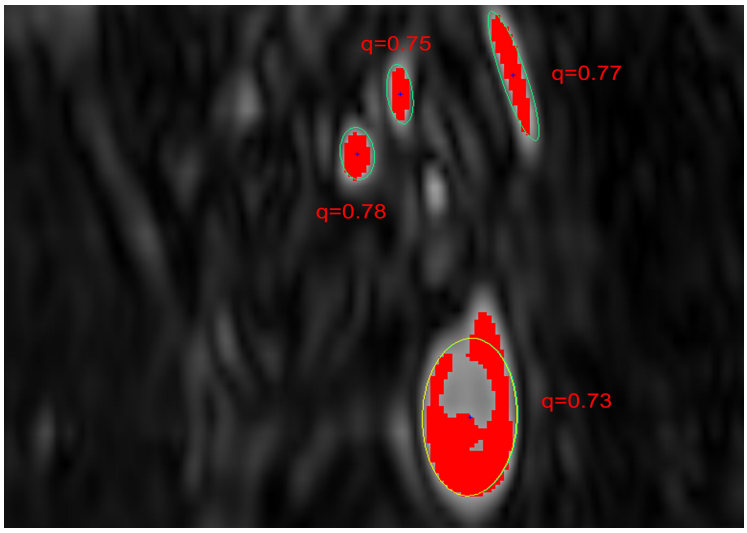


Abbildung 4.9: Erkannte Ellipsen in einem Querschnitt

Bevor die erkannten Ellipsen verbunden werden, müssen falsch erkannte Ellipsen noch entfernt oder modifiziert werden. Es muss also möglich sein, manuell die Ellipsen zu verändern, zu löschen oder neue einzufügen.

4.2.4 Arterienbaum finden

Aus den im letzten Kapitel erhaltenen Ellipsen soll der Arterienbaum aufgebaut werden. Es wird dabei versucht, die Annahmen in Kapitel 4.2.1 über die Eigenschaften von Arterien bestmöglich zu nutzen. Ausgegangen wird dabei von einer ausgezeichneten Ellipse die den Startknoten des Baumes darstellt.

Um die Arterien zuverlässig zu erkennen, können folgende Informationen von den gefundenen Ellipsen verwendet werden:

- Mittelpunkte der Arterien
- Die Hauptachsen a und b der Ellipsen und damit die Fläche der Ellipse
- Der Drehwinkel der Ellipsen
- Die Qualität *q(G,R)* der Ellipse und damit die Wahrscheinlichkeit, dass es sich um eine Bifurkation handelt.
- Der geschätzte neue Mittelpunkt

Aus der Annahme, dass die Arterie nicht plötzlich endet, kann mit den schon verbundenen Ellipsen eine Voraussage getroffen werden, wo die nächste Ellipse liegen soll. Dazu betrachtet man die schon verbundenen Ellipsen und schätzt aus deren Mittelpunkten den nächsten. Je nachdem, wie weit die Ebenen auseinander liegen, werden dabei weniger oder mehr vorangegangene Ellipsen betrachtet.
Im einfachsten Fall werden die letzten zwei Mittelpunkte betrachtet mit einer Geraden verbunden. Der geschätzte Mittelpunkt ist der Schnittpunkt der Gerade mit der nächsten Ebene.
Werden mehrere Mittelpunkte verwendet wird wiederum eine Gerade durch den letzten Mittelpunkt gelegt und ihr Abstand zu den übrigen Mittelpunkten minimiert. Der Schnittpunkt dieser Geraden mit der nächsten Ebene ist der neue Schnittpunkt.
Häufig geschieht es, dass einzelnen oder mehreren Ebenen Ellipsen fehlen, da die ursprünglichen *MR* - Bilder unterschiedlich belichtet waren. Daher wird der vorausberechnete Mittelpunkt keine Ellipse treffen, es muss die Gerade mit weiter darunter liegenden Ebenen geschnitten werden. Ist dabei die Ellipsenfläche groß,

können mehrere Ebenen übersprungen werden, weil man weiß, dass die Arterie nicht abrupt endet. Ist die Fläche klein sollen weniger oder keine Ebenen übersprungen werden.

Die Gefäßverfolgung wird von zwei Werten c_1 und c_2 abhängig gemacht. Der Wert c_1 beschreibt die maximale erlaubte Abweichung des Mittelpunkts zum geschätzten Mittelpunkt innerhalb einer Arterie. Der Wert c_2 beschreibt die maximal erlaubte Distanz zwischen zwei Ellipsen an einer Bifurkation. Der Wert c_1 sollte deshalb kleiner als c_2 gewählt werden.

Zunächst soll erklärt werden, wie eine einzelne Arterie gefunden wird. Anfangs besteht die Arterie aus einer Ellipse, die den Startknoten des Baumes dartstellt. Überschneidet sich die letzte verbundene Ellipse der Arterie mit einer Ellipse in der Ebene darunter, in der gleichen Ebene, oder der Ebene darüber, wird sie auf jeden Fall mit der Arterie verbunden. Gibt es mehrer solche Ellipsen, wird die flächengrößte Ellipse gewählt. Gibt es keine solchen Ellipsen mehr wird eine Ellipse mit weniger Distanz als c_1 zum geschätzten Mittelpunkt gesucht. Dieser Vorgang wird solange wiederholt, bis keine geeigneten Ellipsen mehr gefunden werden.

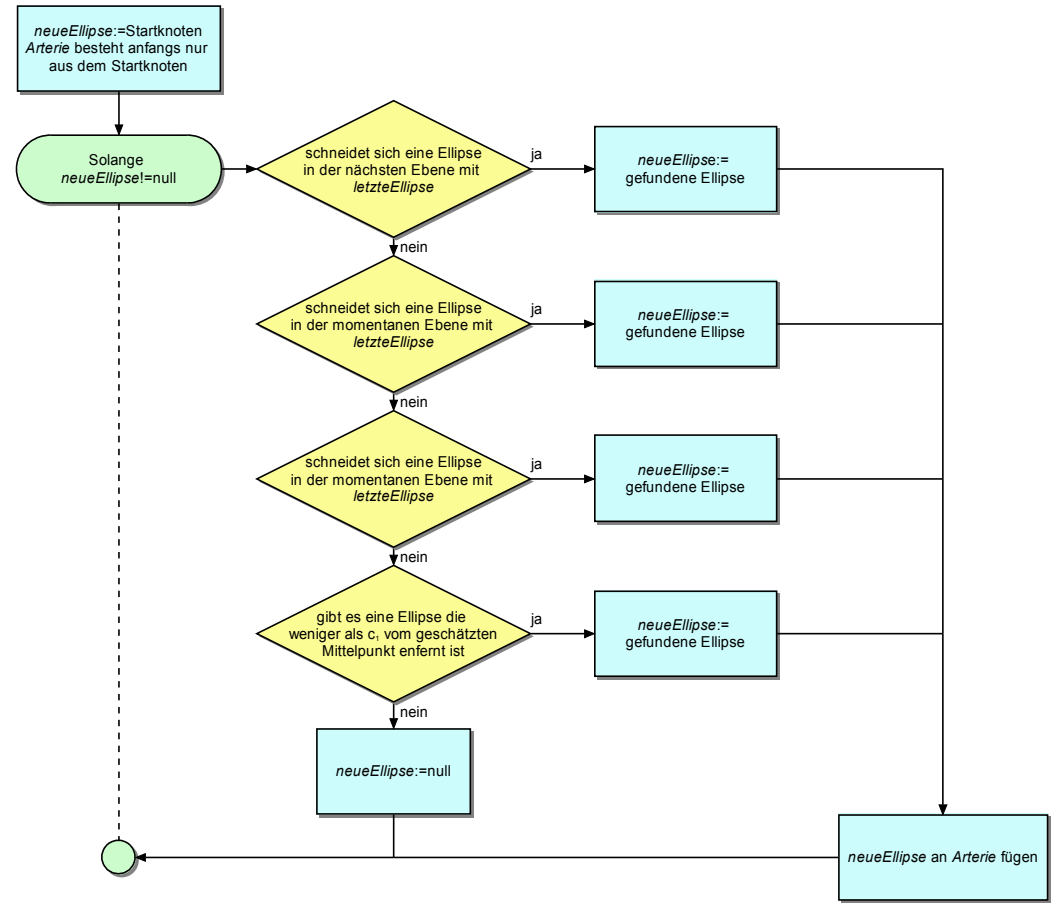


Abbildung 4.10: Finden einer einzelnen Arterie

Jede Ellipse kann nur zu einer Arterie gehören. Aus diesem Grund wird jede Ellipse die zu einer Arterie hinzugefügt wird als verbunden markiert und wird im weiteren nicht mehr berücksichtigt.

Nun muss in der Arterie nach offensichtlichen Bifurkationen gesucht werden. Gibt es noch Ellipsen die sich mit der Arterie überschneiden, werden sie als Startellipsen für neue Arterien verwendet. Als Anschlussstelle werden die Ellipsen verwendet an denen die Überschneidung entdeckt wurde. Werden keine Überschneidungen mehr entdeckt, werden Ellipsen gesucht die nahe an der Arterie liegen und weniger als c_2 Abstand zur Arterie haben.

Eine gute Abstandsfunktion zwischen zwei Ellipsen zu finden ist dabei nicht einfach. Der Abstand zwischen zwei Ellipsen

$$dist(E_1, E_2) = min\{|x-y| : x \in E_1,\ y \in E_2\}$$

ist erstens kompliziert zu berechnen und zweitens, wie in Abbildung 4.11 ersichtlich, nicht gut geeignet.

Denn die Ellipsen in (1) sollten keinesfalls verbunden werden, obwohl sie nahe aneinander liegen. Die Drehung der Ellipsen sollte also stärker berücksichtigt werden. Die eingezeichnete Distanz *d* eignet sich besser, da die Ellipsen in (2) wahrscheinlicher zur selben Arterie gehören als die Ellipsen in (1). Die Disanz *d* ist die minmalen Distanz zwischen den Mittelpunkten und den Punkten an den Hauptachsen der Ellipsen.

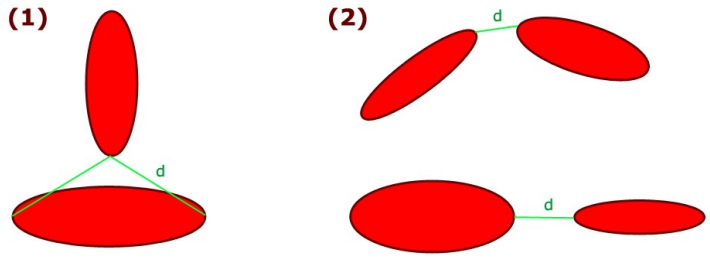


Abbildung 4.11: Abstand zwischen zwei Ellipsen

Weiters kann auch noch die Qualität *q(G,R)* der Ellipse in die Abstandsfunktion einfließen um wahrscheinliche Bifurkationen bevorzugt als Anschlussstelle zu verwenden.

Hat man also schon eine Arterie gefunden und will in dieser Arterie eine Bifurkation suchen verwendet man den in Abbildung 4.12 beschriebenen Algorithmus. Der Aufwand des Algorithmus ist quadratisch, wird also wenn es viele Ellipsen gibt sehr groß. Man kann den Algorithmus beschleunigen indem man Ellipsen, die man schon auf Überschneidungen überprüft hat, markiert. Minimale Abstände zu noch nicht verbundenen Ellipsen können außerdem mit den Ellipsen mitgespeichert werden.

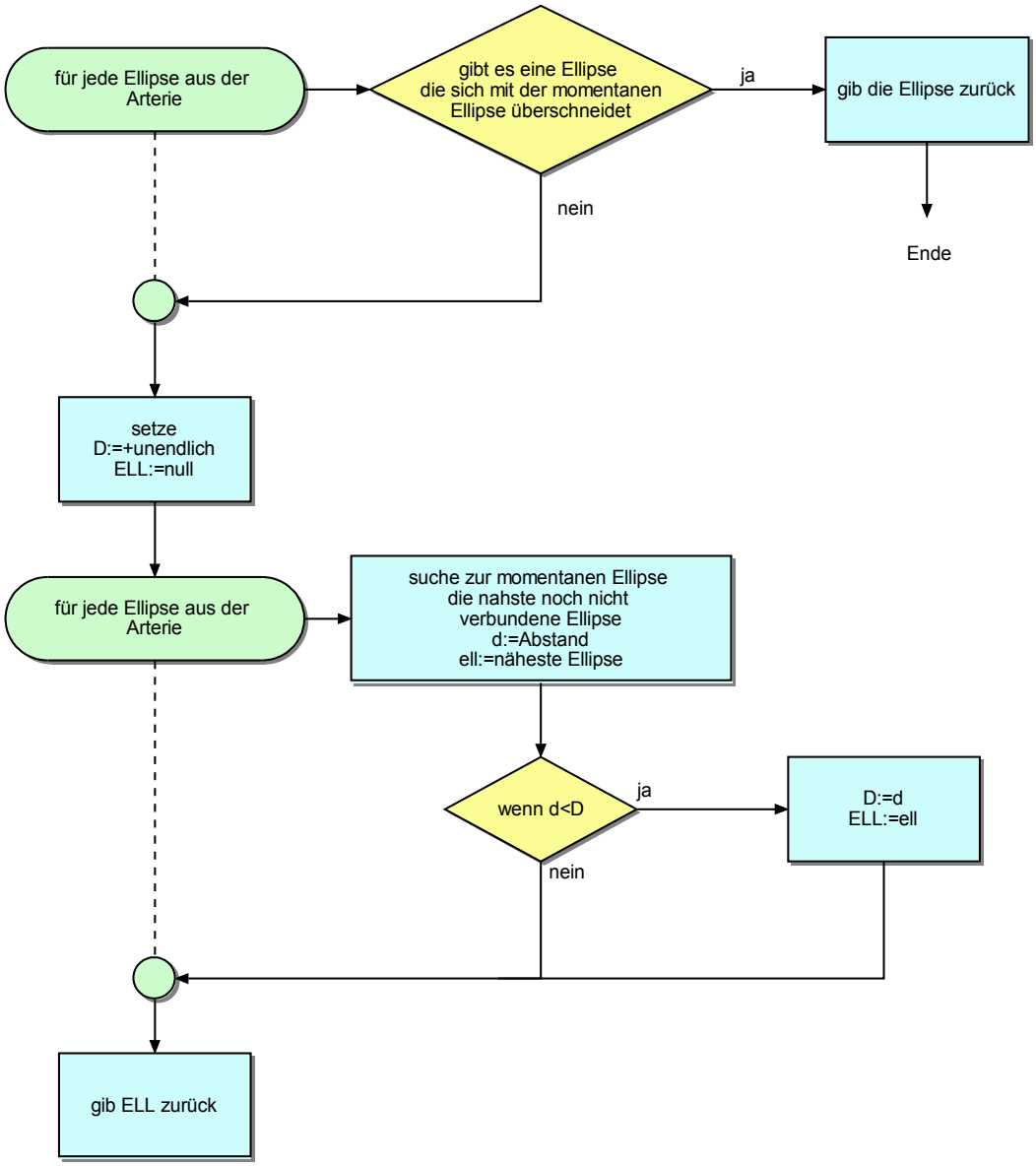


Abbildung 4.12: Finden einer Bifurkation

Die so gewonnenen Arterien kommen in eine Liste und werden sukzessive abgearbeitet. Der Arterienbaum wird mit dem folgenden Algorithmus aufgebaut, der die Methoden *Finden einer Arterie*, Abbildung 4.11, und *Finden einer Bifurkation*, Abbildung 4.12, verwendet.

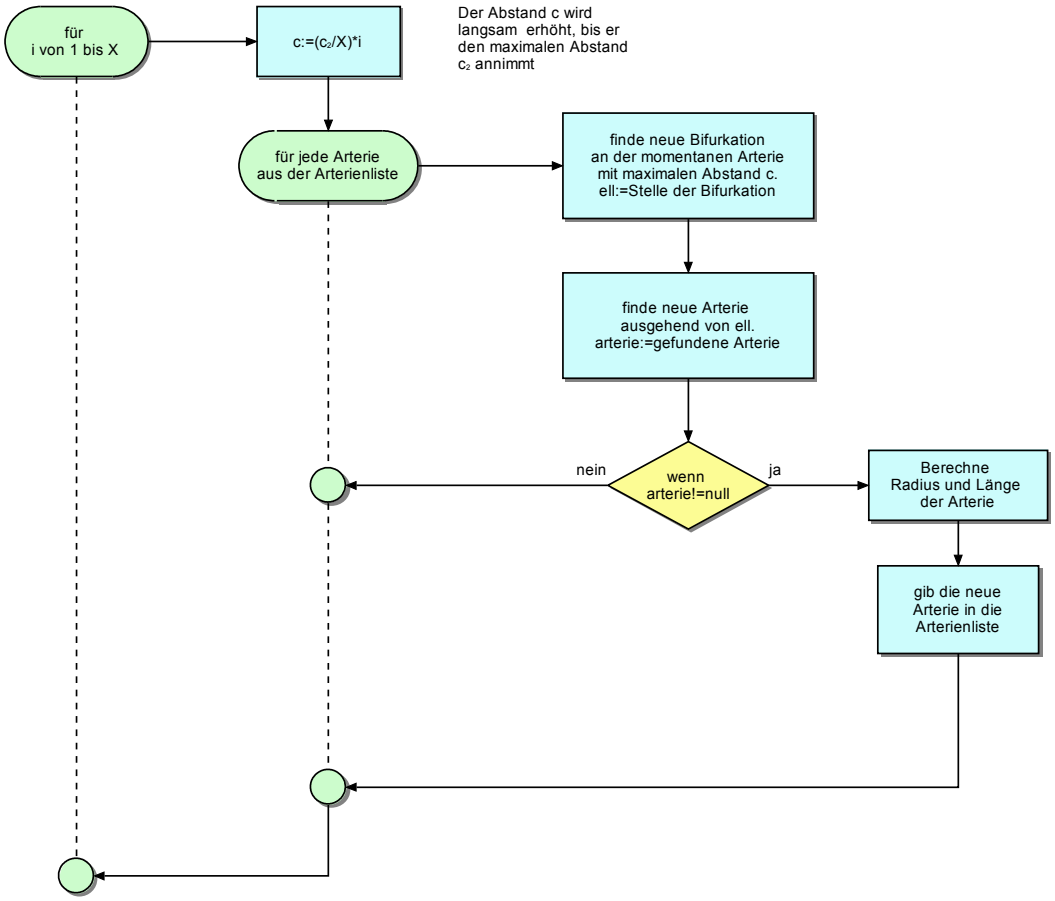


Abbildung 4.13: Aufbauen des Arterienbaums aus den Ellipsen

Die Mittellinie der Arterie wird aus den Mittelpunkten der Ellipsen aufgebaut. Einige Ellipsen können aber schlecht oder falsch erkannt worden sein, außerdem kommt es durch die begrenzte Auflösung der Volumsdaten kommt zu Diskretisierungsfehlern. Die Mittellinie der Arterie erscheint dadurch verwackelt. Um diesen störenden Einflüssen entgegenzuwirken, wird die Mittellinie durch geglättet. Der Kantenverlauf wird mit einem 1-dimensionalen Binomial-[1 2 1] Kern geglättet. Dabei wird die neue Position des Knotens aus dem Mittel seiner ursprünglichen Position und dem Mittel seiner direkten Nachbarn berechnet, siehe Abbildung 4.14.

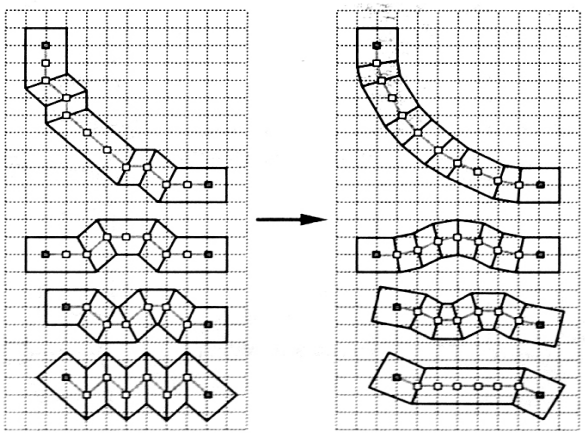


Abbildung 4.14: Die ungeglättete Mittellinie wird mit einem [1 2 1]-Kern geglättet.

Der mit dem obigen Algorithmus gewonnene Arterienbaum muss noch modifiziert werden, bervor er als Eingang für die dynamischen Herkreislaufsimulation verwendet werden kann. Es muss ein Graph exportiert werden der aus Knoten und Kanten besteht, die Kanten haben dabei Attribute wie Länge und Durchmesser. Eine Arterie, die mit dem Algorithmus, der in Abbildung 4.13 beschrieben ist, gewonnen wurde, kann mehrere Bifurkationen haben, wie man an der gelb markierten Arterie in Abbildung 4.15 sieht. Sie muss an jeder Bifurkation zerschnitten werden. Jede Bifurkation wird dabei ein Knoten, jedes Arterienstück eine Kante im Graphen, der exportiert werden soll. Länge sowie gemittelter Durchmesser der Arterienstücke müssen neu berechnet werden, wie es in Abbildung 4.15 am rechten Arterienbaum ersichtlich ist.

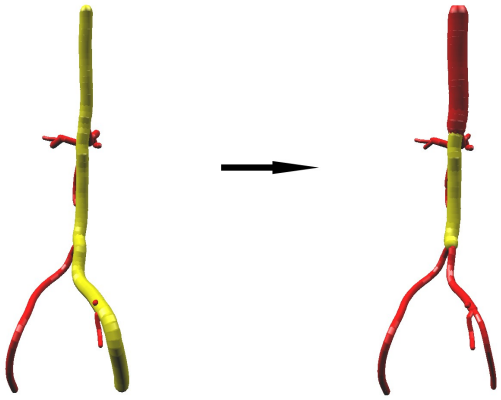


Abbildung 4.15: Die Arterienstücke müssen an den Bifurkationen zerschnitten werden.

Auf diese Weise erhält man ein graphentheoretisches Modell des Arterienbaums, das als XML Datei exportiert werden kann und das für die dynamische Herzkreislaufsimulation verwendet wird. Ergebnisse der Simulation können farblich im idealisierten Modell dargestellt werden.

Kapitel 5
Implementierung

5.1 Wahl der Programmiersprache

Der Algorithmus wurde in Java implementiert. Java bietet gegenüber anderen Programmiersprachen wie Fortran oder C bedeutende Vorteile.

- Java ist Plattformunabhängig und bietet die Möglichkeit Programme mit der Hilfe von Java Applets ins Internet zu stellen.
- Mit Java lassen sich unkompliziert graphische Benutzeroberflächen erstellen.
- Java ist objektorientiert. Das vereinfacht die Modularisierung der Problemstellung und ermöglicht eine gute Kapselung der einzelnen Klassen.
- Der Quellcode kann mit Javadoc dokumentiert werden. Das erleichtert es einzelne Klassen in anderen Projekten wiederzuverwenden.
- Durch die Java3D Erweiterung ist es möglich einfacher interaktive 3D Anwendungen zu erstellen, dabei wird 3D-Graphikhardware bestmöglich unterstützt.
- Java bietet eine hervorragende Unterstützung von *XML*.

Ein weiterer Grund für Java ist, dass auch das stationäre Herzkreislaufmodell in Java implementiert ist. Dieses Programm dient außerdem als graphische Benutzeroberfläche für das rechenintensivere dynamische Herzkreislaufmodell, dass in C++ implementiert wurde. Es liegt daher nahe, das neu entwickelte Programmpaket in die bestehende Oberfläche zu integrieren, was in Form eines Import-Wizzards geschehen könnte. Es wäre so möglich MR-Querschnittsbilder eines Patienten zu importieren, den Arterienbaum zu extrahieren und das so

gewonnene graphentheoretische Modell direkt für die Simulation des Kreislaufs zu verwenden.

5.2 Programmbeschreibung

Das Programm wurde als Wizzard implementiert, das heißt das Extrahieren erfolgt in mehreren Dialogschritten, die hintereinander ausgeführt werden. Das hat zum einem den Vorteil, dass sich das Programm leichter in eine beliebige graphische Benutzeroberfläche integrieren lässt, zum anderen wird die Problemstellung in Teilprobleme zerlegt. Jeder Dialogschritt ist dabei völlig gekapselt, kann also als eigenständige Anwendung betrachtet werden. Das vereinfacht die Entwicklung und die Fehlersuche gewaltig, denn durch klar definierte Schnittstellen zwischen den Dialogschritten ist es nicht möglich, dass sich Fehler von einem Schritt in den nächsten fortpflanzen. In den folgenden Kapiteln werden die einzelnen Dialogschritte erklärt.

5.2.1 Öffnen der Querschnittbilder

Im ersten Schritt sollen Volumsdaten oder Querschnittbilder geladen werden. Werden nur wenige Querschnitte verwendet, werden neue Schnitte durch Interpolation erzeugt und eingefügt.

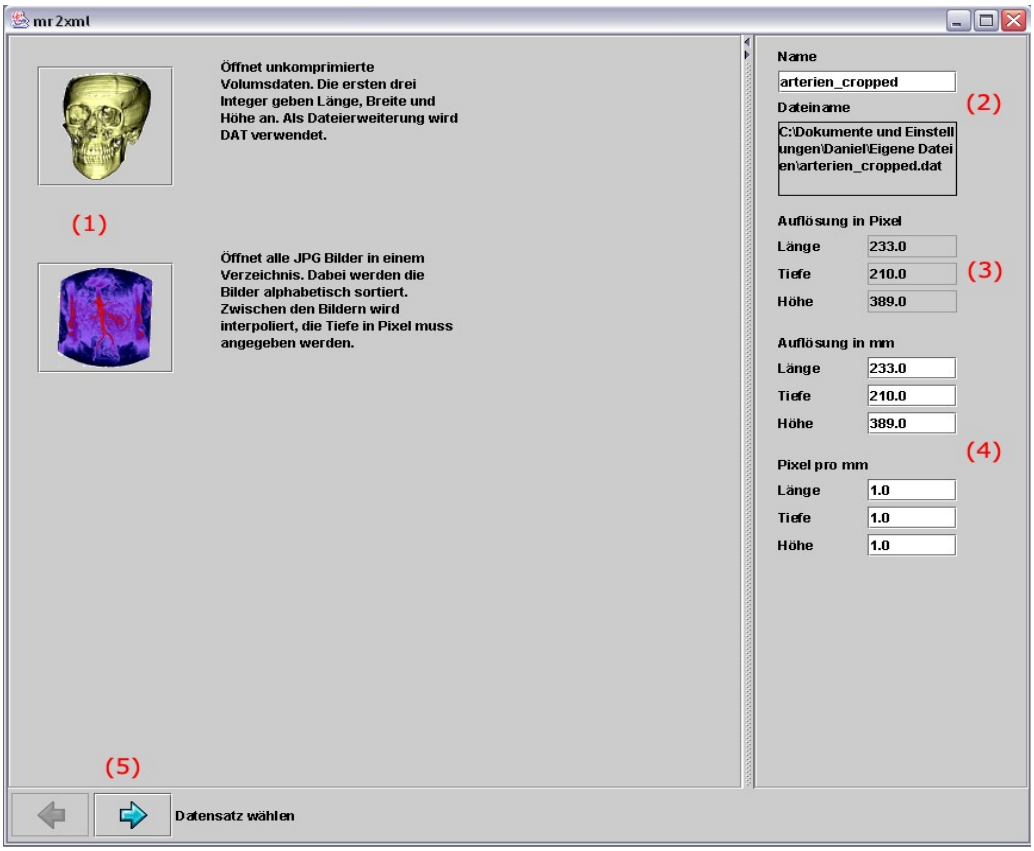


Abbildung 5.1: Öffnen von Volumsdaten oder Querschnittbildern

(1) Es können entweder Volumsdaten oder Querschnittbilder im *JPG* Format geöffnet werden. Das Programm kann jedoch leicht erweitert werden um mehr Dateiformate zu unterstützen, wie zum Beispiel *DICOM* Bilder, die in der Medizin häufig verwendet werden.

(2) Hier kann ein Name für die resultierende *XML* Datei gewählt werden. Darunter wird das momentane Arbeitsverzeichnis, aus dem die Daten geladen wurden, angezeigt.

(3) Es werden die Ausmaße der Querschnittbilder in Pixel, beziehungsweise die des Volumens in Voxel angezeigt.

(4) Ist die Anzahl der Pixel pro mm in den Querschnittbildern bekannt, können sie in die Felder eingetragen werden. Länge, Tiefe und Höhe des Volumens werden dann daraus berechnet. Ansonsten können Länge, Tiefe und Höhe geschätzt werden und die Pixel pro mm errechnen sich aus dieser Schätzung.

(5) Wurde ein Datensatz geöffnet gelangt man zum nächsten Dialogschritt.

5.2.2 Blickrichtung, Bereich und Kontrast wählen

In diesem Dialogschritt ist das Fenster in zwei Hälften geteilt. Auf der linken Seite sieht man ein einzelnes Querschnittblid, wobei mit dem linken Schieberegler gewählt werden kann, wo sich dieser Schnitt im Volumen befindet. Auf der rechten Seite befindet sich eine *MIP* durch das Volumen, wie in Kapitel 3.2.5 beschrieben. In diesem Arbeitsschritt wird versucht die Volumsdaten so zu modifizieren, dass die Arterien im folgendem Schritt optimal erkannt werden können.

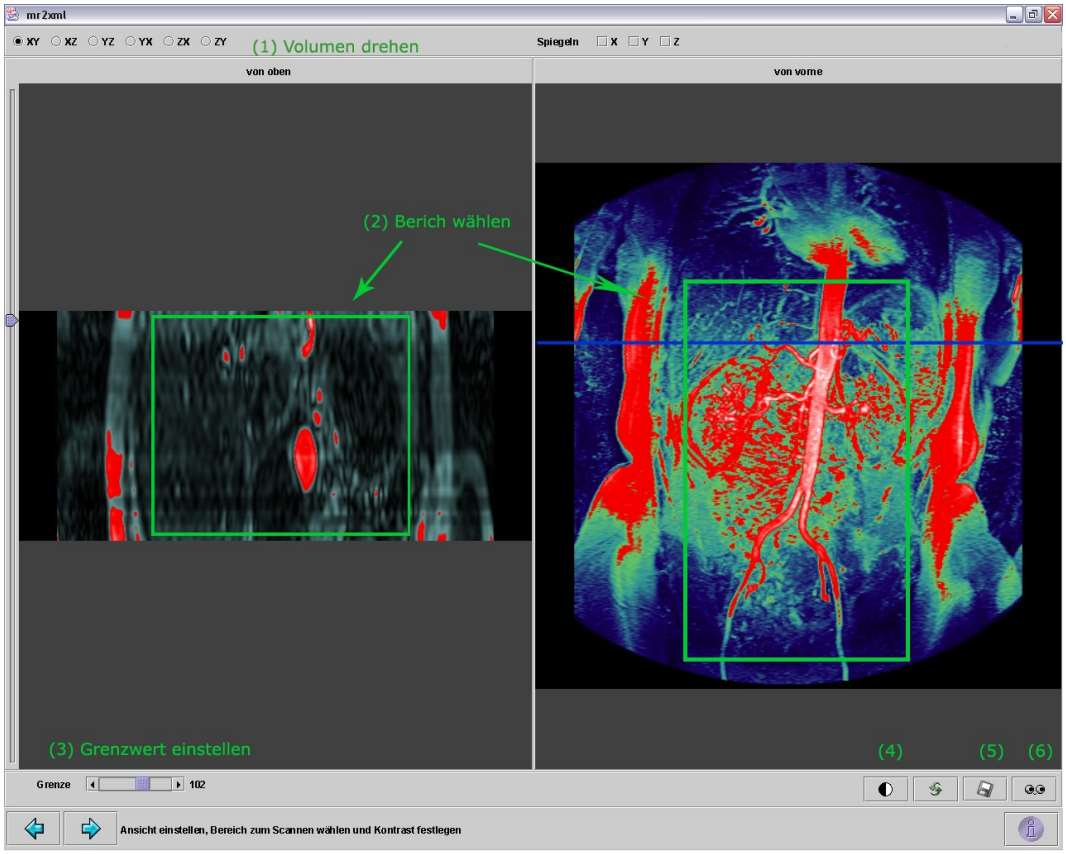


Abbildung 5.2: Blickrichtung, Berreich und Kontrast wählen

(1) Zunächst wird die Blickrichtung gewählt. Es ist dabei möglich von jeder Hauptachse auf das Volumen zu schauen. Im linken Bereich kann die Ebene gewählt werden, die betrachtet wird. Das Volumen sollte dabei so gedreht werden, dass in der linken Hälfte des Fensters die Arterien als Ellipsen erscheinen, also annähernd orthogonal auf die

Schnitte stehen. Auf der rechten Hälfte ist das Volumen dann von vorne zu sehen. Die blaue Linie kennzeichnet, wo der links sichtbare Querschnitt im Volumen verläuft.

(2) Als nächstes sollte der Bereich gewählt werden, der für die Segmentierung interessant ist. Das verkleinert die Volumsdaten und verhindert Übersegmentierung. Der ausgewählte Bereich ist als grünes Rechteck sichtbar. Mit dem Knopf wird der Rest des Volumens außerhalb der Rechtecke abgeschnitten.

(3) Nun wird der Grenzwert für die Segmentierung so gewählt, dass die Arterien im *MIP* rot erscheinen.

(4) Ist der Kontrast wie in Abbildung 5.2 am oberen und unteren Ende des *MIP* schwächer als im übrigen Volumen, kann die Helligkeit und der Kontrast angepasst werden. In Abbildung 5.3 wurde oben und unten im Volumen die Helligkeit und der Kontrast leicht angehoben.

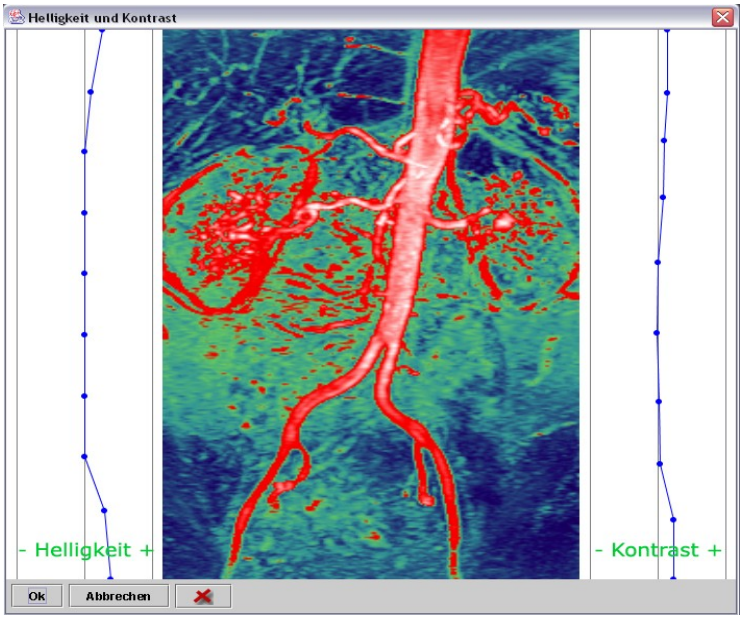


Abbildung 5.3: Helligkeit und Kontrast

(5) Das durch die letzten Schritte modifizierte Volumen, kann als *DAT* Datei zwischengespeichert werden.

(6) Um einen Eindruck zu erlangen, wie gut das Volumen für die Segmentierung geeignet ist, kann ein *First Hit Rendering* oder ein *MIP* des Volumens aus einem beliebigen Blickwinkel in Echtzeit erstellt

werden. Abbildung 5.4 kann als Kontrolle für das Endergebnis der Segmentierung verwendet werden.

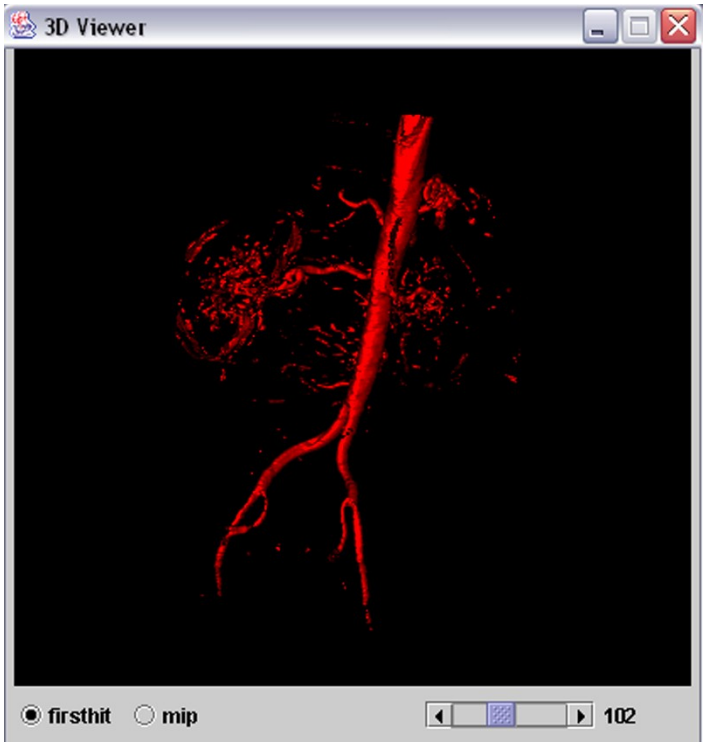


Abbildung 5.4: First HIt Rendering der Volumsdaten

5.2.3 Segmentierung

Die eigentliche Segmentierung der Querschnittbilder findet in diesem Schritt statt. Als Arterien erkannte Bereiche werden rot dargestellt.

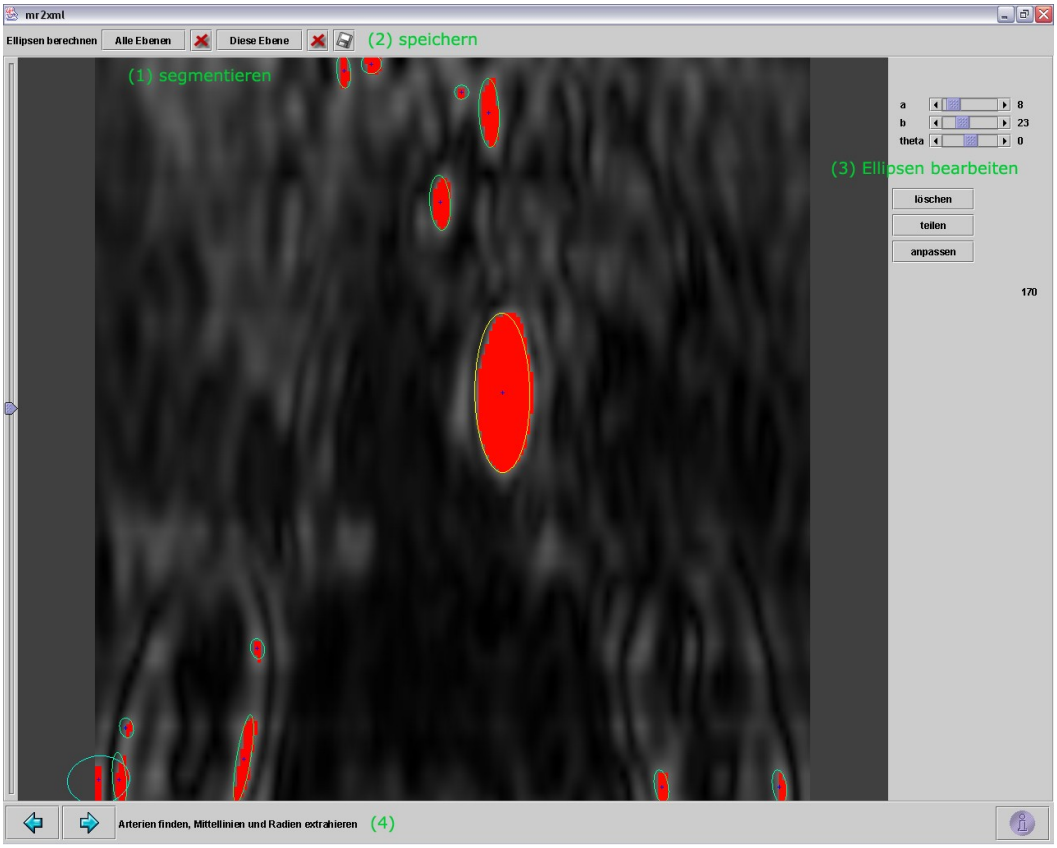


Abbildung 5.5: Segmentierung der Querschnitte

(1) In jedem Schnitt werden Ellipsen über die gefundenen Arterien gelegt. Aus diesen Ellipsen wird im letzten Schritt das graphentheorethische Arterienmodell aufgebaut.

(2) Die gefundenen Ellipsen können zwischengespeichert werden.

(3) Es gibt die Möglichkeit falsch erkannte Ellipsen nachzubearbeiten. Man kann Ellipsen markieren, löschen, teilen, verschieben, drehen und die Größe der Hauptachsen händisch ändern oder automatisch anpassen.

(4) Bevor man zum letzten Dialogschritt gelangt, muss eine Ellipse als Ausgangspunkt für das Verbinden der Ellipsen markiert werden.

5.2.4 Das Verbinden der Ellipsen

Im letzten Schritt werden die Ellipsen zu einem Arterienbaum verbunden. Dieses graphentheoretische Modell kann als *XML* Datei exportiert und zur dynamischen Herzkreislaufsimulation verwendet werden.

Es ist möglich, die erkannten Arterien zu bearbeiten, nicht erkannte Zweige können händisch hinzugefügt werden, falsch erkannte Arterien können gelöscht werden.

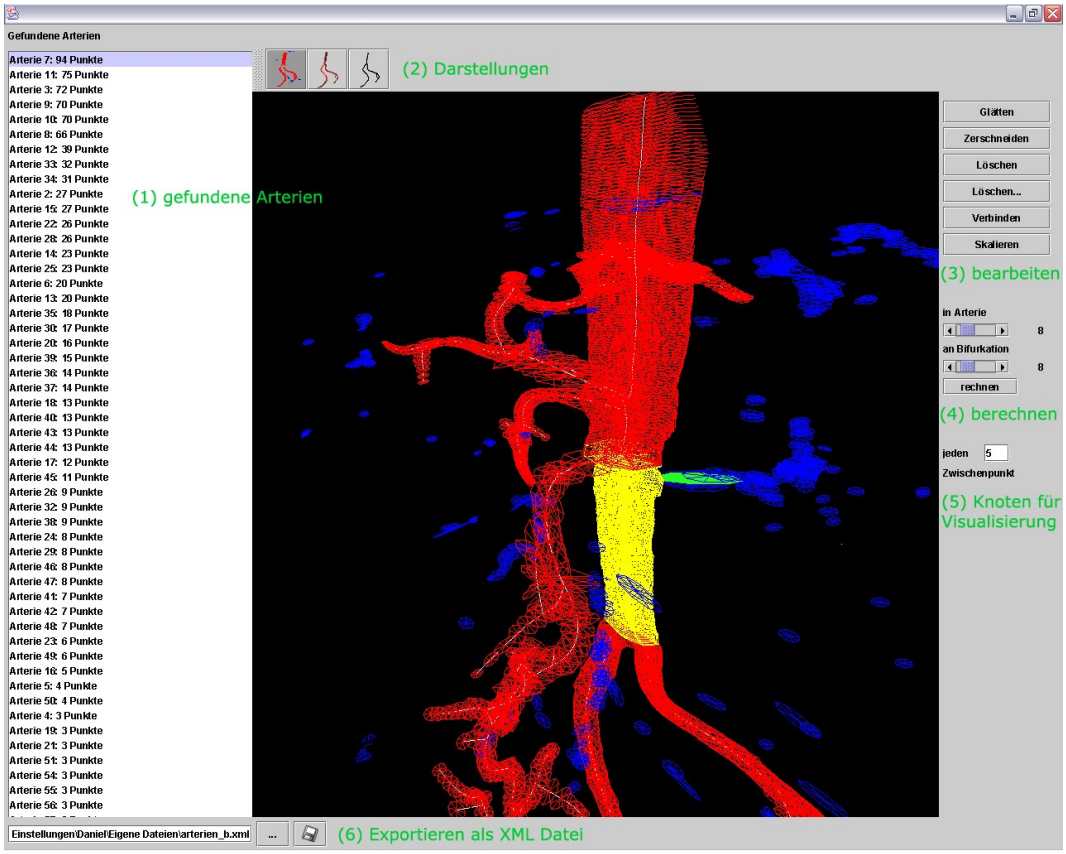


Abbildung 5.6: Exportieren der XML Datei

(1) In dieser Liste stehen die erkannten Arterien. In der Liste markierte Arterien werden in der Darstellung gelb eingefärbt, verbundene Ellipsen sind rot, nicht verbundene Ellipsen werden blau dargestellt. Einzelne Ellipsen markiert man mit einem Rechtsklick, sie erscheinen grün. Ein

Doppelklick auf eine Arterie in der Liste öffnet ein Eingabefeld, in dem die Eigenschaften der Arterie geändert werden können, siehe Abbildung 5.7.

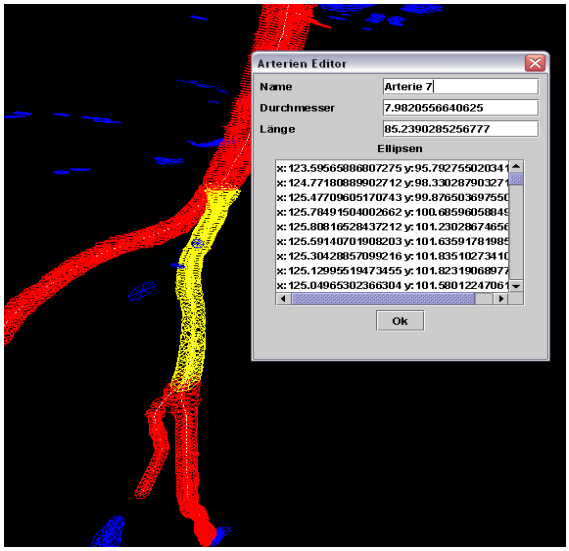


Abbildung 5.7: Bearbeiten von Arterien

(2) Der Arterienbaum kann auf drei verschiedene Arten dargestellt werden. Die erste Darstellung ist jene, in der die Ellipsen sichtbar sind. Die zweite stellt den Baum als Röhren dar, genauso wie er als *XML* Datei exportiert wird. In dieser Darstellung kann das Ergebnis der Segmentierung visuell überprüft werden. Einzelne Arterien können mit einem Rechtsklick markiert und bearbeitet werden. Die dritte Darstellung zeichnet nur die Mittellinie des Graphen. Konnektivität und Glattheit der Mittellinie können so dargestellt werden.

Abbildung 5.8: Dastellung des graphentheoretischen Modells

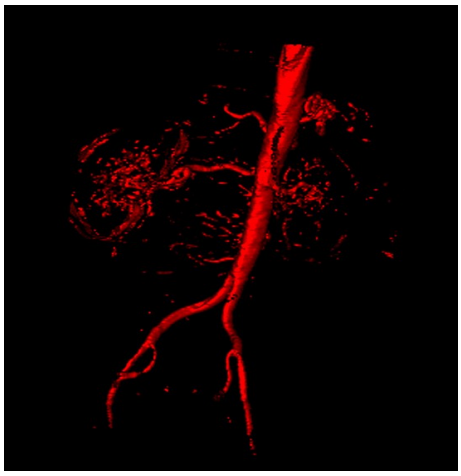


Abbildung 5.9: First Hit Rendering der Volumsdaten

(3) Es gibt verschieden Möglichkeiten das Ergebnis zu verbessern. Zunächst sollte der Arterienbaum geglättet werden, dadurch erscheint die Mittellinie weicher und Diskretisierungsfehler werden aufgehoben.

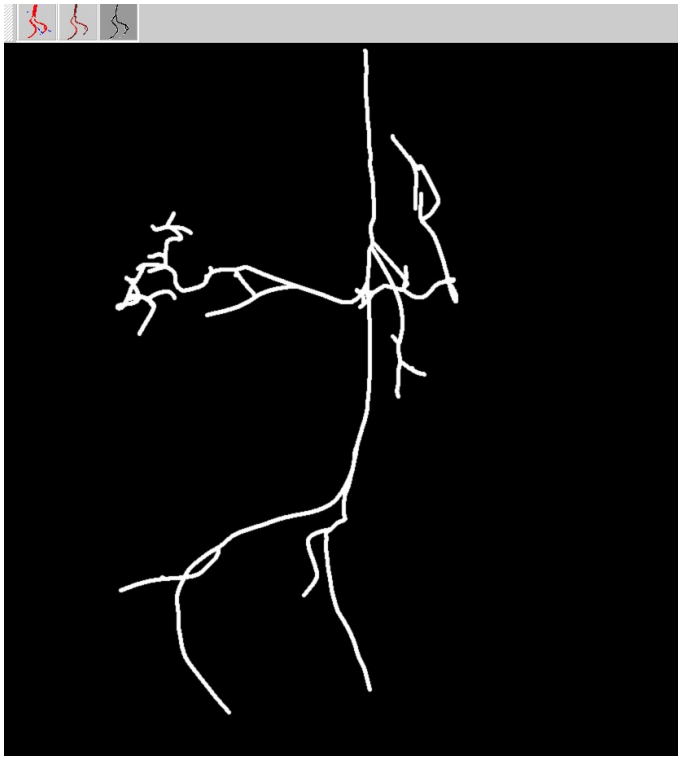


Abbildung 5.10: Geglättete Mittellinie

Damit der Arterienbaum zur dynamischen Simulation verwendet

werden kann, müssen die Arterien an jedem hydraulischen Knoten getrennt werden, wie in Abbildung 4.15 erklärt wird. Falsch erkannte Arterien können markiert und gelöscht werden. Es können auch alle Arterien, deren Radius oder Größe unter einem Grenzwert liegen, gelöscht werden. Wurden ganze Zweige nicht erkannt, kann man die Anfangsellipse des Zweiges markieren und händisch mit einer Arterie verbinden.

(4) Das Verbinden der Ellipsen erfolgt mit zwei Parametern, die in Kapitel 4.2.4 beschrieben wurden. Wird der Arterienbaum neu berechnet, gehen vorangegangene Änderungen verloren.

(5) Die Anzahl der Knoten für die Visualisierung ist für die dynamische Simulation nicht von Bedeutung. Man braucht sie jedoch um Ergebnisse im Gefäßsystem darzustellen. Werden viele Knoten gespeichert, verbessert das die Qualität der Visualisierung.

(6) Der so gewonnene Arterienbaum kann als *XML* Datei exportiert werden.

Kapitel 6
Ausblicke

Ziel der Arbeit war es ein Programm zu entwickeln, das aus *MR* Querschnittsbildern ein Modell der Arterien aus Röhren und hydraulischen Knoten erstellt. Das ist sehr gut gelungen, es ist wird jedoch in nächster Zeit noch nötig sein das Programm mit verschiedenen Datensätzen zu testen, um bestmögliche Ergebnisse für die dynamische Simulation zu erzielen.

Mögliche zukünftige Weiterentwicklungen des Programms umfassen alles, was das Experementieren mit graphentheoretischen Modellen erleichtert.

- Es sollte möglich sein Querschnittbilder im *DICOM* Format zu importieren. Bis jetzt müssen die Bilder zuerst in *JPG* Dateien konvertiert werden, um anschließend ins Programm importiert werden zu können. Da *DICOM* das führende Format für tomographische Aufnahmen ist und Informationen über die Aufnahme, wie Pixel pro mm, und Daten des Patienten mitspeichert, sollte das Programm dahingehend erweitert werden.
- Das Programm sollte zusammen mit der statischen sowie der dynamischen Herzkreislaufsimulation in eine gemeinsame einheitliche Oberfläche eingebettet werden. Das würde die Segmentierung und das anschließende Experimentieren mit den gewonnen Daten in der Simulationssoftware näher zusammenrücken.
- Die Ergebnisse der Simulation sollen in den graphentheoretischen

Modellen veranschaulicht werden, wie in Kapitel 2.4.4 vorgeschlagen wurde. Das würde das Experementieren im Modell um vieles erleichtern, da die Ergebnisse leichter interpretiert werden könnten.

Abbildungsverzeichnis

Abbildung 1.1: Zusammenhang zwischen Visualisierung und HK Simulation 2
Abbildung 2.1: Der Blutkreislauf 5
Abbildung 2.2: Das Herz 6
Abbildung 2.3: Druckverlauf in der Aorta über einen Herzzyklus 7
Abbildung 2.4: Blockschaltbild eines Regelkreises 14
Abbildung 2.5: Verzögerungsglied erster Ordnung mit T = 50 15
Abbildung 2.6: Blockschaltbild eines Systems 16
Abbildung 2.7: Der Arterienbaum als Netzwerk 17
Abbildung 2.8: Schema des ungeregelten Kompartmentmodells nach Möller 20
Abbildung 2.9: Der mittlere arterielle Blutdruck im Ergometertest 22
Abbildung 3.1: Die Visualisierungs-Pipeline 25
Abbildung 3.2: MRT Querschnitt eines Kopfes 26
Abbildung 3.3: CT Querschnitt eines Kopfes 26
Abbildung 3.4: Die Eckpunkte einer Zelle sind Voxel 28
Abbildung 3.5: Die Voxel innerhalb einer Zelle erhält man durch Interpolaiton 28
Abbildung 3.6: Nearest Neighbour 28
Abbildung 3.7: Trilineare Interpolation 28
Abbildung 3.8: Zweidimensionale Transferfunktion 30
Abbildung 3.9: Windowing 32
Abbildung 3.10: Verschieden Auswahlverfahren 33
Abbildung 3.11: Die Strahlen treffen orthogonal auf die Querschnitte 34
Abbildung 3.12: Das gescherte Bild wird durch Warp - Transformation entzerrt 34
Abbildung 3.13: First Hit Projektion von Arterien 35
Abbildung 3.14: MIP eines Arterienbaums 37
Abbildung 3.15: Wolke aus Leuchtpartikeln 38
Abbildung 3.16: Dunkle Wolke 39
Abbildung 3.17: Luminiszierende Wolke 39
Abbildung 4.1: Der für die Segmentierung interessante Bereich muss

ausgeschnitten werden 42
Abbildung 4.2: Durch stärkeren Kontrast sind die Arterien besser erkennbar. 43
Abbildung 4.3: Es wird binär entschieden, ob ein Voxel eine Arterie ist 44
Abbildung 4.4: Die verschiedenen Konfigurationen lokaler Konkavität 46
Abbildung 4.5: Das Graustufenbild wird als topographisches Relief interpretiert 52
Abbildung 4.6: Das Umrechnen der Querschnittbilder 49
Abbildung 4.7: Bifurkation 51
Abbildung 4.8 51
Abbildung 4.9: Erkannte Ellipsen in einem Querschnitt 52
Abbildung 4.10: Finden einer einzelnen Arterie 55
Abbildung 4.11: Abstand zwischen zwei Ellipsen 56
Abbildung 4.12: Finden einer Bifurkation 57
Abbildung 4.13: Aufbauen des Arterienbaums aus den Ellipsen 58
Abbildung 4.14: Die ungeglättete Mittellinie wird mit einem [1 2 1]-Kern geglättet. 59
Abbildung 4.15: Die Arterienstücke müssen an den Bifurkationen zerschnitten werden. 59
Abbildung 5.1: Öffnen von Volumsdaten oder Querschnittbildern 63
Abbildung 5.2: Blickrichtung, Berreich und Kontrast wählen 64
Abbildung 5.3: Helligkeit und Kontrast 65
Abbildung 5.4: First HIt Rendering der Volumsdaten 66
Abbildung 5.5: Segmentierung der Querschnitte 67
Abbildung 5.6: Exportieren der XML Datei 68
Abbildung 5.7: Bearbeiten von Arterien 69
Abbildung 5.8: First Hit Rendering der Volumsdaten 69
Abbildung 5.9: Dastellung des graphentheoretischen Modells 69
Abbildung 5.10: Geglättete Mittellinie 70

Literaturverzeichnis

[1] Dieter Möller, Dobrivoje Popovic and Georg Thiele, *Modeling, Simulation and Parameterestimation of the Human Cardiovascular System*, volume 4 of Advances in Control Systems and Signal Processing, Vieweg, 1983.

[2] Yvonne Stickler, *Dynamisches Modell des Herzkreislaufs mit Regelung*. Diplomarbeit, Technische Universität Wien, Österreich, 2002.

[3] Johannes Kropf, Dynamische *Herzkreislaufsimulation unter Berücksichtigung verteilter Parameter*. Diplomarbeit, Technische Universität Wien, Österreich, 2003.

[4] Christian Roland Almeder, *Hydrodynamic Modelling and Simulation of the Human Arterial Blood Flow,* PhD Thesis, Technical University of Vienna, Austria 1999.

[5] Y. Stickler, J. Kropf, M. Wibmer, S. Wassertheurer and F. Breitenecker, *Dynamisches Modell des Herkreislaufs mit Regelung,* Proceedings of the ASIM 2002, Rostock, 2002.

[6] Wilmer W. Nichols and Michael F. O Rourcke, *McDonalds blood Flow in arteries,* Arnold 1998.

[7] Armin Markus Kanitsar, *Advanced Visualisation Techniques for Vessel Investigation,* Thesis, Technical University of Vienna, Austria 2001.

[8] Petr Felkel, Rainer Wegenkittl, and Katja Bühler , *Surface Models of Tube Trees*, Proceedings of Computer Graphics International 2004, pp.70-77, 2004.

[9] Marc Levoy, *Display of Surfaces from Volume Data*, IEEE Computer Graphics and Applications, Vol. 8, No. 3, May 1988.

[10] William E. Lorensen and Harvey E. Cline, *Marching cubes: A high resolution 3D surface construction algorithm,* ACM SIGGRAPH Computer Graphics, Vol. 21, August 1987.

[11] Gordon Kindlmann and James Durkin, *Semi-automatic generation of*

transfer functions for direct volume rendering, IEEE Symposium on Volume Rendering Proceedings, October 1998.

[12] Philippe Lacroute and Marc Levoy, *Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation*, Proc. SIGGRAPH 94, Orlando, Florida, pp.451-458 , July 1994.

[13] Nelson Max, *Optical Models for Direct Volume Rendering,* IEEE Transactions on Visualization and Computer Graphics, Vol. 1, pp.99–108, June 1995.

[14] Cebral, J.R. Rainald Lohner, *From Medical Images to CFD Meshes,* 8th International Meshing Roundtable, South Lake Tahoe, CA, U.S.A., pp.321-331, October 1999.

[15] Lee Vincent and Pierre Soille, *Watersheds in digital spaces: An efficient algorithm,* IEEE PAMI, Vol. 13, pp.583-598, 1991.

[16] Daniel Selman, *Java3d Programming,* Manning 2002.