

Diplomarbeit

Generierung und Dynamik physikalisch basierter
3D-Modelle zur Segmentierung des linken
Ventrikels in SPECT-Daten

LARS DORNHEIM

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Institut für Simulation und Graphik

Betreuer:
Prof. Dr. Klaus D. Tönnies
Prof. Dr. Bernhard Preim

Für Papi

Zusammenfassung

Herzkrankheiten, gerade Herzinfarkte, sind weit verbreitete und schwerwiegende Erkrankungen. Moderne funktionelle bildgebende Verfahren der Medizin wie SPECT können einen großen Beitrag zur Diagnose und vor allem zur quantitativen Beurteilung dieser Krankheiten leisten. Die Auswertung der großen, dreidimensionalen Datenmengen kann bisher aber nur langwierig durch Fachpersonal erfolgen. Das betrifft vor allem die Segmentierung des linken Ventrikels in SPECT-Daten, der infarktbedingt oft nur lückenhaft zu sehen ist.

Durch diese Arbeit wird zum ersten mal eine zuverlässige, schnelle und vollautomatische Segmentierung des linken Ventrikels in humanen 3D-SPECT-Daten möglich. Diese Segmentierung stützt sich dabei auf Modelle, die ebenfalls mit in dieser Arbeit entwickelten Verfahren zum ersten mal vollautomatisch aus Beispielsegmentierungen generiert werden können.

Die verwendeten, in dieser Arbeit entwickelten Modelle selbst stellen dabei eine neue Klasse expliziter dynamischer 3D-Modelle dar, die erstmalig echte Formstabilität bei gleichzeitiger hoher Anpassungsflexibilität aufweisen.

Abstract

Heart diseases, especially cardiac infarctions, are widespread and serious diseases. Modern functional medical imaging technologies like SPECT can contribute much to diagnosis and particularly to quantitative assessment of these diseases. The evaluation of the large three-dimensional datasets must currently be performed manually by medical experts. This applies in particular to the task of segmenting the left ventricle in 3D SPECT data, which often bears low signal gaps caused by infarcts.

By the results of this work, a reliable, fast and fully automatic segmentation of the left ventricle in human 3D SPECT datasets is rendered possible for the first time. This segmentation is based on 3D anatomical models, which can for the first time be generated fully automatically from sample segmentations by techniques developed in this work.

The employed models developed in this work establish a new class of explicit dynamic 3D models, featuring real 3D shape stability paired with high adaption flexibility.

Danksagung

Ich danke meinem Betreuer Prof. Tönnies. Besonders wertvoll waren die mitunter langen Gespräche über für die Arbeit relevante Ansätze und die mit ihnen verbundenen Probleme. Durch seine Hilfe war es möglich, bereits während des knappen Bearbeitungszeitraumes dieser Arbeit zwei wissenschaftliche Aufsätze aus ihr auszukoppeln und einzureichen.

Ich danke auch Kat Dixon, durch deren Forschung auf den vorliegenden SPECT-Daten dieses Diplomthema überhaupt erst zustande kam. Sie repräsentierte für mich unseren klinischen Partner in Vancouver, Kanada, und war sehr kooperativ. Besonders durch die Bereitstellung von Datensätzen, Handsegmentierungen und Fakten aus dem medizinischen bzw. physikalischen Bereich trug sie wesentlich zum Erfolg dieser Arbeit bei.

Ich danke auch Fitsum Admasu, der durch einige Diskussionen ebenfalls zur Verbesserung der Arbeit beigetragen hat.

Ganz besonders danke ich Jana, meiner Verlobten. Mit ihr konnte ich jederzeit über Belange meiner Diplomarbeit philosophieren. Sie half zudem beim langwierigen Erstellen manueller Vergleichssegmentierungen und las die Arbeit mehrfach Korrektur.

Zum Schluß danke ich noch meinen Eltern, die mich während des ganzen Studiums immer in einer Art und Weise unterstützt haben, daß ich mich voll und ganz auf eben jenes konzentrieren konnte.

Lars Dornheim, im Januar 2005

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Aufgabenstellung	3
1.3	Aufbau der Arbeit	3
1.4	Anmerkungen zur Arbeit	4
2	Verwandte Arbeiten	5
2.1	Modellbasierte Segmentierung	5
2.2	Dynamische Modelle	6
2.2.1	Explizite Modelle	7
2.2.2	Implizite Modelle	8
2.2.3	Modelle mit Steuerungskomponenten	9
2.2.4	Modelle für dreidimensionale Daten	9
2.2.5	Zusammenfassung	11
2.3	Modellgenerierung	11
2.4	Segmentierung des linken Ventrikels in SPECT-Daten	12
3	Grundlagen von Feder-Masse-Modellen	15
3.1	Modelldynamik	15
3.2	Sensoren	17
4	Erweiterte Feder-Masse-Modelle	21
4.1	Torsionskräfte	21
4.2	Explizite Rotationsfähigkeit	22
4.3	Konturflächen	24
4.4	Richtungsgewichtete Kontursensoren	25
4.5	Konturversatz	26
4.6	Zusammenfassung	28
5	Modellgenerierung	29
5.1	Teilstrukturen des Modells	29
5.2	Phasen der Modellgenerierung	30
5.2.1	Erzeugung der Intensitätssensoren	30
5.2.2	Erzeugung der Kontursensoren	31

5.2.3	Vernetzung der Sensorgruppen	32
5.3	Zusammenfassung	32
6	Automatische Segmentierung	35
6.1	Phasen der Segmentierung	35
6.1.1	Phase 1: Startplatzierung	36
6.1.1.1	Mustererkennung durch Faltung	37
6.1.1.2	Mustererkennung durch Vektorähnlichkeiten	38
6.1.2	Phase 2: Formanpassung	41
6.1.3	Phase 3: Konturanpassung	42
6.2	Modellgleichgewicht	43
6.3	Voxelbasierte Modellrepräsentation	44
6.4	Interaktion	45
6.5	Zusammenfassung	46
7	Implementation	47
7.1	Prototyp	47
7.2	Algorithmen und Datenstrukturen	48
7.2.1	Dynamische Sensorkarten	48
7.2.2	Sensorbewegungsverfolgung	49
7.2.3	Voxelbasierte Modellrepräsentation	50
7.3	Benutzeroberfläche	51
8	Evaluierung	55
8.1	Datenmaterial	55
8.2	Evaluierungsmaße	57
8.3	Erweiterte Feder-Masse-Modelle	58
8.4	Modellgenerierung	63
8.5	Experimente zum Segmentierungsprozeß	66
8.5.1	Phase 1	66
8.5.2	Phase 2 und 3	67
8.5.3	Torsionskräfte	74
8.5.4	Explizite Rotationsfähigkeit	74
8.6	Segmentierung der Datensätze	75
8.7	Zusammenfassung	78
9	Zusammenfassung und Ausblick	81
9.1	Zusammenfassung	81
9.2	Weiterführende Ergebnisse	82
9.3	Ausblick	82
A	Parameter der Modellsimulation	85
B	Ergebnistabellen der Segmentierungen	87
	Literaturverzeichnis	91

<i>INHALTSVERZEICHNIS</i>	XIII
Tabellenverzeichnis	95
Abbildungsverzeichnis	97

Kapitel 1

Einleitung

1.1 Motivation

Herzkrankheiten, gerade Herzinfarkte, sind weit verbreitete und schwerwiegende Erkrankungen und eine der häufigsten Todesursachen in der westlichen Welt. Moderne funktionelle bildgebende Verfahren der Medizin können einen großen Beitrag zur Diagnose und vor allem zur quantitativen Beurteilung dieser Krankheiten leisten. So können mittels SPECT (siehe Abbildung 1.1 für ein Beispiel) funktionelle, dreidimensionale Aufnahmen des linken Ventrikels (LV) gemacht werden, wobei z. B. infarktgeschädigte Regionen als Lücken im gut durchbluteten LV sichtbar werden (siehe Abbildung 1.2a).

Zur qualitativen und quantitativen Beurteilung der Schädigung des LV, z. B. durch einen Infarkt, ist es notwendig, den LV anatomisch in den funktionellen Daten zu segmentieren. Diese Segmentierung ist nicht allein durch die Daten gestützt, sondern erfordert zusätzliches Modellwissen, da im Fall letaler LV-Bereiche Teile der LV-Anatomie nicht abgebildet werden. Diese Aufgabe kann bisher leider nur langwierig durch Fachpersonal erledigt werden.

So benötigt unser klinischer Partner allein für Forschungsstudien hunderte Segmentierungen des linken Ventrikels, die mit erheblichem Zeitaufwand (pro Segmentierung bis zu einer halben Stunde) durch Mitarbeiter erstellt werden müssen (siehe Abbildung 1.2b für ein Beispiel). Menschen haben dabei das Problem, daß sie schichtweise segmentieren und somit den Blick für das Ganze verlieren, was in Bereichen verschwommener Organabgrenzungen bzw. nur teilweise sichtbarer Organe zu Problemen führt, da ihr Modellwissen nur zweidimensional angewendet wird. Zudem birgt diese doch recht stupide Segmentierungsarbeit bei längerer Dauer das Risiko von Fehlern bzw. Ungenauigkeiten.

Bisher konnte diese Aufgabe nicht automatisiert oder in geeigneter Weise technisch unterstützt werden, da die Komplexität der dreidimensionalen Daten und das SPECT-typische Problem eines unter Umständen nur teilweise sichtbaren LV zusammen zu hohe Anforderungen an die vorhandene Segmentierungstechnologie stellten.



Abbildung 1.1: Schichtbild einer 3D-SPECT-Aufnahme eines menschlichen Brustkorbes. Die helle Struktur ist der linke Ventrikel.

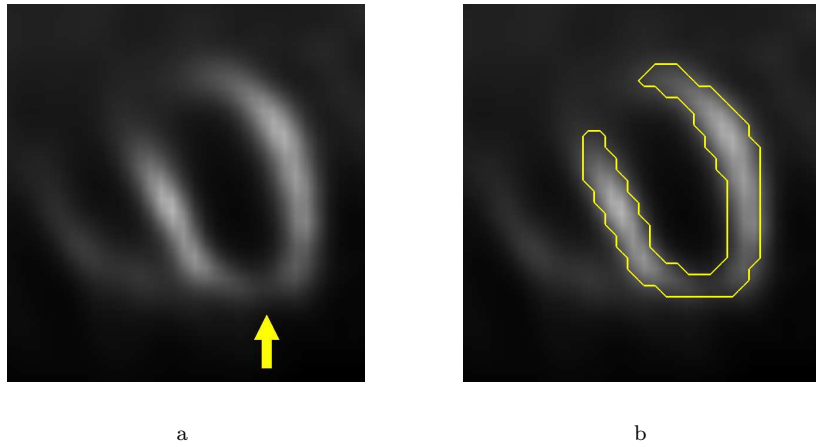


Abbildung 1.2: SPECT-Schichtbildaufnahme des linken Ventrikels, mit einer deutlich sichtbaren infarktbedingten Lücke in der hellen LV-Wand (Pfeil). In Bild (b) ist eine manuell erstellte Segmentierung dieses Organs zu sehen.

1.2 Aufgabenstellung

In humanen 3D-SPECT-Aufnahmen soll die anatomische Segmentierung des linken Ventrikels möglichst automatisch erfolgen. Dabei soll mit Blick auf die erforderliche Einbringung von Hintergrundwissens über die Form des linken Ventrikels ein modellbasierter Ansatz gewählt werden, der stabil auf dreidimensionalen Daten arbeiten muß. Ein wichtiger Punkt bei der Erstellung dieses Segmentierungsverfahren soll in diesem Zusammenhang die Berücksichtigung geeigneter Möglichkeiten für interaktive Eingriffe bzw. Korrekturen sein.

1.3 Aufbau der Arbeit

Kapitel 1 beschreibt Rahmen, Motivation, Aufgabenstellung und Aufbau der Arbeit.

Kapitel 2 untersucht ausführlich die bestehende Literatur im Umfeld des Gebietes dieser Arbeit und gibt getrennt nach methodischen und praktischen Gesichtspunkten einen Überblick über relevante Vorarbeiten.

Kapitel 3 erläutert die Grundlagen der für diese Arbeit relevanten Feder-Masse-Modelle zur Segmentierung. Dabei liegt neben der Modelldynamik ein besonderer Schwerpunkt auf den Sensoren dieser Modelle.

Kapitel 4 stellt den ersten zentralen theoretischen Komplex dieser Arbeit dar und entwickelt bedeutende Erweiterungen der Dynamik expliziter dynamischer Modelle, die für die vorliegende Aufgabenstellung wichtig sind. So werden unter anderem eine neue Stabilisierungskraft eingeführt und die Modelle um die Fähigkeit zur expliziten dreidimensionalen Rotation ergänzt. Nicht zuletzt wird ein verbesserter Sensortyp vorgestellt, der auf dieser erweiterten Modellmechanik aufbaut.

Kapitel 5 ist wiederum theoretisch orientiert und beschäftigt sich eingehend mit der Generierung expliziter dynamischer 3D-Modelle aus Beispielen von Segmentierungen. Auch die automatische Modellerzeugung ist eine Neuentwicklung dieser Arbeit und bildet die Grundlage für den erfolgreichen Einsatz dieser Modelle im Rahmen der vorliegenden Arbeit.

Kapitel 6 bildet den Abschluß des theoretischen Teils der Arbeit und legt aufbauend auf die vorangegangenen Kapitel die Grundlagen für eine komplett automatische Mehrphasensegmentierung. Dabei finden neben Verfahren der Mustererkennung zur Bestimmung des Ausgangspunktes der Modelanpassung auch die Möglichkeiten einer schlüssigen Interaktion Beachtung.

Kapitel 7 gibt einen Einblick in interessante Aspekte der Implementation der Verfahren dieser Arbeit. Wichtig sind dabei vor allem Algorithmen und Datenstrukturen, die für eine Modellsimulation in Echtzeit erforderlich sind und damit bestimmte Arten der Interaktion erst ermöglichen.

Kapitel 8 stellt nun die gewonnenen Ergebnisse bei der Anwendung der in den vorangegangenen Kapiteln entwickelten theoretischen Grundlagen ausführlich dar. Neben den Stärken und Schwächen der Verfahren wird auch auf günstige Parametrisierungen eingegangen. Eine Beurteilung der Ergebnisse im Vergleich zu verschiedenen manuell erstellten Segmentierungen rundet das Kapitel ab.

Kapitel 9 beendet diese Arbeit mit einer Zusammenfassung der erreichten Ergebnisse und gibt einen Ausblick auf zukünftige Fragestellungen in diesem Themengebiet.

1.4 Anmerkungen zur Arbeit

Die vorliegende Diplomarbeit mit dem Titel "Generierung und Dynamik physikalisch basierter 3D-Modelle zur Segmentierung des linken Ventrikels in SPECT-Daten" entstand an der Otto-von-Guericke-Universität Magdeburg in der Arbeitsgruppe Bildverarbeitung in Zusammenarbeit mit unserem klinischen Partner, der Medical Imaging Research Group des Vancouver Hospital und der University of British Columbia, Kanada.

Kapitel 2

Verwandte Arbeiten

In diesem Kapitel werden verwandte Arbeiten zur Segmentierung diskutiert. Auf Grund der Aufgabenstellung befinden sich dabei vor allem modellbasierte Ansätze zur Segmentierung im Blickpunkt. Insbesondere spielt auch der für diese Arbeit relevante Fall der Segmentierung in 3D eine Rolle. Weiterhin soll auch die Generierung dynamischer 3D-Modelle Beachtung finden, da sie im dreidimensionalen Fall gegenüber dem zweidimensionalen Fall deutlich an Bedeutung gewinnt.

Über die hier nicht behandelten elementaren, nicht modellbasierten Segmentierungsverfahren geben z. B. [GONZALEZ und WOODS 1992] oder unter dem Blickwinkel medizinischer Daten [POHLE 2004] einen ausführlichen Abriß.

2.1 Modellbasierte Segmentierung

Die aus der Literatur bekannten Modelle zur Segmentierung können in Anlehnung an [AL-ZUBI 2002] unterschieden werden in:

- statistische Modelle
- strukturelle Modelle
- dynamische Modelle

Statistische Modelle versuchen die Form der zu segmentierenden Objekte samt der vorkommenden Varianz dieser Form statistisch zu modellieren. Sie brauchen daher segmentierte Beispiele dieser Objekte, um die nötigen statistischen Parameter zu berechnen.

Ein erster bedeutender Vertreter dieses Ansatzes waren die *Active Shape Models* (ASMs) ([COOTES et al. 1995]). Sie sind ein Punktverteilungsmodell, bei dem in den verschiedenen Trainingsbeispielen semantisch gleichwertige Stellen (an Objektkanten) durch korrespondierende Punkte markiert werden. Die gesetzten Punkte eines Beispiels bilden dabei einen Formvektor, der die Form des zu segmentierenden Objektes an ausgewählten Stützstellen beschreibt. Durch

Hauptkomponentenanalyse dieser Formvektoren wird danach die in den Beispielen auftretende Formvarianz ermittelt. Durch ihre ersten Komponenten wird dann der Raum der zu betrachtenden Formvariationen gebildet. Bei einer Segmentierung mit ASM's wird nun das gelernte Modell iterativ im Raum der Formvariationen so an die Daten angepaßt, daß die Stützstellen im Bild möglichst starke Kantenwerte aufweisen.

Eine wichtige Weiterentwicklung der ASMs sind die *Active Appearance Models* (AAMs) ([COOTES et al. 1998]). Sie beziehen zusätzlich zur Form der Punkteverteilung auch die normalisierten Grauwerte an den Stützstellen mit ein und lernen somit nicht nur die Form- sondern in Grundzügen auch die Textureigenschaften der zu segmentierenden Objekte.

Im Gegensatz zu den statistischen Modellen, die auf viele Trainingsbeispiele angewiesen sind, benötigen die *strukturellen Modelle* im allgemeinen keine Trainingsdaten, da sie direkt konstruiert werden. Sie segmentieren auch nicht im eigentlichen Sinne, sondern führen eine strukturelle Klassifikation durch und können damit den modellbasierten Teil einer gesamten Segmentierung darstellen. Diese Klassifikation stützt sich dabei im allgemeinen auf den strukturellen Aufbau aus Teilkomponenten.

Ein klassischer Vertreter dieser Kategorie von Modellen sind *L-Systeme* ([PRUSINKIEWICZ und LINDENMAYER 1990]). Durch sie können Axial-Bäume (z. B. durch Skelettierung gewonnen) strukturell auf die Zugehörigkeit zu einer Sprache untersucht werden. Das L-System fungiert dabei als kontextsensitive Grammatik, die diese Sprache mittels eines parallelen Produktionensystems erzeugt. Die eigentliche Bedeutung der L-Systeme liegt aber in der Erzeugung und Beschreibung komplexer Pflanzenwuchsstrukturen.

Ein neuerer struktureller Ansatz sind sogenannte *Shock Grammars* ([SIDDIQI und KIMIA 1996]). Ausgehend von der Mittel-Achsen-Repräsentation der zu untersuchenden Objekte werden von diesen Achsen Schockwellen ausgesandt, deren Treffpunkte durch ihre Art und topologische Beziehung zueinander Aufschluß über die Struktur des ursprünglichen Objektes geben. Diese Art der Beschreibung wurde deshalb gewählt, da sich der Schockgraph, also der Graph dieser Treffpunkte, gut durch eine Grammatik, die Shock Grammar, erzeugen läßt und so relativ einfach strukturelle Ähnlichkeitsmaße zwischen verschiedenen Objekten über Ähnlichkeiten in der Produktion des Schockgraphen definiert werden können.

Die letzte Kategorie bilden die *dynamischen Modelle*, die auf Grund der Aufgabenstellung im Mittelpunkt dieser Arbeit stehen. Deshalb wird diesbezüglich der Stand der Technik im folgenden Abschnitt gründlicher untersucht.

2.2 Dynamische Modelle

Dynamische Modelle sind durch physikalische, dynamische Prozesse inspiriert, die durch das Wirken verschiedener Energiekomponenten in Bewegung gesetzt werden und einem Gleichgewichtszustand zustreben. Zum einen besitzen sie dabei die Fähigkeit, sich an Bildmerkmale anzupassen, die durch eine *externe*

Energie bestimmt wird. Zum anderen lassen sie sich auf Grund ihrer jeweiligen Art und Parametrierung in bestimmter Weise verformen, was über ihre *innere Energie* gesteuert wird. Befinden sich die Modelle im Gleichgewicht bezüglich beider Energien, so verbinden sie Bildanpassung mit vorgegebenem Modellwissen. Der Schwerpunkt zwischen den beiden Einzelverhaltensweisen kann dabei gezielt durch Wichtung dieser Energien verschoben werden.

Es gibt in der umfangreichen Literatur über dynamische Modelle zur Segmentierung zwei prinzipielle Ansätze, die sich auf Grund der Art ihrer Parametrisierung unterscheiden (siehe dazu z. B. [JAIN et al. 1998]):

- explizite (Prototyp-basierte) Modelle
- implizite (analytische) Modelle

Die vorhandenen Ansätze zu beiden Arten sollen in den nächsten beiden Unterabschnitten (2.2.1 und 2.2.2) behandelt werden. Weiterhin soll in Abschnitt 2.2.3 auf eine neue, interessante Richtung eingegangen werden, bei der zusätzlich zum Modellwissen in der Konstruktion und Parametrisierung des Modells auch noch Wissen über den Anpassungsprozeß der Modelle mit Hilfe einer zusätzlichen Steuerungskomponente eingebracht wird. In Abschnitt 2.2.4 schließlich sollen gezielt dynamische Modelle auf 3D-Daten betrachtet werden, die bis jetzt in der Literatur eine untergeordnete Rolle spielen, für die vorliegende Arbeit jedoch von zentraler Bedeutung sind.

2.2.1 Explizite Modelle

Explizite dynamische Modelle verdanken ihren Namen der expliziten Art ihrer Parametrisierung. Ihre Bestandteile passen sich lokal an die Bildinformation an und können direkt parametrisiert werden. Sie ermöglichen somit ein gutes Verständnis durch den Menschen, da ihre im allgemeinen physikalisch basierte Dynamik mit den Erfahrungen der realen Welt der Benutzer korrespondiert. Diese Tatsache und die Möglichkeit der direkten Parametrisierbarkeit einzelner Teile des Modells erlauben im allgemeinen eine einfache und direkte Art der Interaktion mit diesen Modellen.

Im Rahmen der Computergrafik wurden solche Modelle das erste mal 1988 durch Terzopoulos und Fleischer genannt ([TERZOPOULOS und FLEISCHER 1988]). Sie setzten diese zur physikgetreuen Animation von verformbaren Objekten ein. Im gleichen Zug wurden diese Modelle jedoch für die Rekonstruktion / Segmentierung interessant.

Eines der ersten Beispiele expliziter dynamischer Modelle in der Bildverarbeitung waren noch im selben Jahr die *Snakes* ([KASS et al. 1988]). Zwischen zwei festgesetzten Endpunkten windet sich dieses dynamische Modell gleich einer Schlange durch das Bild, um einen Kantenzug zu segmentieren. Diese Art Kontur wird dabei durch Kanteninformation angezogen, gehorcht aber auch einer gewissen Steifheit und Richtungskontinuität, so daß kleinere Lücken im zu segmentierenden Kantenzug nicht zu Problemen führen.

Eine weitere häufig zu findende Ausprägung expliziter dynamischer Modelle sind *Balloons*. Ein frühes Beispiel ist in [COHEN 1991] zu finden. Dort stellen sie eine geschlossene Feder-Masse-Kontur dar, deren Federn Ruhelängen der Länge Null besitzen. Damit sich das Modell nicht auf einen Punkt zusammenzieht, wird zusätzlich eine Kraft eingeführt, die die Kontur wie einen aufgeblasenen Ballon auseinanderdrückt. Die Massen selbst werden von Kanteninformation in ihrer Umgebung entsprechend ihrer Stärke und Entfernung angezogen. Zusätzliche Elastizitäts- und Steifheitsparameter ermöglichen eine gewisse Kontrolle über das Modellverhalten. Durch solche Modelle können relativ robust klar begrenzte, zusammenhängende Gebiete segmentiert werden, auch wenn die Begrenzung Lücken oder Verwischungen aufweist.

So wie die Snakes und Balloons sind die meisten in der Literatur zu findenden expliziten dynamischen Modelle reine Kontur- bzw. Oberflächenmodelle. So gibt es insgesamt zwar eine große Anzahl von Ansätzen, die Feder-Masse-Netze verwenden, aber im allgemeinen werden diese Netze zur Rekonstruktion ([TERZOPOULOS und VASILESCU 1991]) oder zur Segmentierung ([DELINGETTE 1994a]) von 3D-Oberflächen eingesetzt.

Seltener sind Modelle mit zusätzlichen inneren Massen zu finden. Sie berücksichtigen durch die mit den Massen assoziierten Sensoren auch objektinnere Merkmale und können zusätzlich für Stabilität sorgen. Sie kommen z. B. in Form reiner 2D-Feder-Masse-Modelle bei Bergner ([BERGNER 2003]) vor, der verschiedene Sensortypen einsetzt, um über Anziehungskräfte das Streben bestimmter Massen zu spezifischen, lokalen Bildattributen zu erreichen und somit einen relativ hohen Grad an Modellwissen in seinen Modellen verankert.

Auch im Ansatz der *Active Blobs* ([SCLAROFF und ISIDORO 1998]) sind innere Massen zu finden. Durch sie und die damit verbundene innere Vernetzung entstehen Zellen, die die zu verfolgende Bildinformation entsprechend verzerren und versuchen zu registrieren. Somit werden hier Form und Texturinformationen zur Bildverfolgung verwendet. Active Blobs nutzen für ihre aufwendigen Berechnungen hardware-beschleunigte Texturoperationen der modernen Grafikkarten und sind daher ein 2D-Ansatz.

2.2.2 Implizite Modelle

Der Häufigkeit ihres Vorkommens in der Literatur nach zu urteilen, scheinen *implizite dynamische Modelle* weniger verbreitet zu sein als explizite. Das mag einer gewissen Unhandlichkeit geschuldet sein, da ihre implizite Parametrisierung sich im allgemeinen global auswirkt und somit schwer zu kontrollieren ist. Andererseits garantieren die oft wenigen benötigten Parameter im allgemeinen Robustheit und Modelltreue, ermöglichen aber im Gegenzug auch nur einfache Modelle. Daher sind solche Ansätze vor allem für einfache generelle Objektbeschreibungen und höherdimensionale Einsätze geeignet, in denen sich explizite Modelle oft instabil verhalten.

Ein exemplarisches Beispiel ist [KEREN et al. 1992]. Dort werden mit impliziten Polynomen höheren Grades Objekte, die durch bestimmte Bildpunkte gegeben sind, angenähert. Durch ein auf den Polynomkoeffizienten basierendes

Ähnlichkeitsmaß (Mahalanobis-Distanz) kann die Ähnlichkeit eines Objektes zu anderen Objekten oder zu einer Klasse über die angepaßten Polygone ermittelt werden. Auf Grund der Natur der Polynome ist der Ansatz sehr robust und kann leicht auf eine Invarianz gegenüber affinen Transformationen erweitert werden.

An dieser Stelle sei auf [JAIN et al. 1998] verwiesen, die eine gute Zusammenstellung von Beispielen nicht nur für explizite, sondern auch implizite dynamische Modelle aufführen.

2.2.3 Modelle mit Steuerungskomponenten

In jüngster Zeit wurden dynamische Modelle in [HAMARNEH et al. 2001] um eine *Steuerungskomponente* erweitert. Diese ermöglicht es den dynamischen Modellen, ihr Verhalten während des Segmentierungsprozesses zu ändern, um z. B. gefunden Merkmalen im Rahmen eines vorher eingegebenen *Segmentierungsplanes* gerecht zu werden. Durch die Verankerung von Wissen über den Ablauf des Segmentierungsprozesses im Modell kann ein recht robustes, autonomes Modellverhalten erreicht werden.

Den Ansatz einer Steuerungskomponente für dynamische Modelle greift Bergner in seiner Diplomarbeit zur automatischen Klassifikation von Ameisen in fotografischen Aufnahmen ([BERGNER 2003]) auf, interpretiert ihn jedoch teilweise anders. So wird zwar eine Art Segmentierungsplan in Form sequentieller Segmentierung einzelner Strukturen der Ameisen benutzt, wobei nachfolgend zu segmentierende Strukturen mittels Erwartungskarten an wahrscheinlichen Positionen relativ zu bereits segmentierten Strukturen gesucht werden. Anstatt eines einzigen Modells benutzt er jedoch eine Population von Modellen im Rahmen einer stochastischen Suche und forciert über eine Steuerungskomponente die Suche in vielversprechenden Regionen. Um die dafür nötige Einschätzung der Güte eines Modells vornehmen zu können, wird eine Fitness-Funktion eingeführt, die die Qualität der Anpassung eines einzelnen Modells beurteilen soll.

Da es sich bei dieser Arbeit primär um eine Klassifikationsaufgabe handelt, führt der Ansatz zu guten Ergebnissen. Für reine Segmentierungsaufgaben könnte die aufwendige, recht unstrukturierte Suche jedoch ungeeignet sein, da sie mehr auf Objektfindung statt auf genaue Objektbegrenzung setzt. Zudem hat diese Suche die Aufgabe, die Nachteile der recht starren Modelldynamik zu kompensieren und wäre bei flexibleren, aber doch stabilen Modellen nicht in diesem Umfang nötig.

2.2.4 Modelle für dreidimensionale Daten

Die Segmentierung mit dynamischen Modellen in dreidimensionalen Daten stellt bis jetzt noch nicht den Standardfall in der Literatur dar, findet aber zunehmend Beachtung, so auch in dieser Arbeit. Die Modelle für den zweidimensionalen Fall können nicht alle direkt in den dreidimensionalen Fall übertragen werden, weil letzterer gewisse Probleme mit sich bringt:

- Der Suchraum vergrößert sich bedeutend.

- Viele, vor allem explizite dynamische Modelle weisen in 3D auf Grund des hinzukommenden Freiheitsgrades eine deutlich erhöhte Forminstabilität auf.

Auf Grund dieser Probleme sind gerade implizite Modelle eine geeignete Wahl für den dreidimensionalen Fall, da sie das Problem der Forminstabilität durch ihre Parametrisierungsart verringern. Ein gutes Beispiel für dieses Verhalten bilden die Arbeiten zur dreidimensionalen Segmentierung mit Superquadriken ([BARDINET et al. 1998]). Superquadriken sind implizite Oberflächen, die trotz ihrer lediglich fünf freien Parameter komplexe Objekte, wie Würfel, Kugeln, Ellipsoide, Oktaeder, etc., approximieren können. Sie sind somit ideal geeignet, Objekte zu segmentieren, die zumindest elementaren Symmetriebedingungen genügen, wie z. B. der LV, den Wegner ([POHLE et al. 2004]) damit in 4D-SPECT-Daten segmentiert. Da der LV auf diese Weise zwar stabil und einfach, aber nur grob beschrieben werden kann, wird hier im Anschluß noch eine Formdeformation der angepaßten impliziten Oberfläche zur Feinsegmentierung vorgenommen. Trotz allem aber bleiben die in Abschnitt 2.2.2 genannten Nachteile impliziter Modelle bestehen.

Daher gibt es auch explizite Ansätze in 3D, die jedoch alle mit dem Problem der Forminstabilität zu kämpfen haben und dahingehend spezielle Lösungen anbieten. Verbreitet ist dabei das auf den dreidimensionalen Fall erweiterte Balloonmodell. So nutzt z. B. Konrad-Verse in seiner Diplomarbeit ([KONRAD-VERSE 2004]) ein Feder-Masse-Oberflächennetz zur Tumorsegmentierung. Zur Wahrung der Stabilität findet dabei eine Expansionskraft im Zusammenspiel mit einer Art Krümmungswiderstand Anwendung. Eine ähnliche Expansionskraft wurde ebenfalls bereits in [MCINERNEY und TERZOPOULOS 1993] für ein dreidimensionales Finite-Elemente-Modell benutzt wurde.

Ein ähnlicher Ansatz sind die *Simplex Meshes* ([DELINGETTE 1994b]), da auch sie ein 3D-Feder-Masse-Oberflächennetz sind. Dieses zur Objektrekonstruktion eingesetzte Netz ist dreiregulär (also dual zu einem triangulierten Oberflächennetz) und ein besonderer Schwerpunkt dieses Ansatzes liegt auf der topologischen Adaption des Netzes. Expansionskräfte kommen hier nicht zum Einsatz, sondern die Massen des Modells bewegen sich zielgerichtet auf die nächsten Datenpunkte zu, was bei geeigneter Initialisierung ([DELINGETTE 1998] geht darauf noch einmal gesondert ein) zu einer Quasi-Stabilität des dynamischen Verhaltens führt. In den aufgeführten Beispielen wurden dabei die initialen Simplex Meshes in ausreichender Größe jeweils um die zu rekonstruierenden Objekte herum positioniert und zogen sich dann um sie zusammen. In Oberflächenregionen erhöhter Komplexität wies das angepaßte Modell ebenfalls eine erhöhte Netzdichte auf, um die Oberflächen dort adäquat zu beschreiben.

Feder-Masse-Modelle, die eine bestimmte Form und Struktur besitzen, scheinen bisher nicht erfolgreich auf 3D-Daten verwendet worden zu sein. Vermutlich scheint das Problem der Instabilität der Dynamik (Kollapse, Verdrehungen, etc.) gepaart mit der oftmals langsamen Simulation schwerwiegender als eventuelle

Vorteile in Hinsicht auf eingebrachtes Modellwissen und Interaktion zu sein.

2.2.5 Zusammenfassung

Am weitesten verbreitet sind explizite dynamische Modelle. Sie bieten gute Interaktionsmöglichkeiten, haben jedoch im für diese Arbeit wichtigen dreidimensionalen Fall schwerwiegende Probleme bei der Stabilität. Im Vergleich dazu bilden im 3D-Fall gerade die impliziten und sonst eher weniger verbreiteten Modelle eine Alternative, haben jedoch den Nachteil schwieriger Kodierung von erweitertem Modellwissen und schlechter Interaktionsmöglichkeiten, die auch bei automatisch angelegten Verfahren einen deutlichen Nachteil bedeuten, da sie so jeglicher Flexibilität beraubt sind.

Durch die in Abschnitt 2.2.3 beschriebenen Steuerungskomponenten kann zusätzliches Modellwissen über den Ablauf der Segmentierung im Modell verankert werden. Darüber hinausgehende extensive Suchverfahren bei der Segmentierung scheinen nur in speziellen Anwendungsfällen von Vorteil zu sein, in denen die Lage des Objektes überhaupt nicht klar oder die Modelldynamik zu starr ist.

Wünschenwert im Sinne der gegebenen Aufgabenstellung wäre also ein stabiles, schnelles explizites dynamisches 3D-Formmodell oder ein implizites mit deutlich verbesserter Handhabung, sprich Art der Parametrisierung. Da aber die Eigenschaften von letzterem gerade aus der Art der Parametrisierung resultieren, scheint ein erweitertes explizites Modell ein realistischeres Ziel zu sein. Weiterhin wäre auch eine Art Steuerungskomponente sinnvoll, die Wissen über die notwendigen Schritte einer kompletten Segmentierung integriert, falls das nötig ist.

2.3 Modellgenerierung

Nachdem nun ein Überblick über Ansätze zur modellbasierten Segmentierung, mit speziellem Fokus auf den dynamischen Modellen, gegeben wurde, soll im folgenden untersucht werden, wie solche Modelle gegenwärtig erzeugt werden.

Eine automatische Generierung von 3D-Modellen beschränkt sich in der Literatur auf die initiale Anpassung reiner Oberflächennetze an zu segmentierende bzw. zu rekonstruierende Objekte. So behandelt [DELINGETTE 1998] die Erzeugung und Initialisierung von Simplex Meshes unter der Vorgabe topologischer Korrektheit, Objektnähe und auch Modelleinfachheit. Es werden allerdings kein konkretes Formwissen und keine spezifischen Parameter, die über eine Initialisierung hinausgehen, in dem Modell verankert, da mehr in dem in Abschnitt 2.2.4 bereits diskutierten Ansatz auch nicht erforderlich ist.

[ALGORRI und SCHMITT 1995] behandelt in diesem Zusammenhang die Rekonstruktion eines Objektes aus einer Punktwolke. Dabei wird ein Voxelraster über die Daten gelegt und die Voxel, die Datenpunkte (der Objektkontur) enthalten, bilden eine initiale Oberflächenstruktur, deren Triangulierung den Ausgangspunkt für ein dynamisches Modell bildet, mit dessen Hilfe die Oberflächen-

struktur feiner an die Daten angepaßt wird. Wie auch schon bei der initialen Erzeugung der Simplex Meshes wird hier kein Modellwissen dauerhaft in den Modellen verankert, auf das auch bei der Segmentierung bzw. Rekonstruktion anderer, ähnlich gearteter Strukturen zurückgegriffen werden könnte.

Im Themenfeld der Modellinitialisierung ist auch die Diplomarbeit von Benedix ([BENEDIX 2004]) angelegt. Jedoch wird dort keine initiale Position oder Form ermittelt, sondern es wird versucht, die Masse- und Feder-Gewichtungen von 2D-Feder-Masse-Modellen bezüglich einer Segmentierung auf Beispieldatensätzen optimal zu wählen.

Eine komplette automatische Generierung von expliziten dynamischen 3D-Modellen, bei der sowohl Struktur, Form als auch die Parametrisierung erzeugt werden, ist in der Literatur nicht zu finden. Sie wäre allerdings gerade für diese Themenstellung sehr wünschenswert, da im dreidimensionalen Fall die nötige Modellspezifikation schon bei relativ einfachen Modellen sehr umfangreich wird und sich damit bei manueller Erzeugung eine sehr hohe, meist nicht überwindbare Hürde für den Einsatz von solcher Modelle zur Segmentierung von 3D-Daten ergibt. Zur Zeit muß sie aber, wie Pohle in [POHLE 2004] ausführt, auf eben diese Weise durch Benutzerinteraktion durchgeführt werden, weshalb in der Regel nur verhältnismäßig einfache Modelle verwendet werden, deren Anpassung an die Daten oft ungenau oder fehlerhaft ist.

2.4 Segmentierung des linken Ventrikels in SPECT-Daten

Über die Segmentierung des LV wurde in der Literatur viel geschrieben. In der Diplomarbeit von Wegner ([WEGNER 2003]) ist ein guter Abriß über die bereits eingesetzten Verfahren zu finden. Nur ein Bruchteil dieser Literatur beschäftigt sich jedoch mit der Segmentierung des LV in 3D-SPECT-Daten, die gewisse Besonderheiten aufweist.

- (derzeit noch) keine hohe Auflösung
- geringer Kontrast
- hohes Signal-Rausch-Verhältnis
- das Signal wird innerhalb eines Zeitraumes integriert, in dem sich das Herz bewegt, so daß seine Kontur verschwimmt
- funktionelle Bildgebung

SPECT-Daten haben also eine geringe Bildqualität. Die reine Bildinformation ist nicht sehr genau und deutlich. Das alles verlangt nach einem robusten Verfahren, das alle zur Verfügung stehenden Informationen im Bild ausnutzt, also idealerweise nach einem echten 3D-Segmentierungsverfahren, da sonst z. B. bei schichtweisem Vorgehen Kontextinformation ungenutzt bleibt. Nicht zuletzt

2.4. SEGMENTIERUNG DES LINKEN VENTRIKELS IN SPECT-DATEN¹³

sollte auf Grund wahrscheinlicher Uneindeutigkeiten im Bild eine korrigierende Interaktion gerade hier möglich sein.

Hinzu kommt das Problem, daß im Falle einer anatomischen Segmentierungsaufgabe, wie sie auch in dieser Arbeit vorkommt, die Anatomie nicht zwangsläufig adäquat durch die abgebildete Funktion repräsentiert wird. Diese Diskrepanz kann nur durch zusätzlich eingebrachtes Modellwissen überwunden werden, so daß ein modellbasierter Ansatz mit adäquaten Modellierungsmöglichkeiten in solch einem Fall unbedingt erforderlich ist.

[FERNANDEZ-MALOIGNE et al. 2000] bietet einen breiten Überblick über bisherige Versuche der LV-Segmentierung in SPECT-Daten. Ein Fazit ist, daß viele Ansätze (schwellwert-basierte, regionen-basierte, etc.) auf Grund fehlender Modellinformationen schlechte Ergebnisse liefern. Die wenigen, die Modellwissen benutzen, benötigen immer noch mehr oder weniger starken Interaktionsaufwand oder besitzen so starre, vereinfachende Modelle, daß der LV nicht mehr angemessen segmentiert werden kann. Zudem scheint es, als ob das Problem der anatomisch korrekten Segmentierung großer infarktbedingter Lücken in diesem Aufsatz einfach ignoriert wird.

Der Ansatz von Wegner ([WEGNER 2003, POHLE et al. 2004], vergleiche Abschnitt 2.2.4) versucht dieses Problem durch die Verwendung von Superquadriken mit anschließender Freiformdeformation anzugehen. Diese impliziten dynamischen 3D-Modelle wurden zuvor schon für die zeitliche Verfolgung des bereits segmentierten LV in 4D-SPECT-Daten durch Bardin et al. ([BARDINET et al. 1996]) benutzt. Durch die Verwendung dieses impliziten Modells gibt es jedoch die für implizite Modelle typischen Probleme (siehe Abschnitt 2.2.2). Zudem wird auch hier das Problem einer modellgetreuen Segmentierung von Lücken im LV umgangen, da die LV-Kontur lediglich durch Kanteninformationen in einem sehr engen Suchkorridor (Epikard: im Umkreis von 4 Pixeln eines Ausgangspunktes in einer (!) Schicht, Endokard: einer von 3 Pixeln) gebildet wird. Dadurch entstehende dreidimensionale Inkohärenzen werden durch Glättungen beseitigt.

Eine Segmentierung des LV in 3D-SPECT-Daten, die den in der schlechten Bildqualität dieses Aufnahmeverfahrens begründeten Problemen gerecht wird, wurde bereits mehrfach versucht. Am erfolgreichsten wurden dabei modellbasierte Verfahren eingesetzt, um die Ungenauigkeiten in den Daten kompensieren. Eine automatische Segmentierung, die auch Lücken im funktionellen Bild durch anatomisches Modellwissen korrekt behandelt, gibt es jedoch nicht.

Kapitel 3

Grundlagen von Feder-Masse-Modellen

In diesem Kapitel soll die Dynamik der, auf Grund der Aufgabenstellung (Abschnitt 1.2) für diese Arbeit wegen ihrer guten Interaktionsmöglichkeiten (siehe Abschnitt 2.2.5) zentralen, physikalisch basierten 3D-Modelle zur Segmentierung samt deren mathematischer Grundlagen näher betrachtet werden. Neben der Vorstellung der bekannten Feder-Masse-Modelle liegt der Schwerpunkt dabei auf den zugehörigen Sensoren.

3.1 Modelldynamik

Feder-Masse-Modelle sind dynamische, physikalisch basierte Modelle und aus der Literatur zur Bildverarbeitung bekannt (z. B. [BERGNER et al. 2004]). Sie sind ein theoretisches Modell, das ein dynamisches System von *Massepunkten* darstellt, die durch *Federn* verbunden sind. Neben den vorgegebenen physikalischen Parametern, also den Massen m_i der einzelnen Massepunkte i und Ruhelängen $l_{0,i,j}$ und Federkonstanten $k_{i,j}$ (Kraft pro Längenänderung) der einzelnen Federn zwischen den Massepunkten i und j , spielt dabei die Vernetzungstopologie der Massen durch die Federn eine Rolle.

Die Dynamik eines solchen Systems kann mittels der Newton'schen Mechanik dargestellt werden. Die Bewegungen der Massepunkte hängen dabei lediglich von Kräften ab, die auf die Massepunkte wirken. In diesem theoretischen, idealen System sind das nur die Federkräfte $\vec{F}_{i,j}$ (Gleichung 3.1), die durch elastisch verformte Federn auf die mit ihnen verbundenen Massepunkte in Richtung der Federn wirken und deren Größe letztendlich nur von den Positionen der Federsensoren i und j (hier dargestellt durch die Ortsvektoren s_i und s_j), die räumlich mit den Massepunkten i und j assoziiert sind, abhängt. In dieser Arbeit wird immer von genau einem Sensor pro Masse ausgegangen, was jedoch keine Einschränkung darstellt, da null Sensoren bzw. mehr als einer einfach als ein offline-geschalteter Sensor (siehe zum Thema offline-Status von Sensoren

Abschnitt 3.2) bzw. ein zusammengesetzter komplexer Sensor aufgefaßt werden können.

$$\vec{F}_{ij} = k_{ij} \cdot (\|\vec{s}_j - \vec{s}_i\| - l_{0ij}) \cdot \frac{\vec{s}_j - \vec{s}_i}{\|\vec{s}_j - \vec{s}_i\|} \quad (3.1)$$

Wie in der Literatur üblich (z. B. in [KASS et al. 1988, COHEN 1991]), soll auch hier die Modelldynamik nicht durch Differentialgleichungen exakt berechnet werden, sondern aus Gründen des geringeren Berechnungsaufwandes und der ausreichenden Genauigkeit in diskreten Zeitschritten mit dem Abstand Δt simuliert werden. Dabei wird ausgehend von der Geschwindigkeit \vec{v}_i des Sensors i zum Zeitpunkt t seine Geschwindigkeit zum Zeitpunkt $t + \Delta t$ berechnet. Gleichung 3.2 zeigt die entsprechende, die Federkräfte einbeziehende Bewegungsgleichung, in der über alle Sensoren j summiert wird, die mit dem Sensor i verbunden sind. Diese Notation der Summation über verbundene, auch als *benachbart* bezeichnete Sensoren, soll auch im Fortlauf dieser Arbeit beibehalten werden.

$$\vec{v}_{i_{t+\Delta t}} = \vec{v}_i + \frac{\sum_j \vec{F}_{ij}}{m_i} \cdot \Delta t \quad (3.2)$$

Es empfiehlt sich die Hinzunahme einer allgemeinen Dämpfungswirkung, die lediglich durch die zeitlich diskrete Simulation des Systems motiviert ist. Auf diese Weise werden Probleme beim Erreichen eines stabiles Gleichgewichtszustandes (z. B. ein nur durch die nichtkontinuierliche Simulation verursachtes Aufschaukeln) deutlich reduziert. Gleichung 3.3 zeigt die erweiterte Bewegungsgleichung, bei der die Geschwindigkeiten pro Zeitschritt um einen Anteil von d gedämpft werden.

$$\vec{v}_{i_{t+\Delta t}} = (\vec{v}_i + \frac{\sum_j \vec{F}_{ij}}{m_i} \cdot \Delta t) \cdot (1 - d) \quad (3.3)$$

Die Dämpfungswirkung bildet die *dynamischen Kräfte*, die nicht explizit als Kräfte formuliert werden müssen. Sie treten nur während der Bewegung des Modells vor Erreichen eines Gleichgewichtszustands aller Kräfte auf. Die Federkräfte sind jedoch nicht zwangsläufig nur während der Bewegung präsent. Sie sollen im folgenden als *innere Kräfte* bezeichnet werden und kodieren die Forminformation eines Feder-Masse-Modells.

Durch dynamische und innere Kräfte kann das Modell natürlich noch keinen Bezug zu den Bilddaten, die es segmentieren soll, herstellen. Es ist eine weitere Kraftkomponente, die *externen Kräfte*, die gerade das tut, notwendig. Auch sie wirkt auf die Massepunkte und wird durch mit den Massepunkten assoziierten *Sensoren* generiert. Jeder Sensor i erzeugt dabei eine Kraft \vec{F}_i , die von den Bilddaten abhängt. Dabei spielen die Position des Sensors (\vec{s}_i), also des assoziierten Massepunktes, und seine Art und vorgegebene Parametrisierung eine Rolle. Durch die externen Kräfte erweitert sich Gleichung 3.3 zu Gleichung 3.4

$$\vec{v}_{i_t+\Delta t} = (\vec{v}_{i_t} + \frac{\sum_j \vec{F}_{ij} + \vec{F}_i}{m_i} \cdot \Delta t) \cdot (1 - d) \quad (3.4)$$

Um den Einfluß der verschiedenen Kraftkomponenten in Gleichung 3.4 steuern und damit auch die inneren gegen die äußeren Kräfte austarieren zu können, ist es wichtig, diese Bewegungsgleichung um Wichtungsfaktoren zu erweitern zu Gleichung 3.5, wobei w_f die Federkraftkomponente wichtet und $w_s(i)$ die Sensorkraftkomponente. Der letzte Faktor ist dabei abhängig von i insofern, als daß verschiedene Sensortypen auch verschiedene Wichtungen bzw. untereinander unabhängige Wichtungsmöglichkeiten nahelegen (siehe Abschnitt 6.1.3 dazu).

$$\vec{v}_{i_t+\Delta t} = (\vec{v}_{i_t} + \frac{w_f \cdot \sum_j \vec{F}_{ij} + w_s(i) \cdot \vec{F}_i}{m_i} \cdot \Delta t) \cdot (1 - d) \quad (3.5)$$

Wenn das Modell nun einen Gleichgewichtszustand gemäß Gleichung 3.5 erreicht und somit keine Bewegung mehr stattfindet, ist seine Anpassung an die Daten beendet. Auf Grund der durch die inneren Kräfte vorgegebenen Form (Modellwissen) und der durch die externen Kräfte eingeflossenen Dateninformation wurde somit eine modellbasierte Segmentierung der Daten durchgeführt.

3.2 Sensoren

Es gibt eine Fülle verschiedener Arten von Sensoren i (z. B. Intensitätssensoren, Kontursensoren, etc., siehe [BERGNER 2003] für weitere Beispiele), aber etwas ist ihnen allen gemeinsam: Sie haben immer eine aktuelle Position \vec{s}_i im Datensatzkoordinatensystem und sie erzeugen eine Kraft \vec{F}_i , die an sie, und damit an die assoziierte, gleich positionierte Masse, angreift.

Weiterhin ist es sinnvoll, Sensoren mit verschiedenen, dynamisch veränderbaren Eigenschaften auszustatten, die die Steuerung des dynamischen Modellverhaltens erleichtern. Solche Eigenschaften sind:

Aktivität (aktiv / inaktiv): *Inaktive* Sensoren gelten samt ihrer Massen als temporär gelöscht. Sowohl sie, die assoziierten Massen, als auch die angrenzenden Federn spielen keine Rolle mehr für die Modelldynamik und werden auch nicht mitbewegt. Auf diese Weise können Teile eines Modells komplett isoliert betrachtet werden.

Aktive Sensoren verhalten sich wie normale zum Modell gehörige Sensoren.

Meßverhalten (online / offline): Sensoren, die *offline* geschaltet sind, erzeugen keine Sensorkräfte mehr. Sie werden jedoch mitbewegt und ihre Massen und verbundenen Federn sind für die Modelldynamik weiterhin von Bedeutung.

Sensoren, die *online* sind, erzeugen Sensorkräfte wie gewohnt.

Verschiebbarkeit (frei / fest): Sensoren, die als *fest* markiert wurden, bewegen sich mitsamt ihrer Massen nicht mehr, können jedoch noch rotieren (siehe

Abschnitt 4.2). Sie werden weiterhin in der Modelldynamik berücksichtigt. *Freie* Sensoren unterliegen dieser Beschränkung nicht.

Mittels dieser Eigenschaften können Teile des gesamten Modells separat angepaßt und dann fixiert werden. Weiterhin ist es möglich, bei Teilen des Modells die Anpassung nur vom Modellwissen abhängig zu machen und so auf widersprüchliche bzw. falsche Werte in den Daten zu reagieren. Alles in allem wird so die in Abschnitt 2.2.3 beschriebene Idee eines Segmentierungsplanes unterstützt, die in Kapitel 6 wieder aufgegriffen wird.

Sensoren können natürlich die externen Kräfte auf verschiedenste Arten generieren. In dieser Arbeit spielt aber nur eine Art der Kraftgenerierung eine Rolle, nämlich über die Bildung von Gradienten numerischer Eigenschaften T der Daten, z. B. über den Datensatz D selbst, also seine Intensität, oder über sein Konturbild K (siehe Gleichung 3.6), also seine Konturstärkeinformationen. In dieser Arbeit werden nur die in diesem Beispiel genannten Sensoren, also Intensitäts- und Kontursensoren, Verwendung finden, da sie für die Anpassung von Modellbereichen und -konturen auf den gegebenen einkanaligen Daten ausreichend sind.

$$K = \|\nabla D\| \quad (3.6)$$

Bei dieser Art der Kräftebildung wird der Sensor lokal in Richtung des Minimums bzw. Maximums einer durch diese numerischen Eigenschaften (z. B. Intensität, Konturstärke, etc.) gegebenen Zielfunktion gezogen. Dieser Anziehung wird durch die von ihm generierte Kraft \vec{F}_i (Gleichung 3.7) mit entsprechender Richtung und Stärke Ausdruck verliehen. Dabei besitzt jeder Sensor eine eigene Sensorkonstante o_i (Kraft pro Gradientengröße), die den Betrag der Kraft individuell einstellt.

$$\vec{F}_i = o_i \cdot \nabla T \quad (3.7)$$

Damit diese Sensorkonstante einen kontrollierbare Bedeutung bekommt, ist es nötig zu wissen, welcher Gradientenbetrag auf welche Konturstärke bzw. -bedeutung hindeutet. Um die Bedeutung eines Gradientenbetrages unter den verschiedenen Datensätzen D einheitlich zu gestalten, die gerade im Falle der vorliegenden SPECT-Daten deutlich verschiedene Maximalintensitäten aufweisen (siehe Abschnitt 8.1), können diese normalisiert werden (zu D_{norm}), so daß sie einen bestimmten, untereinander gleichen Maximalwert besitzen (In dieser Arbeit ist dieser Wert 1.). Dadurch ändert sich Gleichung 3.6 zu Gleichung 3.8.

$$K = \|\nabla D_{\text{norm}}\| \quad (3.8)$$

Normalerweise entsprechen die Intensitätswerte des LV ungefähr den Maximalintensitäten seines Datensatzes, so daß durch diese Datensatznormalisierung eine Bedeutungsangleichung der Konturstärkeinformation (also des zugehörigen Gradientenbetrages) erreicht wird. Im Falle extrem intensitätsstarker Artefakte in den Datensätzen kann diese Bedeutungsangleichung jedoch nicht mehr mittels der vorgeschlagenen Normalisierung erreicht werden und sollte durch eine

abgewandelte Normalisierung vorgenommen, bei der vor dem eigentlichen Normalisierungsschritt erst alle zu großen Intensitäten auf ein Niveau abgeschwächt werden, das dem üblichen Maximalintensitätsniveau des LV in den Datensätzen entspricht.

Zur Bestimmung der Richtung und Stärke der Kraft zur adäquaten Annäherung des Sensors an ein solches Minimum bzw. Maximum wird der Gradient an der aktuellen Position des Sensors im Datensatz verwendet. Seine Richtung (bzw. deren Umkehrung) und sein Betrag bestimmen die zu generierende Kraft.

Da die entsprechende Zielfunktion direkt oder indirekt lediglich an Stützstellen durch den diskreten Datensatz gegeben ist, kann der Gradient auf ihr nur numerisch angenähert werden. Zu diesem Zweck wurde eine Reihe von Operatoren (bestehend aus Faltungsmasken K_x , K_y , K_z usw. für jede Dimension des Datensatzes) entwickelt, mit denen jeweils die partiellen Ableitungen dieser durch den Datensatz gestützten Zielfunktion T durch Faltung (Operator: $*$) approximiert werden können. Gleichung 3.9 zeigt die Berechnung der partiellen Ableitungen $\frac{\partial T}{\partial x}$ einer solchen Zielfunktion z in Richtung der X-Achse.

$$\frac{\partial T}{\partial x} = T * K_x \quad (3.9)$$

Die gängigen Gradientenoperatoren (Sobel, Canny, etc.) sind jedoch nicht optimal bezüglich einer möglichst kleinen Richtungsabweichung zwischen dem berechneten Gradienten und dem echten Gradienten, also dem Gradienten der Funktion, die durch die gegebenen Daten direkt oder indirekt gestützt wird (siehe [JÄHNE 2002] dazu). Jähne et al. präsentieren in [JÄHNE et al. 1999] Gradientenoperatoren, die für verschiedene Größen von Faltungsmasken bezüglich dieses Kriteriums optimal sind und durch Optimierung ermittelt wurden. Sie sollen daher im weiteren in dieser Arbeit benutzt werden. Die gewünschte Sensorreichweite kann dabei durch die Wahl eines solchen Operators mit entsprechender Größe bestimmt werden. In Gleichung 3.10 ist exemplarisch die Faltungsmaske K_x eines Operators der Größe 5 für 3 Dimensionen gezeigt, der einer Sensorreichweite von 2 entlang der Koordinatenachsen entspricht. Sie entsteht durch sukzessive Faltung einer eindimensionalen (in Ableitungsrichtung X liegenden) Ableitungsmaske (Differenzenmaske) A_x mit eindimensionalen (in den übrigen Richtungen Y und Z liegenden) Glättungsmasken G_y und G_z , die natürlich der nötigen Symmetrie wegen bis auf ihre Richtung identisch sind, da alle Nichtableitungsrichtungen gleichbehandelt werden sollen (für eine tiefgehendere Behandlung der Erzeugung dieser Gradientenoperatoren siehe [JÄHNE et al. 1999]).

$$\begin{aligned} K_x &= A_x * G_y * G_z \\ &= \frac{1}{256} \begin{pmatrix} -21, 38 \\ -85, 24 \\ 0 \\ 85, 24 \\ 21, 38 \end{pmatrix}^T * \frac{1}{256} \begin{pmatrix} 5, 96 \\ 61, 81 \\ 120, 46 \\ 61, 81 \\ 5, 96 \end{pmatrix} * \frac{1}{256} \begin{pmatrix} 5, 96 \\ 61, 81 \\ 120, 46 \\ 61, 81 \\ 5, 96 \end{pmatrix}_z \end{aligned} \quad (3.10)$$

Um eine glatte Modellbewegung zu erreichen, sollten keine Sprünge in den durch die Sensoren erzeugten Kräften bei kleinen Sensorverschiebungen auftreten, wie es durch die Verwendung der Gradientenoperatoren passieren kann, die nur an den Stützstellen des Datensatzes berechnet werden können. Deshalb ist eine Interpolation der berechneten Gradienten sinnvoll. Im einfachsten Fall handelt es sich dabei um eine multilineare Interpolation. Gleichung 3.11 verdeutlicht die trilineare Variante für den dreidimensionalen Fall. Interpoliert wird dabei der Wert D_{abc} an der Position (a, b, c) zwischen den acht Stützstellen $(l, m, n), \dots, (l + 1, m + 1, n + 1)$ des Datensatzes D .

$$D_{abc} = \sum_{i=l}^{l+1} \sum_{j=m}^{m+1} \sum_{k=n}^{n+1} (1 - |i - a|) \cdot (1 - |j - b|) \cdot (1 - |k - c|) \cdot D_{ijk}$$

mit $a \in [l, l + 1) \quad \wedge \quad b \in [m, m + 1) \quad \wedge \quad c \in [n, n + 1)$ (3.11)

Kapitel 4

Erweiterte Feder-Masse-Modelle

Wie schon in Abschnitt 2.2.5 erläutert, ist ein explizites dynamisches Modell auf Grund seiner günstigen Anpassungs- und Korrektureigenschaften wünschenswert. Da ein Hauptaspekt der Themenstellung dieser Arbeit die Dreidimensionalität der gegebenen Daten ist, verbindet sich mit solch einem Modell jedoch das Problem der Instabilität der bekannten Vertreter dieser Klasse von Modellen (siehe Abschnitt 2.2.4).

In diesem Kapitel werden daher die in Kapitel 3 vorgestellten Feder-Masse-Modelle in mehrerlei Hinsicht erweitert, um unter der gegebenen Aufgabenstellung, speziell also auf 3D-Daten, nutzbringend eingesetzt werden zu können.

4.1 Torsionskräfte

Bereits Bergner erkennt in [BERGNER 2003] das Problem der Instabilität expliziter dynamischer 2D-Modelle und schlägt zur Stabilisierung eine sehr dichte Vernetzung der Modelle oder *Winkelkräfte* (die er jedoch als *torque forces* bezeichnet) vor.

Dichte Vernetzungen lösen das Problem der Instabilität (gerade in höheren Dimensionen) nur auf Kosten sehr komplexer Modelle und zudem nur indirekt, da keine Kontrolle über den Grad der Stabilität geschaffen wird, schon gar nicht auf dynamischer Basis. Konkrete Stabilisierungskräfte würden das allerdings ermöglichen.

Die von Bergner vorgestellten Winkelkräfte sind nicht solche Stabilisierungskräfte, da sie jeweils nur die Winkel zwischen zwei von einem Sensor ausgehenden Federn betrachten. Damit können in höheren Dimensionen die Winkelebenen umklappen und eine Vielzahl von Winkelkräften (maximal eine pro möglichem Paar angrenzender Federn) ist nötig, die wiederum die gleichen Probleme wie eine dichte Vernetzung aufwirft. Ein weiteres Problem liegt direkt in der Definition dieser Winkelkräfte, die eine Kraft- statt einer Drehmomentbetrachtung

beinhaltet und zudem diese Kräfte direkt an die Federelastizitäten bindet.

Die Lösung des Problems liegt in der Einführung von (normierten) Ruherichtungen $r_{0_{ij}}$ der einzelnen Federn ausgehend von jeweils einem Sensor i zu seinen benachbarten Sensoren j in Analogie zu den Ruhelängen der Federn. Auftretenden Verdrillungen der Federn, sprich Abweichungen von ihren Ruherichtungen, kann dann mittels entgegengerichteter Drehmomente entgegengewirkt werden. Diese äußern sich dann letztendlich in *Torsionskräften* $\vec{F}_{i,j}$ (Gleichung 4.1), deren Größe von der sensorspezifischen Torsionskonstante t_i (Drehmoment pro Winkel der Verdrehung) abhängt.

$$\vec{F}_{i,j} = \frac{t_i \cdot |\angle(\vec{r}_{ij}, \vec{r}_{0_{ij}})|}{\|\vec{r}_{ij}\|} \cdot \frac{\vec{n}_{ij}}{\|\vec{n}_{ij}\|} \quad \text{mit} \quad \vec{r}_{ij} = \vec{s}_j - \vec{s}_i \quad (4.1)$$

Sie greifen an den mittels Federn mit dem Sensor i im Drehpunkt der Verdrillung verbundenen Sensoren j tangential ihrer Bewegungskurve zum Ausgleich der Verdrillung in Ausgleichsrichtung an. Die Berechnung dieser Angriffsrichtung \vec{n}_{ij} zeigt Gleichung 4.2.

$$\vec{n}_{ij} = \vec{r}_{0_{ij}} - \frac{\langle \vec{r}_{ij}, \vec{r}_{0_{ij}} \rangle}{\|\vec{r}_{ij}\|^2} \cdot \vec{r}_{ij} \quad (4.2)$$

Diese Art der Stabilisierung funktioniert in beliebig hohen Dimensionen. Sie läßt über eine Wichtung w_t der Torsionskräfte eine genaue Kontrolle der Formstabilität zu und erfordert nur so viele Torsionskraftberechnungen pro Sensor, wie Federn an ihn angrenzen. Auf Grund der absoluten Ruherichtungen besteht ebenfalls keine Gefahr des gegenseitigen Verschiebens der Winkelebenen. Gleichung 4.3 zeigt in diesem Zusammenhang die um Torsionskräfte erweiterte Bewegungsgleichung.

$$\vec{v}_{i_{t+\Delta t}} = (\vec{v}_{i_t} + \frac{w_f \cdot \sum_j \vec{F}_{ij} + w_t \cdot \sum_k \vec{F}_{k,i} + w_s(i) \cdot \vec{F}_i}{m_i} \cdot \Delta t) \cdot (1 - d) \quad (4.3)$$

Die Torsionskräfte gehören zu den inneren Modellkräften, da sie Modellwissen über die Form kodieren. Bei dünner Vernetzung kann man dabei davon ausgehen, daß durch die Federn die Größe des modellierten Objektes definiert wird und durch die Federrichtungen seine Form. Somit können in diesem Fall und bei Verwendung beider Kraftkomponenten Form und Größe eines Modells bei der Anpassung individuell in ihrem Einfluß gewichtet werden.

4.2 Explizite Rotationsfähigkeit

Durch die Abhängigkeit der Torsionskräfte von den im vorhergehenden Abschnitt beschriebenen Ruherichtungen wird durch diese Ruherichtungen die Ausrichtung des Modells bestimmt. Handelt es sich um bezüglich des Datensatzes absolute Richtungen, wie es im vorhergehenden Abschnitt der Fall ist, so wird ein Drehen bzw. Rotieren des Modells blockiert. Dieses Verhalten kann in

bestimmten Anwendungsfällen, wenn z. B. die Ausrichtung des zu segmentierenden Objektes bekannt ist, von Vorteil sein.

Um andernfalls jedoch Rotationen zu ermöglichen, dürfen die Ruherichtungen nicht mehr absolut zum Datensatz aufgefaßt werden. Hier ist es sinnvoll, sie relativ zum Sensor, von dem sie ausgehen, aufzufassen, der in diesem Zusammenhang natürlich rotierbar sein muß. Wird dann der Sensor gedreht, werden die Ruherichtungen der von ihm ausgehenden Federn entsprechend mitgedreht.

Die Drehung des Sensors i sollte aus den Verdrehungen, also den aktuellen Richtungen \vec{r}_{ij} (siehe Gleichung 4.2), der von ihm ausgehenden Federn resultieren. Um den Sensor sofort in einen Gleichgewichtszustand bezüglich der Verdrehungen aller m mit ihm verbundenen Federn zu überführen, wird der Durchschnitt aller Ausgleichsdrehungen von ihm, die nötig wären, um jeweils eine Feder j zurück in ihre Ruherichtung $\vec{r}_{0_{ij}}$ zu drehen, auf ihn angewendet. Diese Ausgleichsdrehungen werden dabei mit Hilfe eines Quarternions Q_{ij} auf Grund ihrer einfachen und effizienten Darstellungs- und Verknüpfungsmöglichkeiten von Rotationen in 3D modelliert (siehe [KUIPERS 2002] für eine Einführung zu diesem Thema). Gleichung 4.4 zeigt die Konstruktion dieses Quarternions aus dem Rotationswinkel φ_{ij} (im mathematisch positiven Sinn) und der gerichteten Drehachse \vec{a}_{ij} .

$$Q_{ij} = Q(\varphi_{ij}, \vec{a}_{ij}) = \begin{pmatrix} \cos \frac{\varphi_{ij}}{2} \\ \sin \frac{\varphi_{ij}}{2} \cdot \vec{a}_{ij} \end{pmatrix} \quad \text{mit} \quad \varphi_{ij} = \angle(\vec{r}_{0_{ij}}, \vec{r}_{ij})$$

und $\vec{a}_{ij} = \vec{r}_{0_{ij}} \times \vec{r}_{ij}$ (4.4)

Die Berechnung eines Durchschnittes aller m_i Einzelrotationen eines Sensors i ist nicht trivial, da die Hintereinanderausführung beliebiger Rotationen im dreidimensionalen Raum nicht kommutativ ist. Er entspricht eher einer Überlagerung (gleichzeitigen Anwendung) des m_i -ten Teils jeder Einzelrotation, so daß für große n das Durchschnittsrotationsquarternion Q_i durch Gleichung 4.5 adäquat approximiert wird.

$$Q_i = \left(\prod_j Q\left(\frac{\varphi_{ij}}{m_i \cdot n}, \vec{a}_{ij}\right) \right)^n \quad (4.5)$$

Durch Anwendung der normierten Durchschnittsrotation $Q_{i_{\text{norm}}}$ (siehe Gleichung 4.6) auf alle Ruherichtungen $\vec{r}_{0_{ij}}$ des zu drehenden Sensors i wird dieser sofort nach den anliegenden Verdrehungen optimal ausgerichtet (bei Gleichbetrachtung aller Federn), was auch nicht seiner Modellannahme widerspricht, da ein Massepunkt im Drehzentrum ein Trägheitsmoment von Null besitzt und somit nach der Newton'schen Mechanik beliebig große Drehgeschwindigkeiten erreichen kann. Gleichung 4.7 zeigt in diesem Zusammenhang die Neubestimmung $\vec{r}_{1_{ij}}$ der Ruherichtung $\vec{r}_{0_{ij}}$.

$$Q_{i_{\text{norm}}} = \begin{pmatrix} \varphi \\ \sqrt{\frac{1-\varphi^2}{\|\vec{a}\|^2}} \cdot \vec{a} \end{pmatrix} \quad \text{mit} \quad Q_i = \begin{pmatrix} \varphi \\ \vec{a} \end{pmatrix} \quad (4.6)$$

$$\begin{pmatrix} 0 \\ \vec{r}_{1ij} \end{pmatrix} = Q_{i_{\text{norm}}} \cdot \begin{pmatrix} 0 \\ \vec{r}_{0ij} \end{pmatrix} \cdot \overline{Q_{i_{\text{norm}}}} \quad \text{mit} \quad \overline{Q_{i_{\text{norm}}}} = \overline{\begin{pmatrix} \varphi \\ \vec{a} \end{pmatrix}} = \begin{pmatrix} \varphi \\ -\vec{a} \end{pmatrix} \quad (4.7)$$

Dreht man die Sensoren individuell auf diese Weise, können sie sich je nach Grad der Vernetzung mehr oder weniger stark gegeneinander verdrehen. Das kann im Sinne einer genauen Anpassung an das zu segmentierende Objekte sinnvoll sein. Weiß man jedoch, daß das zu segmentierende Objekt solche Verdrehungen nicht erfordert bzw. möchte man sie explizit verhindern, so wäre eine Rotation des gesamten Modells sinnvoll, bei der sich die Sensoren nicht gegeneinander verdrehen können, sondern sich alle gleichförmig rotieren.

So ein Modellverhalten, also eine durch das Gesamtmodell bedingte Sensorrotation, läßt sich direkt aus der zuvor beschriebenen freien Sensorrotation ableiten. Dazu werden die Sensorrotationen wie beschrieben individuell berechnet, jedoch nicht angewendet. Angewendet wird erst im Anschluß die durchschnittliche Rotation Q aller k Sensoren (Gleichung 4.8) nach Normierung (analog zu Gleichung 4.6) auf alle Sensoren (analog zu Gleichung 4.7). So dreht sich nur das Modell als ganzes und richtet sich dabei den Restriktionen entsprechend optimal nach den Federverdrehungen aus (bei Gleichbehandlung aller Federn und verbotener Sensorverdrehung gegeneinander).

$$Q = \left(\prod_{i,j} Q\left(\frac{\varphi_{ij}}{k \cdot m_i \cdot n}, \vec{a}_{ij}\right) \right)^n \quad (4.8)$$

4.3 Konturflächen

Da die erweiterten dynamischen 3D-Modelle nicht auf zwei Dimensionen beschränkt bleiben sollen, zumal es auch die gegebenen Datensätze nicht sind, repräsentieren Kanten, also Federn, des Modells nicht zwangsläufig seine Kontur. Sie muß durch Flächen des Feder-Masse-Gittermodells (in höheren Dimensionen entsprechend Hyperflächen) explizit repräsentiert werden.

Mit Hilfe dieser *Konturflächen* können Normalen der Sensoren auf der durch diese Flächen gegebenen Modellkontur definiert werden. Idealerweise sollten diese *Sensornormalen* die Normalen der Tangentialebenen des Modells durch eben diese Sensorpunkte sein. Da das jedoch bei einem Gittermodell in den Eckpunkten nicht zu einer eindeutigen Definition führt, wird auf den Durchschnitt \vec{N}_i der n Normalen \vec{N}_{ik} der an den Sensor i angrenzenden Konturflächen des Modells zurückgegriffen, wie es Gleichung 4.9 beschreibt.

$$\vec{N}_i = \frac{1}{n} \sum_k \vec{N}_{ik} \quad (4.9)$$

Neben der Beschreibung der Modellkontur in höheren Dimensionen liegt die Bedeutung der Modellkonturflächen vor allem in der Berechnung dieser Sensornormalen, die für die richtungsgewichteten Kontursensoren (Abschnitt 4.4) eine bedeutende Rolle spielen.

Da diese Kontursensoren einer Rotation nach Abschnitt 4.2 unterliegen können, müssen ihre Sensornormalen in diesem Fall entsprechend mitrotiert werden. Für zeitweise inaktive Kontursensoren ist in diesem Zusammenhang die über die aktiven Sensoren gemittelte Rotation (Gleichung 4.8) interessant, da auch die inaktiven Kontursensoren ihre Sensornormalen möglichst konform zum aktiven Rest des Modells mitbewegen sollten, für den Fall, daß sie wieder aktiviert werden.

4.4 Richtungsgewichtete Kontursensoren

Ein häufiges Problem bei der automatischen Annäherung von Konturen durch dynamische Modelle sind Variationen im Kontrast des Konturzug. Im Extremfall kann das bis zu Unterbrechungen im selbigen führen. Dieses Problem ist ein bedeutendes bei der Segmentierung des LV in 3D-SPECT-Daten, da bedingt durch die funktionelle Bildgebung Lücken in der Herzwand und damit auch in ihrer eigentlich durchgängig zu segmentierenden Kontur auftreten können.

Bei der Verwendung gebräuchlicher *Kontursensoren*, die durch starke Konturinformation, also Gradientenbeträge, angezogen werden, tritt bei solchen Konturlücken der Effekt auf, daß die Sensoren zum Rand der Lücke streben und sich dort ballen bzw. daß sich Modellteile gänzlich von der Lücke entfernen, sich also zu den nächst stärkeren Konturinformationen bewegen. Durch dieses Verhalten kann sich die vorgegebene Modellform unter Umständen stark verzerren, da die Sensoren keine Möglichkeit haben, zwischen fehlender Konturinformation im zu segmentierenden Objekt und verschobener Konturinformation (also einem verschobenen, zu segmentierenden Objekt) zu unterscheiden. Diese Verschiebung von Sensoren an den Rand der Lücke wirkt sich durch die inneren Kräfte des Modells auf die weiteren Sensoren aus und verhindert damit im allgemeinen eine genaue Segmentierung. Zudem besteht so keine Möglichkeit, an Hand der Kontursensoreneingaben zuverlässig festzustellen, ob sie sich im Bereich einer Lücke befinden, um gegebenenfalls während der Modellanpassung dort im Modell z. B. die inneren Kräfte dynamisch zu verstärken. Denn an diesen Stellen kann sowieso nicht mit deutlichen Dateninformationen gerechnet werden und somit sollte vermehrt auf Modellannahmen vertraut werden.

Eine Möglichkeit, diesen Effekt deutlich abzumildern, besteht in der *Richtungsgewichtung* der Eingaben herkömmlicher Kontursensoren, für die die in Abschnitt 4.3 eingeführten Sensornormalen von zentraler Bedeutung sind. Diese Sensornormalen werden bei der Sensorrotation, wie sie in Abschnitt 4.2 beschrieben ist, analog den Ruherichtungen der Federn mitbewegt, spielen jedoch für die Berechnung dieser Drehung keine Rolle. Sie zeigen somit immer in die Richtung, die der Gradient einer durch sie zu findenen Konturstelle bei zutreffender Modellgestaltung haben sollte. Somit kann neben dem Betrag eines Gradienten die Abweichung seiner Richtung von der Normalen des entsprechenden Kontursensors als Gütekriterium der zu findenen Kontur benutzt werden, das sich auf die Anziehung dieses Sensors auswirkt.

Um diesen Effekt zu verdeutlichen, zeigt Abbildung 4.1 die gleiche Schicht

des Konturbildes K ohne und mit Richtungsgewichtung. Das erste der Konturbilder wurde also nach Gleichung 3.6 erzeugt und das zweite zusätzlich noch entsprechend den Abweichungen der Richtungen der Gradienten ∇D auf dem Datensatz D zu der in diesem Beispiel in X-Richtung verlaufenden Sensornormalenrichtung \vec{N}_i des Sensors i gemäß Gleichung 4.10 an jeder Position p des Datensatzes gewichtet.

$$K(p) = \left(1 - \frac{\angle(\nabla D(p), \vec{N}_i)}{\pi}\right) \cdot \|\nabla D(p)\| \quad (4.10)$$

Man sieht in Abbildung 4.1b im Vergleich zu Abbildung 4.1a, daß nur noch wenige Konturinformationen mit starker Anziehung für diesen Sensor übrig geblieben sind. Dadurch konvergieren die Kontursensoren zielstrebig und können auf Grund der durch sie erzeugten Sensorkräfte kaum längere Strecken entlang der Konturlinie verrutschen.

Durch die Abschwächung irrelevanter Konturinformationen durch die Richtungsgewichtung wird die Anpassung des gesamten Modells stabiler und zielstrebig, was gerade in höheren Dimensionen wegen der ebenfalls höheren Komplexität des Konturbildes dort bisher oft entgegengesetzt war. Da man die Richtungsgewichtung auf Grund ihrer Natur problemlos auf höhere Dimensionen erweitern kann, ist ihre Anwendung gerade in diesen Fällen zusätzlich von Vorteil.

4.5 Konturversatz

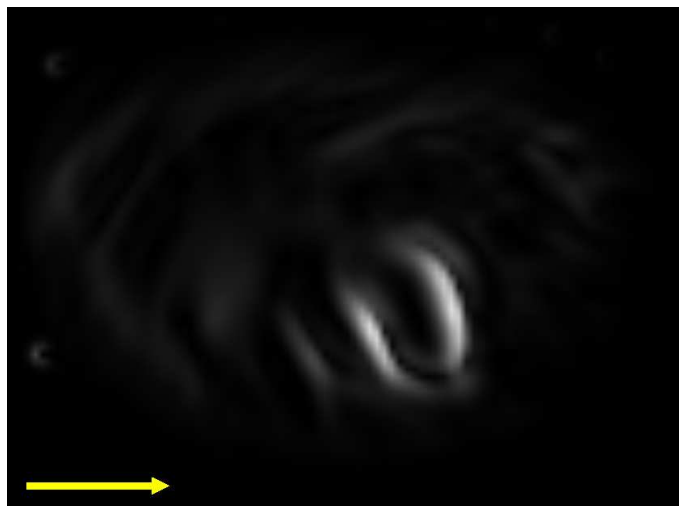
Ein weiteres Problem bei der Segmentierung speziell in SPECT-Daten ist die exakte Bestimmung des Konturverlaufs bei unscharfen Konturen, deren Querschnitt mehrere Voxel umfaßt. Der höchste Gradientenbetrag entlang dieses Querschnittes ist nicht notwendigerweise der ideale Verlauf der Kontur eines abgebildeten Organs, wenn man die Handsegmentierungen als Goldstandard zu Grunde legt, wie z. B. Abbildung 4.2 zeigt. Zusätzliches Modellwissen des Menschen kann also Einfluß auf den genauen Verlauf der Segmentierungskontur nehmen und sollte in den dynamischen Modell im Sinne eines *Konturversatzes* repräsentierbar sein.

Die in Abschnitt 4.3 eingeführten Sensornormalen \vec{N}_i eines Sensors i ermöglichen auf einfache Weise die Berücksichtigung solch zusätzlichen Wissens während der Anpassung, in dem die Meßpositionen der Kontursensoren entlang ihrer Normalen um den entsprechenden Konturversatz Δs verschoben werden, ohne jedoch den Sensor selbst zu verschieben. Gleichung 4.11 zeigt die Ermittlung der neuen Sensormeßposition $\vec{s}_{i_{\text{mess}}}$ des Sensors i auf Grundlage der alten Sensormeßposition \vec{s}_i .

$$\vec{s}_{i_{\text{mess}}} = \vec{s}_i + \Delta s \cdot \vec{N}_{i_{\text{norm}}} \quad \text{mit} \quad \vec{N}_{i_{\text{norm}}} = \frac{\vec{N}_i}{\|\vec{N}_i\|} \quad (4.11)$$



a



b

Abbildung 4.1: Über den Gradientenbetrag erzeugte Konturbilder eines Schichtbildes. Bild (b) wurde zusätzlich einer Richtungsgewichtung in X-Richtung (Pfeil) unterzogen. Die hellen Bereiche sind jeweils die unscharfen Konturen des LV.

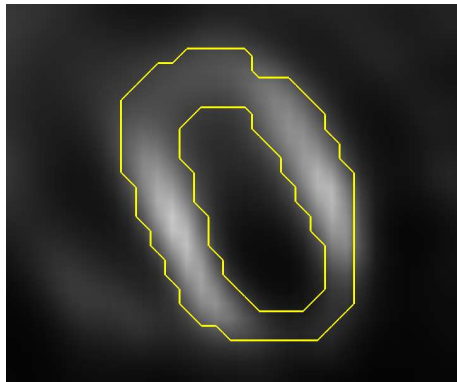


Abbildung 4.2: LV mit zugehöriger Handsegmentierung unseres klinischen Partners. Es ist deutlich zu sehen, daß etwas weiter als bis zur idealen Kontur segmentiert wurde.

4.6 Zusammenfassung

Die in diesem Kapitel eingeführten erweiterten dynamischen Modelle sind so konstruiert, daß sie auf Grund ihrer Mechanik in höheren Dimensionen stabiler und zielstrebigere sind als die in Abschnitt 3.1 beschriebenen einfachen dynamischen Modelle und somit eine deutlich dünnere Vernetzung bei mindestens gleicher Stabilität erlauben. Bereits im zweidimensionalen Fall können sich ihre positiven Effekte zeigen, da auch dort Richtungen von Federn und Konturnormalen eine Rolle spielen. Weiterhin ermöglichen sie erstmals eine kontrollierte Drehung des gesamten Modells, ohne es zu verzerren. Zusätzlich ist es möglich, Modellwissen über die Art des Konturverlaufs einzubringen.

Durch die Einführung der Torsionskräfte ist zudem bei schwacher Vernetzung erstmals die theoretische Grundlage für eine nahezu unabhängige und direkte Gewichtung von Modellgröße und -form gelegt.

Daher stellen diese erweiterten dynamischen Modelle eine für die Lösung der gegebenen Aufgabe wesentliche Erweiterung der herkömmlichen 3D-Modelle dar und sollen deshalb in diesem Rahmen auch Verwendung finden.

Kapitel 5

Modellgenerierung

Die Segmentierung mit den dynamischen 3D-Modellen aus Kapitel 3 setzt das Vorhandensein eines entsprechenden Modells voraus, dessen adäquate manuelle Erstellung auf Grund der erforderlichen Modellkomplexität in 3D zu aufwändig ist (vergleiche dazu Abschnitt 8.4).

Da bisher keine Verfahren zur automatischen Erstellung solcher Modelle existieren (vergleiche Abschnitt 2.3), wird im nun folgenden Kapitel ein solches Verfahren am Beispiel des LV entwickelt, mit dem dynamische 3D-Modelle aus (z. B. handsegmentierten) Beispielen beliebiger anatomischer Strukturen erzeugt werden können.

5.1 Teilstrukturen des Modells

Das zu erstellende Modell besteht aus verschiedenen Gruppen von vernetzten Sensoren gleichen Typs, die zusammenhängende Teilstrukturen der zu modellierenden Struktur darstellen. Diese Teilstrukturen können Begrenzungen der Struktur oder innere und äußere Bereiche ähnlicher Intensität sein.

Für den LV in SPECT-Daten werden 3 Teilstrukturen benötigt:

1. die intensitätsstarke Wand des LV (durchblutete Muskeln)
2. der intensitätsschwache Bereich im Inneren des LV
3. die Begrenzung der hellen Wand des LV

Das Modell wird aus einer vorgegebenen Segmentierung (z. B. händisch gewonnen) automatisch erzeugt. Da in dieser Segmentierung nur die Strukturen für die helle LV-Wand bzw. deren Rand erkennbar sind, muß der innere Bereich des segmentierten LV markiert werden. Das geschieht durch Subtraktion des LV von seiner konvexen Hülle. Die unter Umständen am äußeren Rand des LV dabei entstehenden Artefakte werden anschließend durch die Anwendung einer minimalen Volumenbedingung zusammenhängender und als LV-Inneres markierter

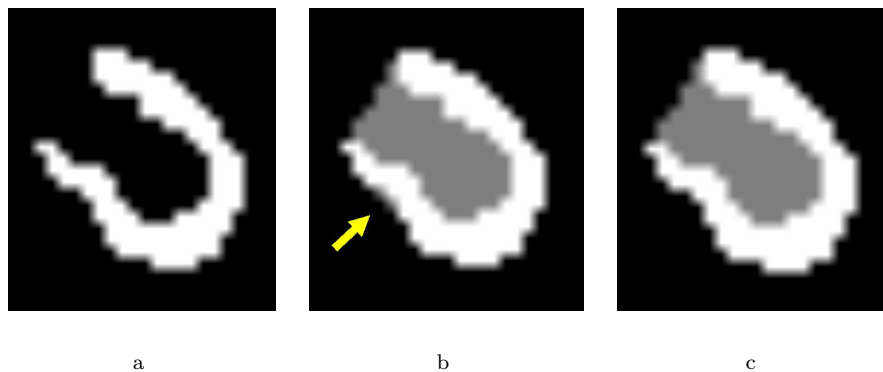


Abbildung 5.1: Prozeß des Markierens des inneren LV-Bereiches einer gegebenen Handsegmentierung. **(a)** ist die ursprüngliche Segmentierung. **(b)** zeigt die Differenz (grau) zur konvexen Hülle des LV. **(c)** zeigt das Ergebnis nach Entfernung der Artefakte (Pfeil)

Teilstrukturen entfernt. Abbildung 5.1 verdeutlicht den Ablauf der Markierung des Inneren einer LV-Handsegmentierung.

5.2 Phasen der Modellgenerierung

Die Sensorgruppen für die einzelnen Teilstrukturen werden separat erzeugt und anschließend untereinander mit Federn vernetzt. Dabei gilt auf Grund des stabilen Charakters der erweiterten dynamischen 3D-Modelle stets die Maßgabe einer (bei hier gegebener Stabilität) vorteilhaften schwachen Vernetzung (siehe Abschnitt 4.6 dazu).

5.2.1 Erzeugung der Intensitätssensoren

Zunächst werden im jeweiligen Segment (Teilstruktur 1 bzw. 2) die Intensitätssensoren mit gleichmäßiger Dichte räumlich innerhalb der entsprechenden Segmente verteilt. Dazu wird ein heuristisches Verfahren eingesetzt, das die späteren Intensitätssensoren mit erhöhter Dichte gleichmäßig (z. B. auf einem Iso-Gitter) verteilt und nach ihrer Generierung solange mit ihren räumlichen Nachbarn zusammenfaßt, bis sich die gewünschte Dichte ungefähr eingestellt hat.

Im Einzelnen wird dabei jeder Sensor i jeweils von seiner Position \vec{s}_i in den Schwerpunkt (Ortsvektor $s_{i_{\text{neu}}}$) aller Sensoren einer vorgegebenen Umgebung ε um ihn verschoben (Gleichung 5.1), wobei die Größe dieser Umgebung natürlich von der vorgegebenen, am Ende zu erreichenden Sensordichte abhängt. Die Sensoren werden in einem Durchgang jeweils alle zugleich repositioniert und danach mit gleich positionierten Sensoren verschmolzen. Die Durchgänge werden solange wiederholt, bis keine weitere Veränderung eintritt, die Dichte also nicht mehr

größer als die gewünschte ist, sondern bei entsprechend großer Startsensoranzahl und gleichmäßiger Startsensorverteilung knapp unter der angegebenen Dichteschranke liegen sollte.

$$\vec{s}_{i_{\text{neu}}} = \frac{1}{|J|} \sum_{j \in J} \vec{s}_j \quad \text{mit} \quad J = \{j \mid \|\vec{s}_j - \vec{s}_i\| \leq \varepsilon\} \quad (5.1)$$

Die so erzeugten Sensoren werden räumlich Delaunay-trianguliert, wobei die Kanten der Triangulation die späteren Federn zwischen den Intensitätssensoren sind. Auf diese Weise ist die Sensorgruppe in sich durchgängig und gewissermaßen topologisch ausgewogen verbunden, was der Maßgabe einer schwachen Vernetzung entgegenkommt. Anschließend wird die im allgemeinen noch zu hohe Zahl dieser Federn so reduziert, daß nur noch Sensoren miteinander verbunden sind, die sich in unmittelbarer räumlicher Nachbarschaft zueinander befinden, wobei dieser Vernetzungsbereich durch eine obere Abstandsschranke spezifiziert wird. Dem liegt die Annahme zugrunde, daß die Korrelation zwischen dem Verhalten zweier Sensoren mit ihrem Abstand abnimmt.

Die Delaunay-Triangulierung ist hierbei ein wichtiger Teil der Ausdünnungsprozedur, da es ohne sie, bei einer Ausdünnung der Federn nur auf Basis räumlich lokaler Beziehungen, häufig zu lokalen Ballungen von kurzen Federn kommt, die nicht erwünscht sind, da sie nicht im Sinne einer gleichmäßigen, dünnen Vernetzung sind.

5.2.2 Erzeugung der Kontursensoren

Die Kontursensoren werden ebenfalls mittels eines heuristischen Verfahrens möglichst gleichmäßig in der durch sie zu modellierenden Teilstruktur (der Kontur der LV-Wand) erzeugt. Das geschieht ausgehend von einer Iso-Oberfläche, die ein trianguliertes Polygonnetz aus Kontursensoren, Federn und den von ihnen eingeschlossenen Randflächen darstellt und aus der Teilstruktur der vorgegebenen Segmentierung generiert wird (z. B. mittels des Marching-Cubes-Algorithmus [LORENSEN und CLINE 1987] auf dem von dieser Teilstruktur umschlossenen Bereich).

Das Polygonnetz der Iso-Oberfläche enthält im allgemeinen jedoch zu viele Sensoren und Federn und muß möglichst formerhaltend vereinfacht werden. Eine der zur Zeit vielversprechendsten Methoden im Sinne eines hochqualitativen Ergebnisses bei gleichzeitig effizienter Berechnung für diesen Zweck (siehe [GARLAND 1999] für einen ausführlichen Vergleich möglicher Verfahren) ist die formerhaltende Polygonnetzvereinfachung mittels Quadric Error Metrics nach [GARLAND und HECKBERT 1997]. Sie soll deshalb in dieser Arbeit zur formerhaltenden Vereinfachung der Iso-Oberfläche eingesetzt werden.

Das Polygonnetz der Iso-Oberfläche wird dieser Vereinfachung wiederholt unterzogen, bis sich die gewünschte minimale Kontursensordichte einstellt bzw. zum ersten mal unterschritten wurde. Hierbei ist es von Bedeutung, kleine Reduktionsraten bei der Polygonvereinfachung zu wählen, damit die Sensordichte

langsam absinkt und nicht direkt stark unter die gewünschte minimale Sensordichte fällt.

Da diese Methode zu Visualisierungszwecken konzipiert wurde, können in ebenen Bereichen der Randfläche starke Inhomogenitäten (große Lücken) im Sensorgitter entstehen, da an diesen Stellen zusätzliche Polygone keine Verbesserung der Darstellung bewirken würden.

In einem dynamischen Modell stellen diese Inhomogenitäten jedoch ein Problem dar, weil auf diese Weise ein großer, sensorloser Konturbereich geschaffen wird, der dem Modell keine Anpassungsinformationen liefert. Daher müssen diese Inhomogenitäten durch Einfügen zusätzlicher Kontursensoren wieder aufgefüllt werden. Das passiert durch Einfügen neuer Kontursensoren in den Schwerpunkt zu großer Randflächen mit daran anschließender Retriangulierung dieser Bereiche. Diese Einfügeprozedur wird bis zur einer Kontursensordichte dieses Bereichs wiederholt, die für eine solch großflächige, ebene Modellkontur ausreichend ist und die im allgemeinen kleiner sein wird, als die zur adäquaten Beschreibung eines zerklüfteten Modellkonturbereichs nötige Kontursensordichte.

5.2.3 Vernetzung der Sensorgruppen

Die wie oben beschrieben erzeugten Sensorgruppen werden entsprechend den durch sie dargestellten Bereichen der Struktur räumlich angeordnet und lokal untereinander vernetzt. Dabei werden $1:n$ -Verbindungen benutzt, so daß für gegebene Anzahlen k und l alle Intensitätssensoren der LV-Wand mit den jeweils k nächsten Intensitätssensoren des LV-Inneren und alle LV-Rand-Sensoren mit den l nächsten Intensitätssensoren der LV-Wand vernetzt werden. Auf diese einfache Weise wird bei der Form des LV eine sehr lokale Vernetzung erreicht, die zudem eine eindeutige und einfache Berechnung der in Abschnitt 6.1.3 beschriebenen Modellkonturkompression ermöglicht.

Für andere zu modellierende Strukturen sind auch andere Arten der lokalen Vernetzung verschiedener Sensorgruppen denkbar, so daß z. B. nur räumlich benachbarte, äußere Sensoren der verschiedenen Gruppen miteinander verknüpft werden.

Nach vollendeter Vernetzung sind alle nötigen Federn in das Modell eingefügt und ihre Ruhelängen und relativen Richtungen können bestimmt werden. Sie ergeben sich aus ihrer Länge und Richtung zu diesem Zeitpunkt.

5.3 Zusammenfassung

In diesem Kapitel wurde ein heuristisches, mehrstufiges Verfahren entwickelt, mit dem erstmals dynamische 3D-Modelle aus (z. B. handsegmentierten) Beispielen beliebiger anatomischer Strukturen erzeugt werden können. Mittels verschiedener Generierungsparameter (Sensor-Dichte, Vernetzungsgrad, etc.) kann direkt Einfluss auf die Modellerzeugung genommen werden, so daß verschiedene Arten von Modellen generiert werden können, die auch in höheren Dimensionen

die Form und Größe der Ausgangsbeispielstruktur genau, topologisch ausgewogen und im Falle erweiterter dynamischer Modelle ebenfalls stabil repräsentieren, aber dabei mit einer geringen Anzahl von Federn auskommen.

Kapitel 6

Automatische Segmentierung

Dieses Kapitel entwickelt einen für die gegebene Themenstellung geeigneten automatischen Segmentierungsprozeß, der aus mehreren Phasen aufgebaut sein wird. Diese Phasen der automatischen Segmentierung repräsentieren einen Segmentierungsplan, wie in Abschnitt 2.2.3 erwähnt und stellen somit Modellwissen über den Prozeß der Segmentierung dar.

6.1 Phasen der Segmentierung

Beim menschlichen Anwender kommt es vor, daß er bei einer Segmentierung in mehrere Phasen vorgeht. Zuerst wird das zu segmentierende Objekt im Datensatz eingegrenzt bzw. gefunden. Dann wird das Objekt grob herausgearbeitet, um letztendlich die Konturen exakt zu segmentieren.

Die automatische Segmentierung soll sich in Analogie dazu in dieser Arbeit in 3 Phasen gliedern:

Phase 1: Startplatzierung des Modells

Phase 2: Formanpassung des Modells

Phase 3: Konturanpassung des Modells

In Phase 1 wird die initiale Position des dynamischen Modells ermittelt. Dieser Schritt ist entscheidend für das Gelingen der nachfolgenden Phasen, da dynamische Modelle lediglich lokale Anpassungen vornehmen können und daher die Startposition des Modells ungefähr der des zu segmentierenden Objektes im Datensatz entsprechen sollte.

In der zweiten Phase wird dann ausgehend von der in Phase 1 ermittelten initialen Modellposition mittels des dynamischen Modells die Formanpassung

durchgeführt. Dabei werden im Einzelnen Form, Größe und Ausrichtung angepaßt.

In der abschließenden dritten Phase findet ausgehend von dem bereits angepaßten Formmodell die eigentliche Segmentierung der Objektkontur statt. Die Anpassung aus Phase 2 bildet dabei den nötigen stabilen Rahmen für die Konturanpassung.

6.1.1 Phase 1: Startplatzierung

In der Literatur zur Segmentierung mit dynamischen Formmodellen wird diese erste Phase einer automatischen Modellpositionierung im allgemeinen ausgelassen (Pohle erwähnt in ihrer ausführlichen Übersicht über die Anwendung dynamischer Modelle in [POHLE 2004] nichts dergleichen.) und durch Benutzerinteraktion ersetzt bzw. als bereits auf irgendeine Weise erfolgt angesehen. Dabei bieten gerade Formmodelle hervorragende Möglichkeiten diese Phase zu automatisieren bzw. die alternative Benutzerinteraktion gehörig zu unterstützen, was gerade in höherdimensionalen Datensätzen durch die dort naturgemäß schlechteren Interaktionsmöglichkeiten sinnvoll wäre.

Dynamische Formmodelle beinhalten die komplette Kontur eines idealen, durchschnittlichen oder beispielhaften Vertreters der Klasse des zu segmentierenden Objektes. Mittels dieses Wissens läßt sich dieser Vertreter als künstlicher Datensatz rekonstruieren, wie es Abschnitt 6.3 beispielhaft für die erweiterten dynamischen Modelle dieser Arbeit beschreibt. Faßt man diesen Datensatz als Muster auf, so kann durch eine tolerante Mustererkennung auf dem zu segmentierenden Datensatz ein geeigneter Startplatz für das Modell bzw. eine Reihe möglicher und verschiedener (also einander nicht zu nahe liegender) Kandidaten für selbigen gefunden werden. Letztere können dann zur Unterstützung der Benutzerinteraktion durch manuelle Auswahl verwendet werden. Das alles funktioniert natürlich nicht, wenn das initiale Modell das Objekt im zu segmentierenden Datensatz nicht nur schlecht annähert. Von einer gewissen Ähnlichkeit zwischen Muster und zu segmentierendem Objekt muß hierbei aber ausgegangen werden, da sonst das angebliche Modellwissen für die Segmentierung eher hinderlich als nützlich wäre.

Laut Aussage unserer klinischen Partner kann bei den durch sie erzeugten Datensätzen, die die Grundlage dieser Arbeit bilden, davon ausgegangen werden, daß Ausrichtung und Größe des zu segmentierenden LV auf Grund der gleichbleibenden Modalitäten des Aufnahmeverfahrens nur gering variieren. Daher können Verfahren zur Mustererkennung benutzt werden, die eine feste Größe und Ausrichtung des Musters voraussetzen.

Eine einfache Form solch einer Mustererkennung ist mittels *Faltung* möglich. Sie kann schnell im Fourierraum berechnet werden und ist im allgemeinen für die vorliegende Aufgabenstellung hinreichend (siehe Abschnitt 8.5.1). In seltenen Fällen versagt sie jedoch auf Grund von sehr hellen Strahlungsartefakten, da bei dieser Form der Mustererkennung die Intensitäten im Bild eine Rolle spielen. Alternativ sollte dann ein Verfahren benutzt werden, das sich auf die reine Forminformation in den Daten konzentriert. Die Bestimmung der Musterüber-

einstimmung mittels *Mustervektorähnlichkeiten* tut gerade das, ist jedoch bedeutend langsamer als das erstere. Beide sollen nun in den folgenden beiden Unterabschnitten (6.1.1.1 und 6.1.1.2) im Einzelnen beschrieben werden.

6.1.1.1 Mustererkennung durch Faltung

Eine Faltung eines reellwertigen Datensatzes f mit dem in jeder Dimension entlang ihr am Ursprung gespiegelten reellwertigen Muster g als Kern wird häufig für Mustererkennungsaufgaben eingesetzt ([GONZALEZ und WOODS 1992]) und wird im Englischen in diesem Zusammenhang auch als *Korrelation* (Operator: \circ) bezeichnet. Da dieser Operator, wie Gleichung 6.1 verdeutlicht, eine diskrete Faltung der Größe M ist, kann sie effizient über die *Schnelle Fourier Transformation* berechnet werden.

$$f(x) \circ g(x) = \frac{1}{M} \sum_{m=0}^{M-1} f(m) \cdot g(x+m) \quad (6.1)$$

Diese Operation ist proportional zum Skalarprodukt des vom Muster g überdeckten Bereichs des Datensatzes f mit g selbst, so daß die Werte der Intensitäten im Datensatz und Muster eine wichtige Rolle bei der Berechnung des Faltungswertes spielen, der als Maß für die Übereinstimmung des Musters mit dem Datensatz an einer spezifischen Stelle gewertet werden kann. Diese spezifische Stelle ist hier so gewählt, daß sie sich in der Mitte des quasi auf dem Datensatz liegenden Musters befindet. Nach dem das Muster über den ganzen Datensatz geschoben wurde und somit an allen seinen Stellen eine Ähnlichkeit zu dem Muster berechnet wurde, entsteht eine Karte des Datensatzes mit zum Muster ähnlichen Bereichen.

Gleichung 6.2 zeigt die Berechnung des auf einer entsprechend verschobenen Faltung basierenden Ähnlichkeitswertes e für den dreidimensionalen Datensatz D und das wie beschrieben gespiegelte Muster P mit den Ausmaßen P_x , P_y und P_z . Der konstante Faktor $\frac{1}{M}$ aus Gleichung 6.1 wurde hier nicht mehr mit einbezogen, da er für die vergleichende Betrachtung der Musterähnlichkeiten innerhalb eines Datensatzes keine Rolle spielt.

$$e(x, y, z) = \sum_{i=1}^{P_x} \sum_{j=1}^{P_y} \sum_{k=1}^{P_z} D\left(x - \left\lfloor \frac{P_x}{2} \right\rfloor + i, y - \left\lfloor \frac{P_y}{2} \right\rfloor + j, z - \left\lfloor \frac{P_z}{2} \right\rfloor + k\right) \cdot P(i, j, k) \quad (6.2)$$

Um vergleichbare Ergebnisse bei verschiedenen, verwendeten Mustern zu erhalten, sollte das Muster das Objekt durch einen normierten Wert darstellen, z. B. 1. Um weiterhin alle Informationen des Modells bei der Ähnlichkeitsberechnung zu nutzen, sollte das aus ihm erzeugte Muster (siehe Abschnitt 6.3 dazu) sowohl die garantiert intensitätsstarken, als auch die garantiert intensitätsschwachen Bereiche des Modells repräsentieren. Das wird bei Beibehaltung einer in dieser Arbeit uniformen, normierten Darstellung durch die Kennzeichnung der intensitätsschwachen Bereiche durch den Wert -1 und der intensitätsstarken

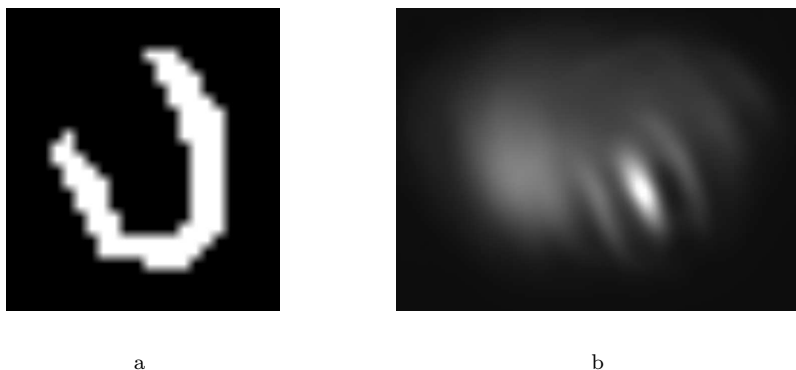


Abbildung 6.1: Aus dem Modell generiertes Muster (a) und Ergebnis der faltungsbasierten Erkennung dieses Musters auf einem Datensatz (b). Helle Bereiche stehen dort für hohe Ähnlichkeitswerte des Musters.

Bereiche durch den Wert 1 erreicht. So tragen starke Intensitäten nur dort zu einem hohen Ähnlichkeitswert bei bzw. vermindern einen schwachen nur wenig, wo sie durch das Muster erwartet wurden. Andererseits wirken sich den Erwartungen entgegengesetzte Intensitätswerte nachteilig auf einen hohen Ähnlichkeitswert aus, da sie ihn im Falle von starken Intensitäten im Datensatz deutlich reduzieren bzw. bei schwachen nur wenig stützen. Bereiche ohne zuverlässige Modellinformation werden mit 0 gefüllt, so daß sie keinen Einfluß auf den berechneten Ähnlichkeitswert haben. Abbildung 6.1 zeigt solch ein Muster und das Ergebnis seiner Anwendung auf einen Datensatz.

Es bleibt jedoch das Problem durch Artefakte verursachter Bereiche von im Vergleich zum Rest des Datensatzes deutlich überhöhter Intensität. Sie verstärken den berechneten Ähnlichkeitswert, der im Prinzip einem Skalarprodukt entspricht, bei Überlagerung mit intensitätsstarken Bereichen des Musters unverhältnismäßig stark und führen so zu einer deutlich überhöht berechneten Ähnlichkeit an diesen Stellen. Auf Grund dieses Effektes können Regionen des Datensatzes bezüglich der Ähnlichkeit zum Muster überbewertet werden, so daß die Mustererkennung falsche Ergebnisse liefert. Abbildung 6.2a zeigt solch einen Datensatz mit den entsprechenden Resultaten.

6.1.1.2 Mustererkennung durch Vektorähnlichkeiten

Die möglichen Probleme bei der Verwendung der faltungsbasierten Ähnlichkeitsberechnung aus dem vorhergehenden Unterabschnitt beruhen auf deren direkter Abhängigkeit von der Höhe der einzelnen Intensitätswerte. Ein im Ergebnis ähnliches Verfahren, das den Fokus mehr auf die Form des Musters als auf die einzelnen Intensitäten legt, benutzt Mustervektorähnlichkeiten.

Es orientiert sich am *Kosinus-Klassifikator* ([KRATZER 1990]), einem einfachen neuronalen Netz zur Mustererkennung, bei dem die einzelnen Intensitäten

der Voxel nicht direkt, sondern nur in Relation zueinander für die Bewertung der Ähnlichkeit eine Rolle spielen. Dabei werden die Muster und gleich große Stellen im Datensatz durch normierte Vektoren repräsentiert, deren Ähnlichkeit sich durch die Ähnlichkeit der sie repräsentierenden Vektoren errechnet. Als Ähnlichkeitsmaß dieser Vektoren findet dabei der Kosinus des durch sie eingeschlossenen Winkels Verwendung, wie er bereits bei [KOHONEN 1987] für diesen Zweck motiviert wurde.

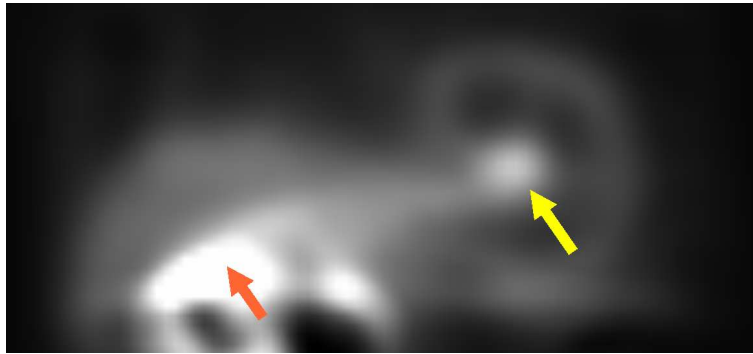
Beim hier vorgestellten Verfahren der Berechnung der Musterähnlichkeiten über Mustervektoren wird das zu findende Muster wiederum sukzessive über den ganzen Datensatz geschoben, wobei an jeder Stelle p (wie auch in Abschnitt 6.1.1.1 dabei wieder in der Mitte des Musters) die Ähnlichkeit zwischen Muster und Datensatz ermittelt wird. Zur Bestimmung dieser Ähnlichkeit werden die Werte des Musters sequentiell in vorbestimmter Reihenfolge als Komponenten des Vektors $\vec{P}(p)$ und die Werte des darunterliegenden Datensatzes als Komponenten des Vektors $\vec{D}(p)$ aufgefaßt. Der Kosinus des Winkels zwischen beiden Vektoren dient nun als Ähnlichkeitsmaß $e(p)$ und berechnet sich nach Gleichung 6.3.

$$e(p) = \cos \angle(\vec{D}(p), \vec{P}(p)) = \frac{\langle \vec{D}(p), \vec{P}(p) \rangle}{\|\vec{D}(p)\| \cdot \|\vec{P}(p)\|} = \langle \vec{D}(p)_{\text{norm}}, \vec{P}(p)_{\text{norm}} \rangle$$

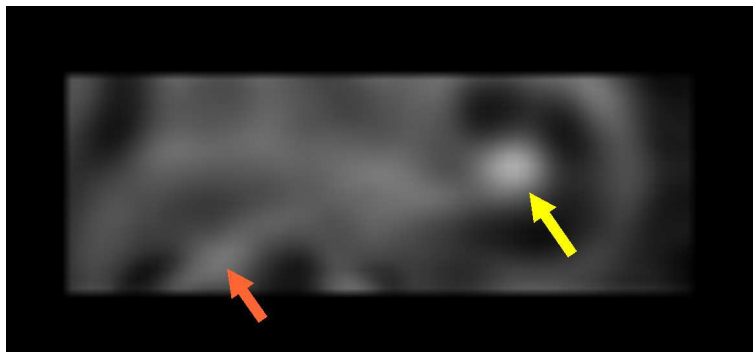
mit $\|\vec{D}(p)_{\text{norm}}\| = \|\vec{P}(p)_{\text{norm}}\| = 1$ (6.3)

Durch diese Art der Bestimmung der Ähnlichkeit wirken sich Intensitäten nur noch auf die Richtungen der Vektoren aus. Zwar handelt es sich auch hier um eine Skalarproduktberechnung wie bei der faltungsbasierten Mustererkennung, jedoch wird diese hier nur auf normierten Vektoren ausgeführt, so daß ihre absoluten Werte nicht mehr das Niveau der Ähnlichkeit beeinflussen. Dieses Verhalten zeigt sich z. B. deutlich auf dem Datensatz aus Abbildung 6.2a, auf dem das faltungsbasierte Mustererkennungsverfahren aus dem vorigen Abschnitt versagte, das hier vorgestellte Verfahren jedoch nicht durch das Artefakt gestört wird (Abbildung 6.2b).

Leider ist die Bestimmung dieser Art der Ähnlichkeit nicht so effizient durchführbar wie die Ähnlichkeitsbestimmung mittels einer Faltung, wie in Abschnitt 6.1.1.1 beschrieben, so daß das Verfahren aus Abschnitt 6.1.1.1 immer noch seine Berechtigung hat. Gerade bei extrem starken Strahlungsartefakten sollte jedoch das Verfahren aus diesem Abschnitt zum Einsatz kommen. In diesen Fällen ist auch jeweils die in Abschnitt 3.2 aufgeführte abgewandelte Normalisierung des Datensatzes mit vorherigem Abschwächung der zu starken Intensitäten von Interesse, um der späteren Modellanpassung günstige, weil immer ähnliche, Bedingungen zu schaffen. Sie soll daher in dieser Arbeit bei Verwendung dieser Variante der Phase 1 Verwendung finden.



a



b

Abbildung 6.2: **(a)** zeigt ein Schnittbild in Y-Richtung des Ergebnisses der faltungsbasierten Mustererkennung eines LV-Musters. **(b)** zeigt dieselbe Ansicht des Ergebnisses der Mustererkennung mittels Vektorähnlichkeiten desselben Musters. Helle Bereiche stehen jeweils für hohe Ähnlichkeitswerte des Musters. Deutlich ist zu sehen, daß in **(a)** ein Artefakt (roter Pfeil) bedeutend heller ist als die Mitte des LV (gelber Pfeil), es in **(b)** jedoch keine Rolle spielt. Dort ist die LV-Mitte am hellsten ist.

6.1.2 Phase 2: Formanpassung

Nach der initialen Positionierung des Modells wird in der zweiten Phase der automatischen Segmentierung das Modell an die Form des zu segmentierenden Objektes angepaßt, wobei nur die Intensitätsverteilung, nicht jedoch die genauen Konturen, eine Rolle spielen. Da die Intensitätsinformationen sich im Vergleich zu den Konturinformationen durchgängig und gleichmäßig über einen Bereich erstrecken, sind sie besser geeignet, das Modell daran auszurichten, zu verformen und leicht zu skalieren, so daß es mit dem zu segmentierenden Objekt grob zur Deckung kommt.

Auf diese Weise kann sich das Modell schnell, flexibel und robust ausrichten, ohne daß dieser Prozeß durch zuviel Detailinformationen durch die Beachtung der Konturen behaftet wird, was im allgemeinen einer zielstrebigem und auf das Wesentliche konzentrierten Konvergenz entgegen wirkt. Ebenfalls fallen Lücken im LV, wie sie durch einen Infarkt verursacht werden, nicht so stark ins Gewicht, da durch die Intensitätsinformation genug, die LV-Wand umgebende Information existiert, um weiterhin robuste und korrekte, grobe Anpassungen zu gewährleisten.

In dieser Phase sind also die Kontursensoren des Modells nicht aktiv oder nicht online (je nach Einstellung, siehe Abschnitt 6.1.3 dazu). Vielmehr sind es die *Intensitätssensoren*, die das Modell grob dem zu segmentierenden Objekt anpassen. Intensitätssensoren sind Sensoren, die durch starke bzw. schwache Intensität (je nach Einstellung des Sensors) angezogen werden (vergleiche dazu Abschnitt 3.2). Sie sind im allgemeinen mit Massen des Modells assoziiert, die innere oder äußere Knoten des Feder-Masse-Gittermodells sind, also nicht auf dem in Abschnitt 4.3 beschriebenen Konturgitter des Modells liegen. Neben einer gewissen Stützfunktion der Modellkontur durch innere Kräfte haben diese inneren und äußeren Gitterbereiche die Aufgabe, Sensorpositionen für Intensitätssensoren im Umfeld der Modellkontur anzubieten, da oftmals Wissen über die erwarteten Intensitäten in diesen Bereichen existiert. So ist z. B. für die Aufgabe dieser Arbeit bekannt, daß das Innere des LV sehr intensitätsschwach ist, die Wand des LV dafür sehr intensitätsstark (siehe z. B. Abbildung 1.2a).

Der LV entspricht, grob gesehen, einem konvexen, geschlossenen Körper, dessen Inneres intensitätsschwach und dessen Rand intensitätsstark ist. Phase 1 liefert im Idealfall den Mittelpunkt des LV (vergleiche Abschnitt 6.1.1.1 und 6.1.1.2). Wenn jedoch dieser Idealfall nicht eintritt und das Muster mit einem deutlichen Versatz im Datensatz detektiert wird (z. B. auf Grund von abweichender Ausrichtung oder Skalierung), gibt es doch wahrscheinlich noch eine gewisse Überschneidung, da sonst keine Detektion stattgefunden hätte. Das heißt, daß der Mittelpunkt des Musters immer noch im Inneren des LV des Datensatzes liegt, jedoch nicht mehr zentral, wie im Idealfall.

Um das Modell nun nicht unnötig mit einem größeren Versatz zum Objekt im Datensatz zu starten, so daß es Schwierigkeiten bei seiner korrekten Positionierung im Lauf der Modellanpassung geben könnte, kann es auch komprimiert an dem in Phase 1 ermittelten Startpunkt gestartet werden. So kann es mit sehr hoher Wahrscheinlichkeit innerhalb des zu segmentierenden LV seine anfänglich

expansive Bewegung beginnen. Da die Größe des Modell durch die Ruhelänge der Federn gegeben ist, wird es während der Bewegung seine originale Größe erreichen und kann sich dabei geleitet durch seine Intensitätssensoren direkt an die richtige Position drücken. Auf Grund des konvexen, im Inneren komplett intensitätsschwachen LV, der fast komplett von einem intensitätsstarken Bereich begrenzt wird, der auch die Modellexpansion begrenzen könnte, würde das Modell so robust, auch bei versetzter initialer Positionierung, die richtige Position finden und mit dem LV des Datensatzes grob zur Deckung kommen, wie es durch diese Phase erreicht werden soll.

6.1.3 Phase 3: Konturanpassung

Nach in Phase 2 erfolgter grober Modellanpassung über die Intensitätswerte soll nun in Phase 3 die endgültige und exakte Anpassung des Modells und damit auch seiner Kontur an den LV im Datensatz erfolgen.

Wie in der vorhergehenden Phase bewegt sich das Modell auch in Phase 3 wieder, bis es sich angepaßt hat. Diesmal werden allerdings die Kontursensoren benutzt, damit eine exakte Anpassung an das zu segmentierende Objekt möglich wird. Die Anpassung kann schnell, zielstrebig und robust erfolgen, da das Modell bereits grob angepaßt wurde und keine wirklich großen Änderungen der Konturlinie mehr erforderlich sein sollten.

Da das Modell bereits grob angepaßt vorliegt, die Kontursensoren also aus stabiler Position heraus starten, sind auch Lücken in der LV-Wand kein Problem mehr. Denn die Kontursensoren im Bereich der Lücken haben durch die inneren Sensorenkräfte des Modells nur einen geringen Spielraum und finden bei entsprechend großen Lücken keine Kontur, so daß sie an ihrer hauptsächlich durch die inneren Kräfte bestimmten Position festhalten und nicht zu einer entfernten Kontur abdriften und das Modell über Gebühr verzerren. Der Einsatz der richtungsgewichteten Kontursensoren aus Abschnitt 4.4 unterstützt dieses robuste Verhalten zusätzlich.

Damit die Kontursensoren, gerade wenn sie sich in Lücken der LV-Wand befinden, das gesamte Modell nicht aus seiner Lage drücken, braucht es ein Gegengewicht. Das sollte sinnvollerweise durch die Intensitätssensoren geschaffen werden, die das Modell in der vorigen Phase auch in diese Lage bewegt haben. Die Intensitätssensoren sollten also auch in dieser Phase benutzt werden, obwohl sie sich bereits vollständig angepaßt haben sollten. Für erhöhte Lagestabilität können alternativ die Intensitätssensoren auch als fest markiert werden, so daß sie ihre in Phase 2 gefundene Position nicht mehr verlassen können.

Da in dieser Phase also verschiedene Arten von Sensoren (Intensitäts- und Kontursensoren) gleichzeitig im Einsatz sind, muß dafür gesorgt werden, daß die durch sie verursachten Effekte ausgewogen sind, denn die Intensitätssensoren sorgen für die grobe, eher globale Anpassung des Modells (Ausrichtung, Verformung), während die Kontursensoren die Modellkonturen lokal zur Deckung mit denen des Objektes im Datensatz bringen. Um auf diese Effekte Einfluß zu haben, müssen die durch beide Sensorarten erzeugten externen Kräfte unabhängig gewichtet werden können.

Wurden in Phase 2 die Kontursensoren lediglich offline geschaltet, so bewegten sie sich dort durch die inneren Kräfte geführt bereits mit und können in dieser Phase gleich von Positionen aus starten, an denen das Modell bereits die Konturen erwartet.

Wenn in Phase 2 die Kontursensoren jedoch inaktiv waren, wurde dort nur der Teil des Modells mit den Intensitätssensoren benutzt. Die Kontursensoren wurden damit nicht beachtet und somit auch seit Phase 1 nicht mitbewegt. Daraus ergibt sich die Frage nach ihrer Startposition in dieser Phase. Die einfachste Möglichkeit ist, daß sie ihre Position aus Phase 1 behalten, die nicht extrem von ihrer Zielposition entfernt sein sollte, da ab Phase 2 durch das dynamische Modell nur noch mehr oder weniger große lokale Anpassungen vorgenommen werden können. Alternativ kann auch hier wieder eine Kompression vorgenommen werden, die ähnlich motiviert ist, wie die in Abschnitt 6.1.2. Denn auch hier kann vorausgesetzt werden, daß etwas innerhalb eines Objektes ist, nämlich die Intensitätssensoren für starke Intensitäten innerhalb der (anatomischen) LV-Wand, deren Kontur durch die Kontursensoren gefunden werden soll. Werden nun die Kontursensoren nahe der mit ihnen verbundenen Intensitätssensoren der LV-Wand positioniert, so expandieren sie in Richtung der LV-Wandkontur und werden bei ihrer Expansion durch keine andere Kontur behindert, so daß sie diese Kontur auf robuste und direkte Weise finden sollten, da ihnen durch die Ruhelänge der Federn zudem der ungefähre Abstand dieser Kontur von ihrer Startposition gegeben ist.

6.2 Modellgleichgewicht

Nach der Betrachtung der einzelnen Phasen der Segmentierung in den vorhergehenden Abschnitten wurde eine Frage noch nicht gestellt, nämlich die nach dem Zeitpunkt der Phasenübergänge. Nur Phase 1 hat ein definiertes Ende, nämlich das Ende des Mustererkennungsprozesses. Sowohl Phase 2 als auch Phase 3 hingegen beziehen sich auf die erfolgte Anpassung des entsprechenden dynamischen Modells. Sie ist im Idealfall erreicht, wenn sich das Modell nicht mehr bewegt, also seine verschiedenen Kraftkomponenten ausgeglichen sind. Doch wie kann im konkreten Anwendungsfall das Vorliegen des *Modellgleichgewichts* adäquat bestimmt werden?

Die Literatur schenkt diesem Thema nicht allzu viel Aufmerksamkeit, da es hinter der eigentlichen Anpassung oft zurücktritt und erst bei vollautomatischen Segmentierungsprozessen an Bedeutung gewinnt. Einige Beiträge gibt es aber. So wird in [ENGEL 2004] hierfür z. B. die Anpassung für beendet erklärt, wenn entweder die externen oder die internen Kräfte minimal werden, wobei Details zur Berechnung leider nicht ausgeführt werden. Passiert das innerhalb einer vorgegebenen Iterationsanzahl in der Modellbewegung nicht, so wird diese danach auch beendet. Diese Kriterien scheinen ungünstig gewählt zu sein, wie einzelne Ergebnisse in [ENGEL 2004] zeigen, die eine Fortbewegung des Modells aus guter Anpassung heraus mit anschließendem Modellgleichgewicht in weniger günstiger Lage schildern. Sie sollen daher hier keine Anwendung finden. Sonst

werden in der Literatur Abbruchkriterien in Form reiner Simulationszeit- bzw. Simulationsschrittbegrenzungen verwendet, die nicht direkt mit dem Erreichen des Gleichgewichtszustandes korreliert sind.

In dieser Arbeit soll nun vielmehr der intuitive Ansatz formuliert und verwendet werden: Ein Modell ist im Gleichgewicht, wenn es sich nicht mehr signifikant bewegt. Algorithmisch gesprochen heißt das, daß das Modell sich dann einem Gleichgewichtszustand genähert hat, wenn alle seine Massen bzw. Sensoren innerhalb einer festen Zeit von n Zeitschritten jeweils eine gewisse, feste (und natürlich klein zu wählende) Umgebung ε nicht verlassen haben, wenn also Gleichung 6.4 gilt, wobei \vec{s}_{i_t} der Ortsvektor des Sensors i zum Zeitschritt t (beginnend bei 1) ist.

$$\forall i \forall 0 \leq j < n - 1 : \|\vec{s}_{i_{t-j}} - \vec{s}_{i_{t-(n-1)}}\| \leq \varepsilon \quad \text{für } t \geq n \quad (6.4)$$

Durch dieses Kriterium wird bei adäquater Parametrisierung sichergestellt, daß das Modell wirklich zum Stillstand gekommen ist, da diese Bedingung direkt von der Bewegung des Modells abhängt und nicht nur indirekt bzw. überhaupt nicht, wie bei anderen Ansätzen.

6.3 Voxelbasierte Modellrepräsentation

Sowohl für die in Abschnitt 6.1.1 erwähnte Erzeugung eines Musters aus einem noch nicht angepaßten Modell, als auch für die letztendliche Generierung einer voxelbasierten Segmentierung aus einem fertig angepaßten Modell, wie sie von unseren klinischen Partnern gewünscht wird, ist es nötig, die Voxel zu bestimmen, die zu dem durch die Modellkontur umschlossenen Bereich gehören.

Neben den vollständig in diesem Bereich enthaltenen Voxeln, die natürlich dazugehören, stellt sich die Frage, wie mit den partiell zu diesem Bereich gehörenden Randvoxeln zu verfahren ist. Sie komplett dazu- bzw. nicht dazuzurechnen, würde die Ergebnissenauigkeit systematisch in eine Richtung verfälschen. Idealerweise sollten diese Randvoxel jeweils zu der Seite gezählt werden, mit der sie den größten Teil ihres Volumens teilen.

Die dazu nötigen Berechnungen (pro Voxel unter Umständen mehrere Schritte des Voxels mit durch die Modellkonturflächen induzierten Halbräumen und anschließender Durchschnittsbildung) sind jedoch sehr rechenaufwendig und führen zu übermäßig langen Berechnungszeiten, die die Majorität des gesamten Segmentierungsprozesses auf den gegebenen Daten ausmachen würden. Vereinfacht man jedoch das Kriterium darauf, den Randvoxel zu der Seite zu zählen, zu der sein Mittelpunkt gehört, vereinfacht und beschleunigt sich damit auch die Berechnung erheblich, so daß sie den Zeitaufwand der gesamten automatischen Segmentierung nicht länger deutlich belastet. Zudem entspricht dieses vereinfachende Kriterium dem idealen Majoritäts-Volumen-Kriterium in dem Fall, daß ein Randvoxel von genau einer Konturfläche geschnitten wird, was im allgemeinen dem Normalfall entspricht.

Abbildung 6.3 zeigt, wie die aus einem Modell auf diese Weise erzeugte voxelbasierte Segmentierung aussieht.

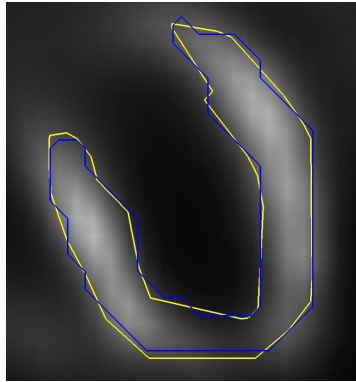


Abbildung 6.3: die Kontur eines Modells (gelb) und seine daraus erzeugte voxelbasierte Repräsentation (blau)

6.4 Interaktion

Der vorgestellte Segmentierungsprozeß erfordert keinerlei Interaktion, bietet dem Benutzer jedoch unzählige Möglichkeiten, gezielt, verständlich und ohne viel Aufwand in ihn einzugreifen. So bieten gerade die Übergänge von einer Phase zur nächsten adäquate und einfache Punkte zur Nachkorrektur, da klar ist, wie die Positionierung bzw. Anpassung nach einer Phase sein sollte. Zudem wird so die Möglichkeit geboten, nach Korrekturen mit den anschließenden Phasen automatisch fortzufahren, so daß die Interaktion den automatischen Segmentierungsprozeß nicht beendet, sondern Korrekturen so früh wie möglich zuläßt. So werden unter Umständen umfangreiche, weil exakte Korrekturen am Ende zur Fertigstellung der Segmentierung vermieden, weil sie durch eine kleine Korrektur in einer frühen Phase des Segmentierungsprozesses, der dann automatisch fortgesetzt wurde, überflüssig wurden.

Die nötigen Arten der Interaktion zur Korrektur sind nach Phase 1 die manuelle Bestimmung eines Punktes im Inneren des zu segmentierenden LV bzw. nach Phase 2 und 3 Positionskorrekturen einzelner, in dieser Phase verwendeter Sensoren (und somit der mit Ihnen assoziierten Massen). Das alles kann bei geeigneter Darstellung in kurzer Zeit bequem mit der Maus ausgeführt werden (siehe Beschreibung des Prototyps in Abschnitt 7.3).

Zusätzlich kann es sinnvoll sein, die Wichtungen der einzelnen Kraftkomponenten (Feder-, Torsions-, Intensitäts- und Kontursensorkraft) des dynamischen Modells zu ändern, um Schwerpunkte auf bestimmte Aspekte der Anpassung (Skalierung, Form, Position, Kontur) zu legen. Auch wenn eine Einstellung der Wichtungen häufig prinzipiellen Charakter für eine Klasse von Datensätzen und zugehörigen Modellen besitzt, kann es doch von Interesse sein, diese im Einzelfall anzupassen.

6.5 Zusammenfassung

In diesem Kapitel wurde ein automatischer Segmentierungsprozeß für den LV in SPECT-Daten entwickelt, der aus drei Phasen besteht. Seine aufeinander aufbauenden Phasen, die das Modell Stück für Stück genauer an das zu segmentierende Objekt des Datensatzes anpassen, sorgen dafür, daß lokale und detaillierte Informationen erst zur Anpassung verwendet werden, wenn das Modell bereits so gut im Groben angepaßt wurde, daß diese lokale Information richtig eingeordnet werden kann und nur noch lokale Auswirkungen auf das Modell hat. Das macht die dynamische Modellanpassung insgesamt stabiler.

Da somit Informationen erst benutzt werden, wenn sie gebraucht werden und die Modellanpassung nicht von Anfang stören können, unterstützt der mehrphasige Aufbau nicht nur eine robuste, sondern auch eine schnelle Segmentierung. Zudem bietet er aber auch sehr gute Möglichkeiten, gezielt und frühzeitig manuell in den Segmentierungsprozeß einzugreifen und beschränkt somit die nötigen interaktiven Korrekturen auf ein Minimum.

Weiterhin ist interessant, daß bei Verwendung der erweiterten dynamischen Modelle aus Kapitel 4 nicht nur Form und Größe unabhängig gewichtet werden können (Abschnitt 4.1), sondern auch globale und lokale Anpassung, wie es Abschnitt 6.1.3 erläutert. Somit können vier elementare Eigenschaften der Modellanpassung unabhängig voneinander gewichtet werden, was eine gezielte, einfach steuerbare Fokussierung des Modells auf bestimmte Aspekte während seines Anpassungsprozesses ermöglicht.

Kapitel 7

Implementation

Dieses Kapitel behandelt verschiedene Aspekte der Implementation der in den vorangegangenen Kapiteln theoretisch vorgestellten Verfahren in einem Prototypen, der im Rahmen dieser Diplomarbeit entwickelt wurde. Neben allgemeinen Daten zu diesem Programm werden dabei einerseits spezielle Algorithmen und Datenstrukturen und andererseits bestimmte Aspekte der Benutzeroberfläche im Blickpunkt stehen.

7.1 Prototyp

Der Prototyp wurde unter Linux (2.4.21) in C++ (Compiler: gcc 3.3.1) auf einem 1,3 GHz Pentium-M-PC mit 512 MB Arbeitsspeicher und einer GeForceGo5200-Grafikkarte mit 64 MB Grafikspeicher (dem Referenz-PC) entwickelt und umfaßt ca. 24000 Zeilen Quellcode. Zur Darstellung wird ein hardwarebeschleunigtes OpenGL benutzt. Weitere spezielle Hardware ist nicht erforderlich, so daß ein moderner PC zum Betrieb der Software genügt und somit auch die in Abschnitt 7.3 erwähnte Benutzerinteraktion in Echtzeit zuläßt.

Durch die Verwendung der Programmiersprache C++ und der im folgenden aufgeführten Softwarebibliotheken ist eine Portierung auf UNIX-Derivate, neuere Mac-OS- und Windows-Versionen möglich, da es die Programmiersprache und diese Softwarebibliotheken dort auch gibt:

CGAL 3.0.1 ist eine Bibliothek für geometrische Algorithmen. Neben diversen geometrischen Primitiven fanden ihre Algorithmen zur effizienten Berechnung der konvexen Hülle und der Delaunay-Triangulierung im Prototyp Anwendung.

FFTW 3.0.1 ermöglicht eine sehr schnelle Fouriertransformation, die bei der Umsetzung von Phase 1 des automatischen Segmentierungsprozesses Verwendung fand, um die Mustererkennung per Faltung (Abschnitt 6.1.1.1) effizient zu implementieren.

VTK 4.4.1 ist ein sehr umfassendes Projekt zur Visualisierung. Es bietet komfortable Routinen zur (mittels OpenGL) hardware-beschleunigten Darstellung komplexer 2D- und 3D-Szenen und zu deren Management. Es ist die Grundlage jeglicher Daten- und Modellvisualisierung des Prototyps.

GTK+ 2.4.0 (gtkmm 2.2.10) ist eine Bibliothek für grafische Benutzeroberflächen (Menüs, Bedienelemente, Listen, etc.) samt entsprechender C++-Anbindung, mit deren Hilfe die gesamte Benutzeroberfläche des Prototypen realisiert wurde.

GtkGLExt 1.0.6 (gtkglextmm 1.0.1) ist eine Erweiterung zu GTK+ wiederum samt passender C++-Anbindung, die OpenGL-beschleunigte Elemente in der GTK+-Benutzeroberfläche zuläßt, was die Voraussetzung für in die Oberfläche eingebettete, hardware-beschleunigte Darstellungsbereiche ist.

In diesem Rahmen wurde auch eine Anbindung von VTK an GtkGLExt entwickelt, so daß es möglich wurde, VTK-Visualisierungen direkt in GTK+-Benutzeroberflächen einzubetten.

7.2 Algorithmen und Datenstrukturen

Um einen zügigen automatischen Segmentierungsprozeß zu gewährleisten und als Voraussetzung für eine intuitive und direkt rückgekoppelte Benutzerinteraktion ist eine Simulation der Modelldynamik in Echtzeit (im Sinne einer flüssigen, sichtbaren Modellbewegung) notwendig. Auf Grund der gegebenen Datensatzgröße und der dafür erforderlichen nötigen Modellkomplexität ist auch auf modernen PCs eine besonders effiziente Implementation einiger der in dieser Arbeit vorgestellten Verfahren vonnöten, um diese Anforderung zu erfüllen.

Drei interessante Beispiele dafür sollen in den folgenden drei Unterabschnitten näher beleuchtet werden.

7.2.1 Dynamische Sensorkarten

Sensorwerte (also die Sensorkräfte) sind nicht direkt aus dem Datensatz abzulesen, sondern müssen unter Umständen aufwendig berechnet werden (siehe Abschnitt 3.2 und Abschnitt 4.4), was den Gesamtrechenaufwand im Laufe der Anpassung des Modells erheblich belasten kann, da diese Berechnungen für viele Sensoren und in jedem Zeitschritt anfallen. Um letzteres zu verhindern, können die Sensorwerte jedes Sensors an allen Positionen des Datensatzes vor Beginn der Modellsimulation vorberechnet und dann zwischengespeichert werden (siehe [BERGNER 2003]), so daß bei der eigentlichen Modellsimulation nur noch auf diese vorberechneten *Sensorkarten* zurückgegriffen werden muß.

Diese Vorgehensweise hat zwei Nachteile. Erstens werden mehr Sensorwerte berechnet, als im allgemeinen nötig, da die Sensoren einen Großteil der möglichen Positionen im Datensatz im Normalfall nie annehmen (speziell an den Rändern des Datensatzes und verstärkt bei höherdimensionalen Daten).

Zweitens funktioniert dieser Ansatz in der Praxis nur, solange sich eine Gruppe von Sensoren gleichen Typs die gleiche Sensorkarte teilt, da ein zusätzlicher Speicheraufwand in Höhe des Speicherbedarfs des Datensatzes für jede vorberechnete Sensorkarte entsteht. Dieser Speicherbedarf kann sich noch vergrößern, wenn für jede Komponente des Gradienten dieser Sensorkarte ebenfalls die Werte vorberechnet und im Speicher gehalten werden, da sie sonst auch jedes mal zur Bestimmung des Sensorkraftvektors erneut berechnet werden müßten.

Für den hier vorgestellten Prototyp wurden daher *dynamische Sensorkarten* entwickelt. Diese entsprechen zu Beginn der Modellsimulation leeren Sensorkarten, die erst während der Bewegung des Modells gefüllt werden. Dabei werden die Sensorwerte erst dann berechnet, wenn sie das erste mal benötigt werden. Im gleichen Moment werden sie auch in die dynamische Sensorkarte eingetragen, wobei Speicher für sie reserviert wird. Werden die Sensorwerte der gleichen Positionen von den gleichen Sensoren wieder benötigt, so werden sie nun direkt aus der dynamischen Sensorkarte gelesen. Auf diese Weise werden nur wirklich benötigte Sensorwerte berechnet und das auch nur einmal, so daß keine Rechenzeit verschwendet und nur das wirklich Nötige gespeichert wird.

Benutzt man aufeinander aufbauende dynamische Sensorkarten, so wie z. B. auf dem Konturbild das richtungsgewichtete Konturbild aufbaut und darauf wieder seine Gradientenkomponentenbilder aufbauen, so ergeben sich Pipelines von dynamischen Sensorkarten, durch die die nötige Berechnung eines Wertes in eine Richtung und seine Abfrage in die Gegenrichtung propagiert werden. Auf Grund des in der Realität im Normalfall geringen Speicherbedarfs der einzelnen dynamischen Sensorkarten sind diese Pipelines überhaupt erst möglich geworden.

Weiterhin ist es durch den geringen realen Speicheraufwand dieser dynamischen Sensorkarten überhaupt erst möglich auf den gegebenen Daten und mit den verwendeten Modellen für jeden richtungsgewichteten Kontursensor (siehe Abschnitt 4.4) eigene Sensorkarten (inklusive der zugehörigen Gradientenkomponentenkarten) zu benutzen. Das ist notwendig, da jeder dieser Sensoren auf Grund seiner speziellen Richtung andere Sensorwerte produziert. Bei festen Sensorrichtungen sind diese Sensorkarten trotz des erheblich höheren Berechnungsaufwandes sogar besonders effizient.

Jedenfalls ergibt sich durch die Verwendung der dynamischen Sensorkarten bei der Benutzung der richtungsgewichteten Kontursensoren kein merklicher Geschwindigkeitsverlust gegenüber den nicht richtungsgewichteten Kontursensoren, so daß den besseren (siehe Abschnitt 8.5.2) richtungsgewichteten Kontursensoren problemlos der Vorzug gegeben werden kann.

7.2.2 Sensorbewegungsverfolgung

Um wie in Abschnitt 6.2 zu bestimmen, ob das Modellgleichgewicht während der Modellanpassung erreicht wurde, ist es nötig, die letzten n Positionen jedes Sensors zu kennen, da nur so über eine Stagnation der Modellbewegung entschieden werden kann.

Diese Positionen können jedem Sensor über eine eigene FIFO-Liste (First

In, First Out) der Größe n zugeordnet werden, um sie so auf effiziente Weise zu aktualisieren. Nachdem die Liste nach den ersten n Zeitschritten aufgefüllt ist, wird dabei in jedem Zeitschritt jeweils die älteste gespeicherte Sensorposition aus der Liste entfernt und die aktuelle zu ihr hinzugefügt, was beides mit konstantem, nicht merklichem Zeitaufwand möglich ist. Auf diese Weise sind immer die letzten n Positionen eines Sensors verfügbar.

7.2.3 Voxelbasierte Modellrepräsentation

Um zu einem Modell die ihm entsprechende voxelbasierte Repräsentation, wie in Abschnitt 6.3 beschrieben, effizient zu erzeugen, darf nicht für jeden Voxel des Datensatzes einzeln berechnet werden, ob er sich inner- oder außerhalb des durch die Modellkonturen begrenzten Bereiches befindet.

Daher wird im Prototyp zu dieser Arbeit ein dem Scan-Line-Prinzip (siehe [OTTMANN und WIDMAYER 1996]) verwandtes Verfahren benutzt, das die Voxelwerte, über die die Zugehörigkeit zum modellierten Objekt ausgedrückt werden soll (z. B.: 0 für nicht zugehörig und 1 für zugehörig), nicht für jeden Voxel einzeln, sondern in einem Durchgang für alle Voxel in einer Dimension (also auf einer Linie) berechnet.

Dazu wird zuerst ein leerer Datensatz, erzeugt, der so dimensioniert und positioniert ist, daß er das Modell komplett beinhaltet. Entlang einer Achse dieses Datensatzes (hier z. B. der Z-Achse) werden nun Linien durch alle Voxelmitelpunkte gelegt (hier also durch jedes X-Y-Koordinatenpaar des noch leeren Datensatzes). Für jede dieser Linien werden nun alle Schnittpunkte mit den Konturflächen des Modells berechnet und auf der entsprechenden Linie markiert.

Nun können die Voxelwerte entlang jeder Linie der Reihe nach bestimmt werden, wobei beim Eintritt der Linie in die leere Segmentierung die Voxel der Linie nicht zum durch das Modell beschriebenen Objekt gehören. Bei Erreichen eines markierten Schnittpunktes auf dem Strahl ändert sich die Art der ab dort folgenden Voxel immer abwechselnd zwischen Voxeln, die zum Objekt gehören, und solchen, die es nicht tun, so daß diese Zugehörigkeiten nun durch entsprechende Voxelwerte in der Segmentierung markiert werden können. Auf diese Weise wird Linie für Linie die anfangs leere Segmentierung jeweils mit den gewünschten Voxelwerten gefüllt.

Durch dieses Verfahren, das sich an einen zweidimensionalen Polygonfüllalgorithmus (siehe z. B. [FOLEY et al. 1996]) anlehnt, wird bei den dreidimensionalen Datensätzen dieser Arbeit nur ein zweidimensionaler Bereich (der X-Y-Schnitt der Segmentierung) sukzessive durchiteriert, da in jedem Durchgang die ganze entsprechende Z-Dimension an Voxelwerten am Stück berechnet wird. Dadurch ist es dem Prototypen möglich, aus den Modellen ihre voxelbasierten Repräsentationen in wenigen Sekunden zu erzeugen.

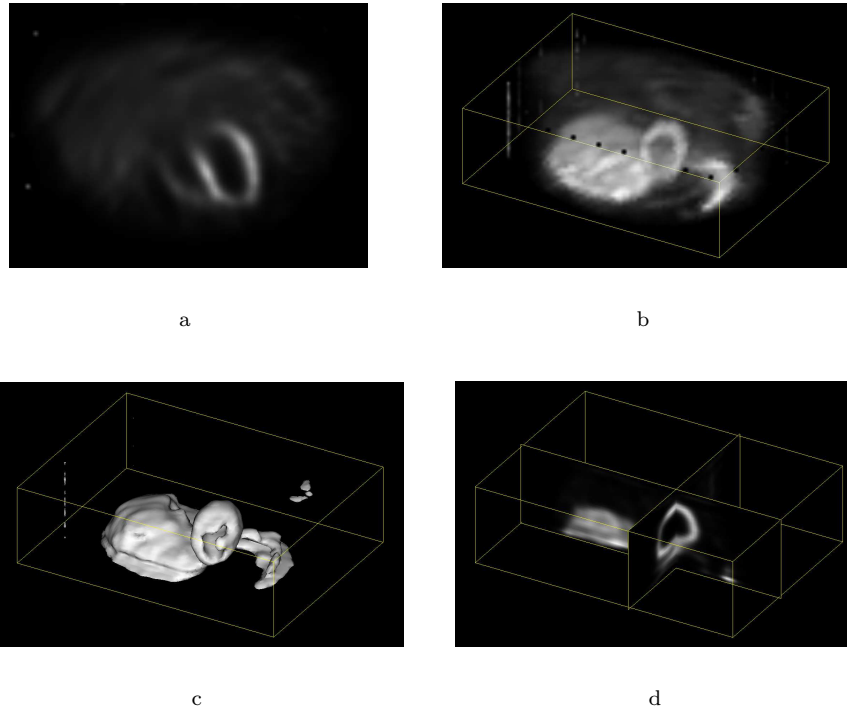


Abbildung 7.1: Verschiedene mögliche Darstellungen eines 3D-SPECT-Datensatzes im Prototypen. (a) Schichtbild. (b) Volume Rendering. (c) Iso-Oberfläche. (d) Verschiedene Schnittebenen in der 3D-Ansicht.

7.3 Benutzeroberfläche

Die Benutzeroberfläche des Prototypen vereint Steuerung und Darstellung des Modells, sowie die Interaktion mit ihm. Die Steuerung erfolgt dabei komplett über übliche GTK+-Bedienelemente.

Die Visualisierung der Modelle und Datensätze kann sowohl in 2D- als auch in 3D-Ansichten erfolgen, wobei pro Modell bzw. Datensatz mehrere verschiedene Ansichten gleichzeitig möglich sind. Während in der 2D-Ansicht der Datensatz als Schichtbild (Abbildung 7.1a) dargestellt wird, können in der 3D-Ansicht Volume-Rendering (Abbildung 7.1b), Iso-Oberfläche mit variablem Iso-Wert (Abbildung 7.1c) und frei bewegliche Schnittebenen (Abbildung 7.1d) miteinander kombiniert werden.

Auch die Modelle können auf verschiedene Weise dargestellt werden. So können sie in 2D als geschnittene geometrische Struktur (Abbildung 7.2a) oder als Projektion eines gewissen Tiefentoleranzbereiches um die Anzeigeebene (Abbildung 7.2b) angezeigt werden, während in 3D einfach ihre Polygonalstruktur (Abbildung 7.2c und 7.2d) für die Darstellung verwendet wird.

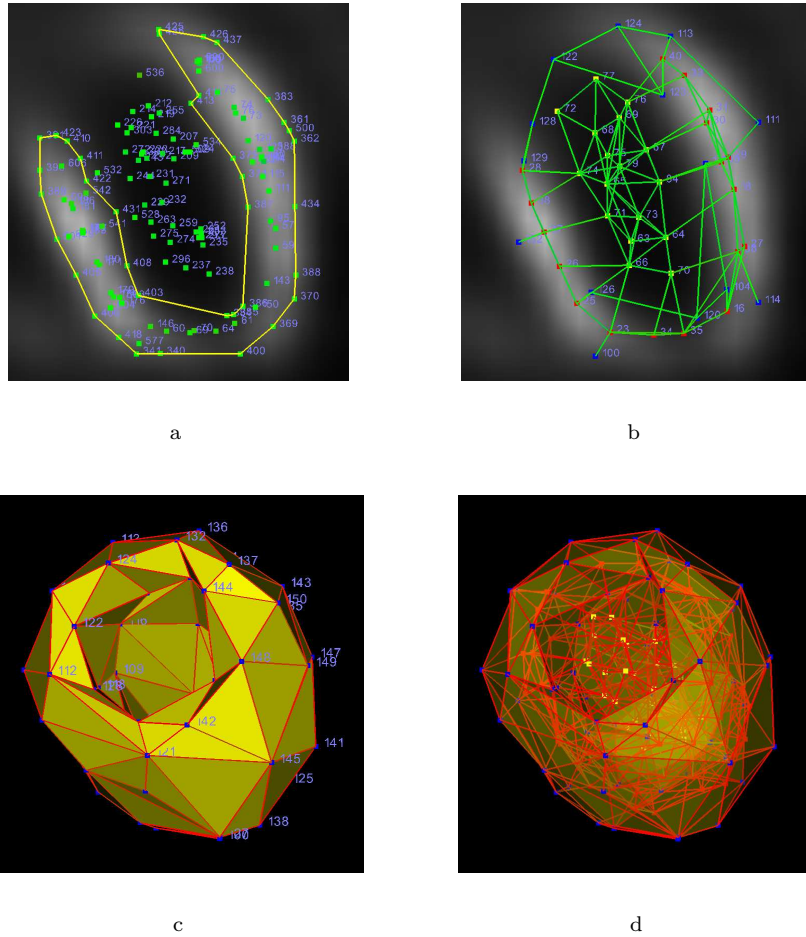


Abbildung 7.2: Verschiedene mögliche Darstellungen eines 3D-Modells im Prototypen. **(a)** 2D-Schnitt in Höhe der angezeigten Schicht des Datensatzes durch die Modellstruktur mit IDs der geschnittenen Federn. **(b)** 2D-Projektion des Modells innerhalb eines Toleranzbereichs von 2 Voxeln um die angezeigte Schicht des Datensatzes mit IDs der Sensoren. **(c)** 3D-Ansicht der Modellkontur mit Sensor-IDs. **(d)** 3D-Ansicht der gesamten Modellstruktur.

Sensoren, Federn und Konturflächen können unabhängig angezeigt werden, wobei die Darstellung nach den Typen der begrenzenden Sensoren gefiltert werden kann. Farben (z. B. für die Dehnung der Federn) und Beschriftungen (z. B. für die ID-Nummern der Sensoren) ermöglichen zudem die intuitive Darstellung von Kontextinformationen des Modells (Abbildung 7.2a bis 7.2c). Weiterhin ist es bei gleichzeitiger 2D- und 3D-Betrachtung eines Datensatzes bzw. Modells möglich, die dargestellten 2D-Ebenen in die 3D-Darstellungen einzublenden, um ihre Position im Datensatz und ihre Beziehungen zueinander leichter zu erfassen.

Die Interaktion am Modell erfolgt in der 2D-Ansicht, da in der auf einem zweidimensionalen Ausgabegerät dargestellten 3D-Ansicht keine eindeutige Positionierung möglich ist. In den 2D-Ansichten können jeweils Punkte (also Sensoren bei der Projektionsansicht bzw. Federquerschnitte in der Schnittansicht des Modells) mit der Maus verschoben werden. Dabei werden die Punkte bzw. Federn parallel zur Anzeigeebene bewegt. Eine Interaktion ist zu jedem Zeitpunkt des automatischen Segmentierungsprozesses möglich, also nicht nur zwischen den einzelnen Phasen, sondern auch in ihnen während der Modellanpassung. Natürlich kann auch nach erfolgter automatischer Segmentierung eine manuelle Feinkorrektur des Modells vorgenommen bzw. direkt die erzeugte voxelbasierte Segmentierung manipuliert werden.

Auf Grund der effizienten Implementation der Simulationsumgebung können dabei mit den in dieser Arbeit verwendeten Modell auf den gegebenen Datensätzen im Schnitt ca. 100 Simulationsschritte pro Sekunde (siehe Abschnitt 8.5.2) ohne Grafikausgabe berechnet werden, so daß durch die Verwendung hardware-beschleunigter Darstellungsmethoden eine flüssige Darstellung mit direkter Interaktion in Echtzeit sogar am laufenden Modell möglich ist.

Visualisierung und Interaktion greifen also direkt ineinander und ermöglichen durch kombinierte 2D-/3D-Ansichten verschiedener Art mit Kontextinformationen, transparenten Überblendungen und nicht zuletzt frei wählbare Kameraeinstellungen (Position, Richtung, Zoom, etc.) ein gutes, intuitives Verständnis für die Daten und die Modelldynamik und zusätzlich eine schnelle Einschätzung der Ergebnisse (der Modellgüte) samt eventuell nötigen zielgerichteten, schnell und genau vollziehbaren Korrekturen.

Kapitel 8

Evaluierung

In diesem Kapitel geht es um die Parametrisierung und Evaluierung der in dieser Arbeit entwickelten Verfahren am Beispiel der Segmentierung des LV auf den gegebenen Daten.

Neben den Datensätzen selbst werden dabei zuerst die erweiterten dynamischen Modelle aus Kapitel 4 und die zugehörige Modellgenerierung aus Kapitel 5 betrachtet.

Im Anschluß daran werden durch Experimente beobachtete Eigenschaften der erweiterten dynamischen Modelle quantitativ überprüft und eine geeignete Konfigurationen des mehrstufigen, automatischen Segmentierungsprozesses aus Kapitel 6 wird ermittelt, mit deren Hilfe zum Schluß des Kapitels der LV in den gegebenen Daten, wie in der Aufgabenstellung gefordert, segmentiert wird.

8.1 Datenmaterial

Das Datenmaterial samt zugehörigen Informationen stammt aus der Kooperation mit der Medical Imaging Research Group des Vancouver Hospital und der University of British Columbia in Kanada, und besteht aus 41 3D-SPECT-Aufnahmen von 25 verschiedenen Patienten.

Jede Schicht eines Datensatzes hat eine Auflösung von 128×128 Voxeln bei einer Voxelkantenlänge von 4.795 Millimetern. Der Wert jedes Voxels (seine Intensität) gibt in diesem Zusammenhang die rekonstruierte Anzahl der Zerfälle über den Meßzeitraum wieder, der dem Voxelvolumen zugeordnet werden konnte, ist also nicht negativ. Die Schichtanzahl der verschiedenen Datensätze variiert zwischen 25 und 38. Die LV hatten im Schnitt eine Längsausdehnung von ca. 20 Voxeln (entlang der Hauptachse) und einen Querschnittsdurchmesser von ca. 15 Voxeln. Ihre Intensität war nie größer als 25 und lag im allgemeinen sogar deutlich darunter (bis zu 75 Prozent). Zudem bestanden die Randregionen des Datensatzes aufnahmebedingt aus leeren Bereichen (der Intensität 0), die im Schnitt etwa die Hälfte des Datensatzvolumens ausmachten und die im Sinne effizienterer Berechnungen vernachlässigt wurden.

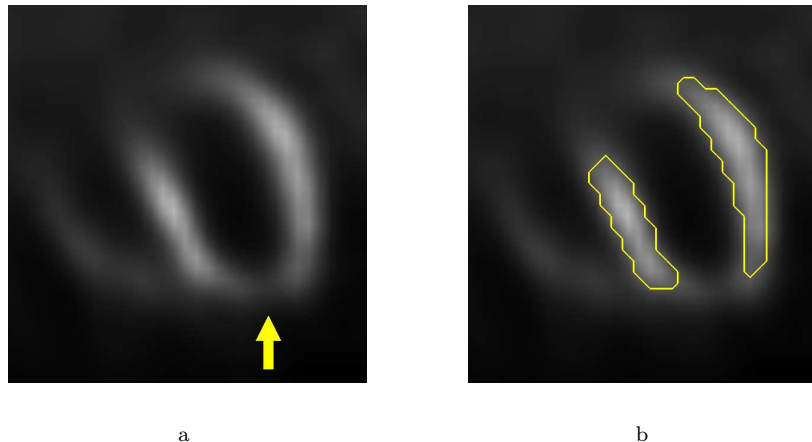


Abbildung 8.1: SPECT-Schichtbildaufnahme des linken Ventrikels, mit einer deutlich sichtbaren infarktbedingten Lücke in der hellen LV-Wand (Pfeil). In Bild (b) ist eine halbautomatisch erstellte, schwelwertbasierte Vorsegmentierung dieses Organs zu sehen.

Zu sieben dieser Datensätze (Datensatz 1 bis 7) existieren von unserem klinischen Partner manuell erstellte Segmentierungen, die im folgenden als Goldstandard dienen sollen. Diese entstanden auf Basis von Vorsegmentierungen (siehe Abbildung 8.1b), die zum Ziel hatten, das gesunde LV-Gewebe mittels eines empirisch ermittelten Schwellwertes zu isolieren. Auch das geschieht nicht vollautomatisch, da nicht nur gesundes LV-Gewebe, sondern auch gesunde Teile anderer Organe (z. B. der Leber) durch diesen Schwellwert segmentiert werden. Insgesamt dauerte die Berechnung einer Infarktgröße mittels der Erstellung einer kompletten Handsegmentierung, die den Großteil dieser Zeit beansprucht, durchschnittlich 20 Minuten bei unserem klinischen Partner.

Einige der Datensätze weisen Infarktregionen auf, die sich als Lücken in der intensitätsstarken LV-Wand bemerkbar machen, wie es z. B. in Abbildung 8.1a zu sehen ist. Diese Abbildung zeigt mit knapp 25 Prozent des LV-Wandvolumens zugleich eine der größten Infarktlücken in unseren Datensätzen, wobei beim lebenden Menschen theoretisch bis 40 Prozent möglich sind.

Alle Datensätze sind auf Grund gleicher Aufnahme- und Rekonstruktionstechnik direkt nach der Rekonstruktion gleich skaliert und ausgerichtet. Einige der Datensätze liegen uns direkt in dieser Form vor, andere sind jedoch in der Art gedreht, daß die X-Achse entlang der LV-Achse liegt und beide Herzkammern in Y-Richtung hintereinander liegen. Diese Transformation betrifft jedoch nur die Datensätze, für die keine Handsegmentierungen vorliegen.

8.2 Evaluierungsmaße

Um im weiteren Verlauf dieses Kapitels die Güte von Segmentierungen (z. B. im Vergleich zu einem Goldstandard) quantitativ einschätzen zu können, sind Vergleichsmaße nötig, die die Ähnlichkeit zweier Segmentierungen A und B (hier Mengen zur Segmentierung gehörender Voxel) beschreiben können. Da es kein absolutes Maß dieser Ähnlichkeit gibt, kann man sich verschiedener bekannter Maße bedienen, die alle etwas unterschiedlich geartete Schwerpunkte setzen (siehe [POHLE 2004] für eine allgemeine Übersicht und [HEIMANN et al. 2004] für eine detaillierte Diskussion der gebräuchlichsten Evaluierungsmaße).

Eine Möglichkeit zwei Segmentierungen A und B zu vergleichen, besteht darin, ihre Ähnlichkeit über das Volumen zu erfassen. In diesem Zusammenhang ist die *relative Über- bzw. Untersegmentierung* und deren Summe, die *relative Falschsegmentierung* von Interesse. Ebenfalls volumenbasiert ist der *Tanimoto-Koeffizient* D_T (Gleichung 8.1), der den gemeinsamen Volumenanteil an dem durch beide Segmentierungen abgedeckten Volumen berechnet.

$$D_T(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (8.1)$$

Ein anderes grundsätzliches Prinzip zur Bestimmung der gewünschten Ähnlichkeit führt über Abstandsbetrachtungen der einzelnen Voxel der Segmentierungen bezüglich einer Distanzfunktion d (in dieser Arbeit die L_2 -Norm). Hierbei kann einmal der *Hausdorff-Abstand* D_H (Gleichung 8.2) berechnet werden, der den maximalen kleinsten Abstand eines Voxels einer Segmentierung zur anderen Segmentierung angibt. Der Hausdorff-Abstand kann somit als die maximale lokale Abweichung beider Segmentierungen aufgefaßt werden. Daher reagiert er auch etwas sensibler auf Änderungen, die nur wenige oder gar einen Voxel betreffen, weil schon dadurch diese maximale Abweichung erhöht werden kann.

$$D_H(A, B) = \max\{\max_{a \in A} \min_{b \in B} d(a, b), \max_{b \in B} \min_{a \in A} d(b, a)\} \quad (8.2)$$

Weiterhin kann aber auch eine Art *durchschnittliche Konturabweichung* D_D beider Segmentierungen bestimmt werden, die sich über die durchschnittliche minimale Entfernung aller Konturvoxel beider Segmentierungen zur Kontur der jeweiligen anderen Segmentierung berechnet. Gleichung 8.3 verdeutlicht die Berechnung, wobei A_{Rand} bzw. B_{Rand} jeweils für die Menge der Konturvoxel (also Randvoxel) der entsprechenden Segmentierung steht.

$$D_D(A, B) = \frac{\sum_{a \in A_{\text{Rand}}} \min_{b \in B_{\text{Rand}}} d(a, b) + \sum_{b \in B_{\text{Rand}}} \min_{a \in A_{\text{Rand}}} d(b, a)}{|A_{\text{Rand}}| + |B_{\text{Rand}}|} \quad (8.3)$$

Mittels aller hier aufgeführter empirischer Ähnlichkeitsmaße für Segmentierungen soll im Rest des Kapitels die Güte der Segmentierung quantitativ zu einem Goldstandard in Bezug gesetzt werden, der in dieser Arbeit in Form von

durch Experten erstellter Handsegmentierungen vorliegt (vergleiche Abschnitt 8.1). Auf Grund dieser verschiedenen Maße kann so die Segmentierungsqualität unter verschiedenen Gesichtspunkten gemessen und auf breiter Basis beurteilt werden.

8.3 Erweiterte Feder-Masse-Modelle

In diesem Abschnitt sollen die wichtigsten Eigenschaften der in Kapitel 4 eingeführten erweiterten Feder-Masse-Modelle, die während der praktischen Arbeit mit ihnen beobachtet wurden, beschrieben und auf ihre Bedeutung für die Segmentierung des LV in den gegebenen Daten untersucht werden. Quantitative Belege für das Wirken einzelner aufgeführter Effekte sind im Abschnitt 8.5 zu finden.

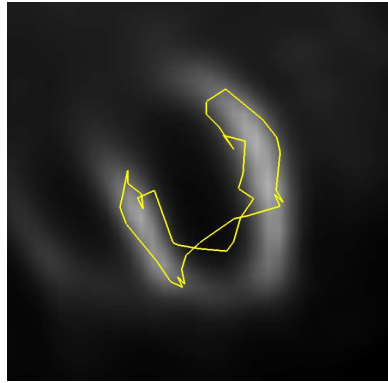
Torsionskräfte

In der praktischen Anwendung fiel zuerst die extreme Instabilität der Modellanpassung ohne die Verwendung von Torsionskräften auf. Sie kann bei ungünstigen Startbedingungen leicht mit dem kompletten Verlust der anfänglichen Modellform enden. Bei günstiger Startposition (quasi fast direkt in der gewünschten, zu findenden Anpassungsposition des Modells) ist jedoch eine teilweise stabile Anpassung ohne Torsionskräfte möglich, zu sehen in Abbildung 8.2a und 8.2c. Gut zu sehen ist dabei, daß das Modell an den Stellen der normalerweise intensitätsstarken LV-Wand kollabiert, die infarktbedingt intensitätsschwach sind und daher das Modell dort nicht stützen. Selbst das lediglich über die Ruhelängen der Federn kodierte Modellwissen kann dies nicht ausgleichen. Die Feder- und Sensor konstanten wurden dabei möglichst ausgewogen zueinander gewählt, so daß sowohl die Einflüsse der Federn (das Modell ist nicht komplett kollabiert oder unkontrolliert expandiert), als auch die der Sensoren (das Modell konnte sich lokal an gegebene Daten anpassen) während der Anpassung des Modells wahrnehmbar waren.

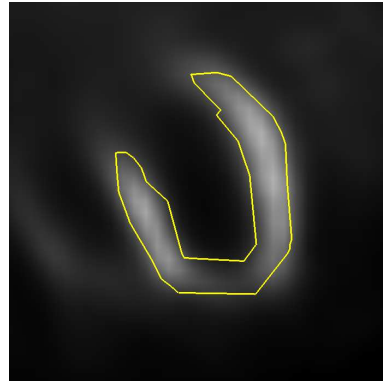
Werden die Torsionskräfte nun in der eben beschriebenen Situation aktiviert, so ist zu beobachten, wie innerhalb weniger Sekunden das Modell seinen Kollaps in der Infarktregion überwindet und durch Modellwissen gestützt, diese Region richtig segmentiert (Abbildung 8.2b und 8.2d verdeutlichen die neue Situation). Die Stabilisierung des Modells bezüglich der originalen Form findet dabei direkt und zielstrebig statt und ist so flexibel, daß die bereits richtig angepaßten Modellbereiche nicht verändert werden, sondern lediglich der Modellbereich ohne klare Datenstützung einer größeren Veränderung unterliegt.

Explizite Rotationsfähigkeit

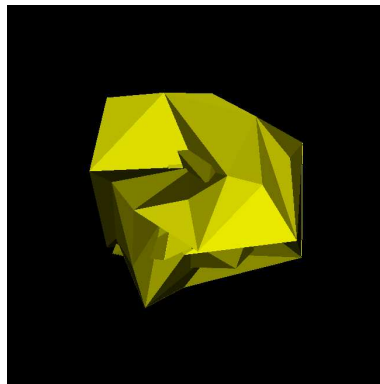
Ebenfalls die explizite Rotationsfähigkeit der erweiterten dynamischen Modelle bringt klar sichtbare positive Effekte bei der Modellanpassung gerade aus schlechter Startausrichtung heraus. So liegt z. B. in Datensatz 1 der LV bezüglich seiner Hauptachse etwas gedreht gegenüber dem LV in Datensatz 3, aus dem das



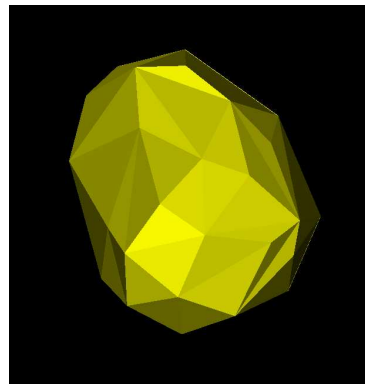
a



b



c



d

Abbildung 8.2: Anpassung eines LV-Modells an den LV in 2D- und 3D-Ansicht. **(a)**, **(c)** Ohne Verwendung von Torsionskräften. **(b)**, **(d)** Mit der Verwendung von Torsionskräften.

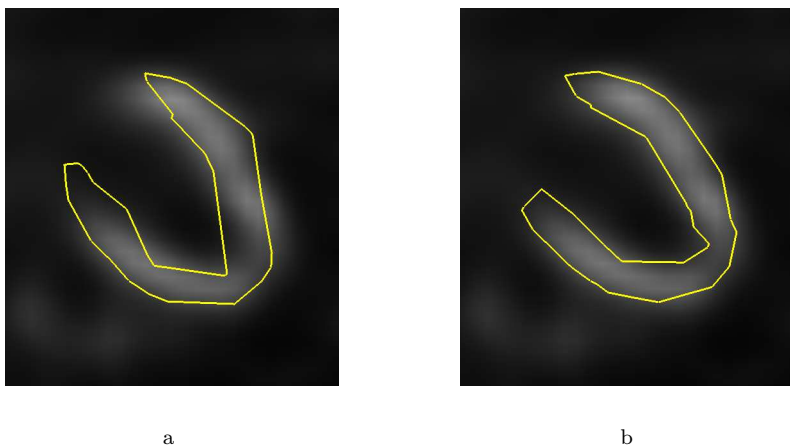


Abbildung 8.3: Anpassung eines LV-Modells an einen LV abweichender Ausrichtung unter Verwendung von Torsionskräften. **(a)** Ohne explizite Rotationsfähigkeit des Modells. **(b)** Mit expliziter Rotationsfähigkeit des Modells.

Modell generiert wurde (vergleiche Abschnitt 8.4), vor. Diese räumliche Drehung von ca. 30 Grad wird bei der Modellanpassung ohne Probleme wiederum auf direkt und robuste Art ausgeglichen, wie die Abbildung 8.3b im Vergleich zu einem vergleichbaren Anpassungsprozeß ohne explizite Rotationsfähigkeit des dort verwendeten Modells in Abbildung 8.3a zeigt, bei dem ein deutlicher Ausrichtungsunterschied bestehen bleibt.

Robustheit bezüglich der Startposition

Auch aus sehr schlechter, weil ferner Startposition heraus gelingt dem Modell durch die Torsionkräfte und die explizite Rotationsfähigkeit eine adäquate Anpassung an den LV in den Daten, wie die Abbildungsfolge 8.4a bis 8.4f illustriert. Die Kombination aus breit und flächendeckend angeordneten, über das ganze Modell verteilten Intensitätssensoren und einer durch die Torsionskräfte gegebenen Formstabilität ermöglicht hier eine zielgerichtete Anpassung auch aus entlegener Ausgangsposition heraus, wenn ein gewisser Teil des Modells sich bereits dort mit dem Zielobjektbereich im Datensatz überschneidet.

Richtungsgewichtete Kontursensoren und Konturversatz

Zwei weitere Möglichkeiten beim Einsatz dieser Modelle, namentlich die Richtungswichtung der Kontursensoren und die mögliche Konfiguration eines Konturversatzes, basieren auf den Normalen der Kontursensoren erweiterter dynamischer Modelle. Beide Effekte wirken wie beabsichtigt, was sehr klar im Bild zu erkennen ist.

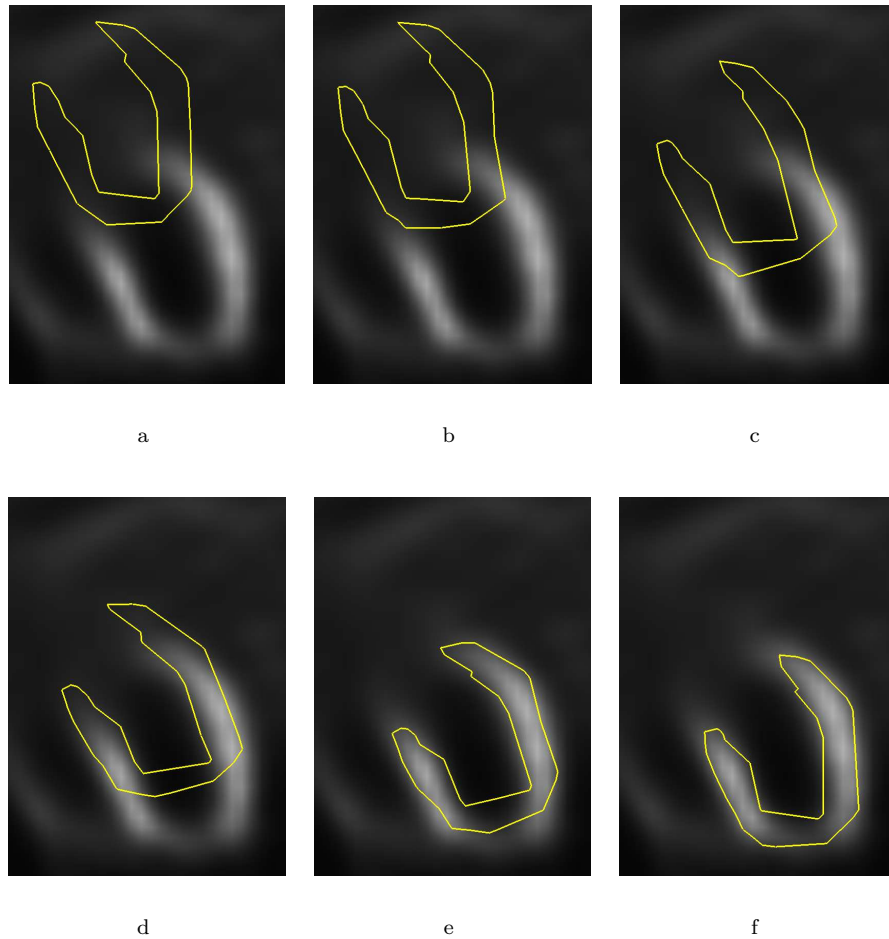


Abbildung 8.4: Anpassungsprozeß eines LV-Modells an einen LV aus sehr schlechter, weil ferner Startposition heraus. Torsionkräfte, explizite Rotationsfähigkeit und richtungsgewichtete Kontursensoren wurden benutzt. Bild **(a)** bis **(c)** zeigen, wie das Modell unter Verformung in Richtung des LV gezogen wird. Bild **(d)** bis **(f)** zeigen den Rest der Anpassung, bei dem sich das Modell mittels eines durchgängigen Kontaktes zur intensitätsstarken LV-Wand richtig auf dem LV ausrichtet.

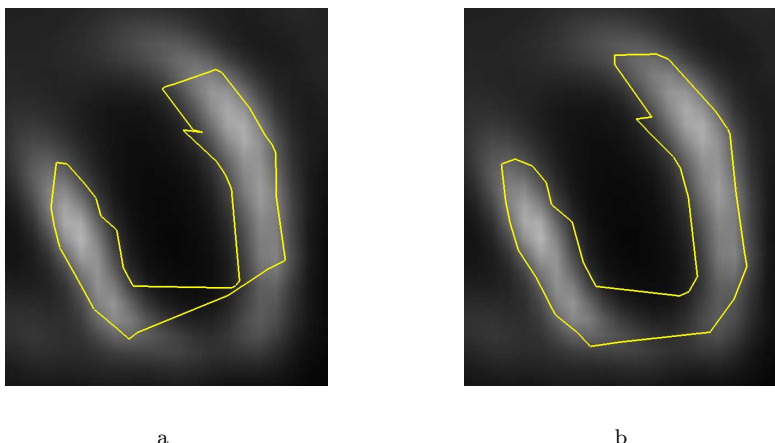


Abbildung 8.5: Einfluß der Richtungsfilterung der Kontursensoren auf die LV-Modellanpassung. **(a)** Ohne Richtungsfilterung. **(b)** Mit Richtungsfilterung. Die Anpassung wurde zur besseren Sichtbarkeit des Effektes dabei jeweils mit 5-fach stärker gewichteten Kontursensorkräften durchgeführt.

So zeigen Abbildung 8.5a und Abbildung 8.5b dieselbe abgeschlossene Modellanpassung ohne und mit Richtungswichtung der Kontursensoren. Es ist deutlich zu erkennen, daß bei deaktivierter Richtungswichtung der Kontursensoren die Konturlinie von der Objektkontur abdriftet, besonders in Bereichen einer schwachen Objektkontur (also hier im Bereich infarktbedingter Lücken in der LV-Wand). Dieses Verhalten ist in der Verschiebung der Kontursensoren zu stärkeren Konturinformationen in ihrer Umgebung begründet und wirkt sich somit direkt auf die dreidimensionale Form des Modells aus. Auf Grund der stabilisierenden inneren Kräfte des Modells ist diese Verschiebung nur lokal, aber die dadurch abnehmende Anpassungsqualität ist durch Zahlen zu belegen, wie Abschnitt 8.5.2 zeigen wird.

Ebenfalls deutlich im Bild zu erkennen sind die Auswirkungen des Konturversatzes. Abbildung 8.6b zeigt die Anpassung an einen LV ohne einen Konturversatz (der Konversatz ist 0). Im Vergleich dazu ist in Abbildung 8.6a sehr deutlich der Einfluß eines negativen Konturversatzes (also in Richtung des Inneren der Herzwand) und in Abbildung 8.6c der Einfluß eines positiven Konturversatzes (in die entgegengesetzte Richtung) bei sonst gleicher Modellanpassung zu sehen.

Wichtung der Modellverhaltensaspekte

Letztendlich sei auch angemerkt, daß die in Abschnitt 6.5 erwähnten vier unabhängigen Wichtungparameter zur Priorisierung einzelner Aspekte des Modellverhaltens in vorhergesagter Weise funktionierten. So konnte bei automatisch

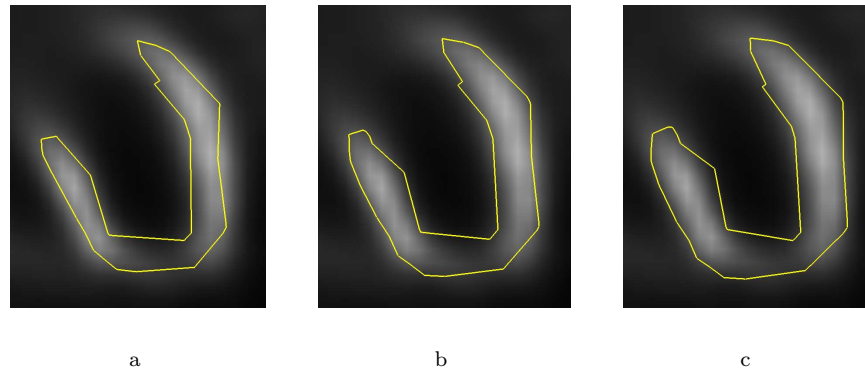


Abbildung 8.6: Einfluß eines Konturversatzes auf die LV-Modellanpassung. **(a)** Negativer Konturversatz. **(b)** Kein Konturversatz. **(c)** Positiver Konturversatz.

erzeugten, dünn vernetzten, erweiterten dynamischen Modellen die Bedeutung der modellierten Skalierung durch die Wichtung der Federkräfte und die Bedeutung der gelernten Form durch die Wichtung der Torsionskräfte gesteuert werden. Weiterhin konnte über die Wichtung der Intensitäts- bzw. Kontursensoren die Bedeutung der globalen bzw. lokalen Anpassung für die verwendeten Modelle kontrolliert werden.

Fazit

Die erweiterten dynamischen Modelle ermöglichen also auf dreidimensionalen Daten eine zielstrebige Segmentierung, die robust gegen kleinere bis mittlere Abweichungen der Startposition (idealerweise direkt auf dem zu segmentierenden Objekt) und -ausrichtung (idealerweise gleich der Ausrichtung des zu segmentierenden Objektes) ist. Sie sind extrem stabil bezüglich ihrer originalen Modellform, was jedoch ihre Anpassungsfähigkeit an die gegebenen Daten bei angenommener Adäquatheit des Modellierungsansatzes in keiner Weise einschränkt (zudem der Einfluß dieser Stabilisierungskomponente fließend gewichtet werden kann). Dadurch ist es ihnen möglich, auch größere Datenbereiche ohne Sensoreinformationen durch Modellwissen korrekt zu ergänzen.

8.4 Modellgenerierung

Mittels des in Kapitel 5 vorgestellten Verfahrens konnten dynamische 3D-Modelle aus einer einzigen Handsegmentierung innerhalb sehr kurzer Zeit erstellt werden, die die anatomische Struktur des LV adäquat modellieren. Im folgenden sollen die für diese Modellerzeugung günstigen Parameter und die Qualität des generierten Modells erörtert werden.

Modellgenerierungsparameter	Wert in Voxeln
Startsensorabstand auf Iso-Gitter	1.0
minimaler Intensitätssensorabstand	1.3
minimaler Kontursensorabstand	2.0
max. Sensorabstand bei Konturauffüllung	4.0
gruppeninterner Vernetzungsbereich	5.5

Tabelle 8.1: günstige Parameter zur automatischen Generierung eines dynamischen Modells auf den gegebenen Daten

Vor einer solchen Modellerstellung muß dabei der Bereich der Handsegmentierung gesondert markiert werden, in dem Intensitätssensoren auf schwache Intensitäten reagieren, also der innere LV-Bereich. Wird dieser Bereich mittels Subtraktion des LV von seiner konvexen Hülle, wie in Abschnitt 5.1 beschrieben, ermittelt, so war als minimale Volumenbedingung zur Entfernung dabei entstehender Artefakte eine Größe von 27 Voxeln in allen Fällen auf den gegebenen Handsegmentierungen hinreichend.

In den praktischen Experimenten zur Modellanpassung so erzeugter erweiterter dynamischer Modelle war eine 1:1-Sensor-Verknüpfung im Bereich der Sensorgruppenvernetzung (statt der in Abschnitt 5.2.3 genannten 1: n -Verknüpfung) für die Modellstabilität hinreichend. Diese Tatsache ist direkt in der Verwendung der stabilen erweiterten Feder-Masse-Modelle aus Kapitel 4 begründet. Somit wiesen die Modelle eine minimale Komplexität bezüglich der beschriebenen Generierungsprozedur auf, die zur hohen Geschwindigkeit der kompletten Segmentierung (deutlich unter einer halben Minute auf modernen PCs, siehe Abschnitt 8.5.2) beitrug.

Als weitere geeignete Parameter (vergleiche Kapitel 5) für die Modellerzeugung aus den gegebenen Daten stellten sich die in Tabelle 8.1 aufgeführten Werte heraus. Die Parameter zur Modellerzeugung lassen dabei eine Schwerpunktsetzung auf verschiedene Bereiche des generierten Modells zu, sind jedoch nicht gänzlich unabhängig voneinander. So hat z. B. der minimale Kontursensorabstand im allgemeinen Einfluß auf die sinnvolle Wahl der Auffüllichte für Sensoren in ebenen Modellkonturbereichen.

Mittels der Generierungsparameter aus Tabelle 8.1 wurde auf einem der gegebenen Datensätze mit vorliegender Handsegmentierung ein Modell generiert (siehe Abbildung 8.7), das im weiteren Verlauf dieses Kapitels zur Segmentierung des LV auf diesen Datensätzen eingesetzt werden soll.

Das Modell wurde auf Datensatz 3 erzeugt, weil seine Handsegmentierung einen gut geformten LV enthielt, der adäquat schien, als allgemeines Beispiel für einen LV zu dienen. Es hätte im Prinzip auch jede andere gegebene Handsegmentierung eines Datensatzes die Basis für dieses Modell bilden können, das für alle Experimente in diesem Kapitel die Grundlage bilden soll.

Tabelle 8.2 und Tabelle 8.3 geben in diesem Zusammenhang Auskunft über einige quantitative Eigenschaften dieses Modells, dessen Erzeugung auf dem

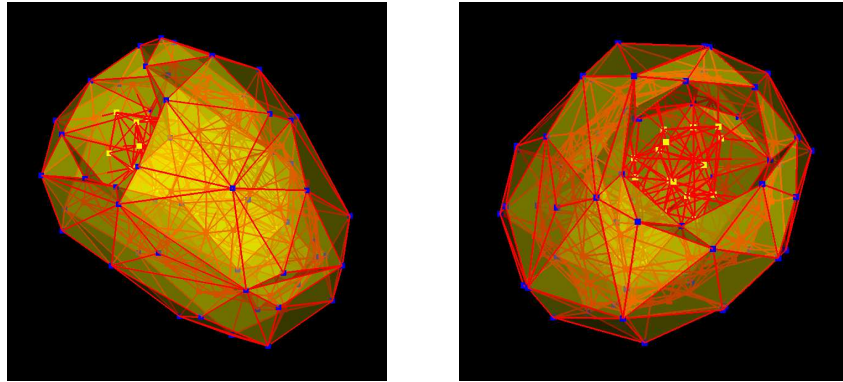


Abbildung 8.7: 3D-Ansichten des mittels der Handsegmentierung zu Datensatz 3 generierten Modells

Modellelemente	Anzahl
Sensoren	153
Federn	630
Konturflächen	124

Tabelle 8.2: Anzahl der einzelnen Elemente des erzeugten dynamischen Modells

Referenz-PC ca. 15 Sekunden dauerte. Interessant ist dabei neben der Anzahl der einzelnen Elemente dieses Modells der mittlere Vernetzungsgrad der Sensoren von ca. 8. Er zeigt, wie aufwendig es bereits wäre, solch ein (nach Maßgabe der Generierungsprozedur) schwach vernetztes Modell manuell zu erstellen, so daß es die gewünschte Struktur aus der Handsegmentierung adäquat beschreibt.

Die guten Segmentierungsergebnisse in Abschnitt 8.6 hängen eng mit der Qualität des dort benutzten, automatisch erzeugten dynamischen Modells zusammen. Es beschreibt die zu segmentierenden Strukturen mit nur sehr wenigen Elementen (Sensoren und Federn) formgenau und räumlich ausgewogen. Dadurch bietet es gegenüber vergleichbar dimensionierten, komplett manuell

Sensortyp	Anzahl	mittlerer Vernetzungsgrad
Intensität: stark	62	8.71
Intensität: schwach	27	10.52
Kontur	64	6.81
gesamt	153	8.24

Tabelle 8.3: Anzahl und mittlerer Vernetzungsgrad der verschiedenen Sensortypen des erzeugten dynamischen Modells

erzeugten Modellen neben der bedeutend schnelleren und für den Benutzer einfacheren Generierung auch eine vermutlich simplere, aber doch adäquate Strukturbeschreibung des zu segmentierenden Objektes.

Im Verlauf der Arbeiten hat sich herausgestellt, daß diese Modellerzeugung sehr allgemein anwendbar ist, da sie kaum Anforderungen an die Art der zu segmentierenden Strukturen stellt. Aus beliebigen 2D- oder 3D-Beispielsegmentierungen können so schnell passende Modelle für vielfältige Strukturen erzeugt werden.

8.5 Experimente zum Segmentierungsprozeß

Der in Kapitel 6 entwickelte automatische Segmentierungsablauf läßt in den einzelnen Phasen mehrere Möglichkeiten zur Parametrisierung offen, die nun in praktischen Tests auf ihre Wirkung hinsichtlich einer hohen Segmentierungsqualität untersucht werden sollen. Gleichzeitig soll dabei die geeignetste Konfiguration dieser Parameter für die gegebene Aufgabe ermittelt werden.

Weiterhin sollen mit Hilfe dieser Experimente die bereits in Abschnitt 8.3 qualitativ beschriebenen Eigenschaften erweiterter dynamischer Modelle quantitativ überprüft werden.

8.5.1 Phase 1

In Phase 1 wird das Modell mit Hilfe einer Mustererkennung initial plaziert. Ziel dieser Plazierung ist dabei, das Modell so zu positionieren, daß der Modellschwerpunkt im Inneren des LV zu liegen kommt, so daß das Modell eine gute Chance hat, sich, wie in Abschnitt 8.3 beschrieben, in die korrekte Endposition zu schieben.

Tabelle 8.4 zeigt, daß das faltungsbasierte Verfahren (**Faltung**) auf den sieben untersuchten Datensätzen, zu denen es einen Goldstandard gibt, lediglich einmal in Datensatz 4 versagt hat. Das geschah auf Grund eines Artefaktes, dessen Intensitätswert ein 20-faches des LV in diesem Datensatz überstieg. Das auf den Mustervektorähnlichkeiten basierende Verfahren (**Vektor**), versagte hingegen nie.

Trotz des Fehlers in Datensatz 4 ist das faltungsbasierte Verfahren eine gute Wahl für den Normalfall, in dem keine extrem hohen Strahlungsartefakte in den Datensätzen vorkommen. Unter allen 41 gegebenen Datensätzen ist Datensatz 4 der einzige mit einem solchen Artefakt. Der Hauptvorteil des faltungsbasierten Verfahrens liegt in der durch die schnelle Fouriertransformation extrem effizienten Berechnung, so daß die Mustererkennung hier pro Datensatz unter einer Sekunde Berechnungszeit benötigte, wohingegen das Verfahren mittels der Mustervektoren ca. 35 Sekunden brauchte. Auf Grund seiner Natur ist es jedoch für Artefakte hoher Intensität nicht anfällig und sollte daher als Option für artefaktbelastete Datensätze zur Verfügung stehen.

Datensatz	Faltung	Vektor
1	richtig	richtig
2	richtig	richtig
3	richtig	richtig
4	falsch	richtig
5	richtig	richtig
6	richtig	richtig
7	richtig	richtig

Tabelle 8.4: Richtigkeit der Startpunktbestimmung mittels Mustererkennung durch Faltung (**Faltung**) bzw. Vektorähnlichkeiten (**Vektor**)

8.5.2 Phase 2 und 3

Die optimale Konfiguration der Phasen 2 und 3 der automatischen Segmentierung soll nun in einem Schritt ermittelt werden, da diese Phasen direkt aufeinander aufbauen und eine schlüssige Evaluation der Ergebnisse erst nach Phase 3, also am Schluß der automatischen Segmentierung wieder möglich ist. Es gilt dabei zu untersuchen, welche Konfiguration der sieben folgenden noch aus den Abschnitten 6.1.2 und 6.1.3 offenen Parameter am geeignetsten für eine automatische Segmentierung auf den gegebenen Daten ist:

Parameter in Phase 2:

Modellkompression (aktiv / inaktiv): Sie entscheidet, ob das Modell in Phase 2 nahezu auf einen Punkt komprimiert wird und so garantiert komplett innerhalb des zu segmentierenden LV startet oder ob es Phase 2 ohne Veränderung seiner initialen Form beginnt.

Rotationsfähigkeit (freit / gesamt): Hier wird festgelegt, ob die Intensitätssensoren in Phase 2 völlig frei rotieren können oder ob alle gleichförmig die gemittelte berechnete Einzelrotation aller Intensitätssensoren ausführen, um so das Modell rotationsstabil in sich selbst zu halten.

Kontursensoren (inaktiv / offline): Hier kann gewählt werden, ob die in Phase 2 planungsgemäß nicht beteiligten Sensoren komplett inaktiv oder nur offline geschaltet werden, so daß sie sich durch die inneren Kräfte des Modells mitbewegen und drehen und in Phase 3 eventuell gleich von einer guten, der Modellerwartung entsprechenden Position aus starten können.

Parameter in Phase 3:

Konturkompression (aktiv / inaktiv): Durch die Kompression der Kontur(sensoren) vor Beginn von Phase 3 auf die mit ihnen verbundenen Intensitätssensoren wird gewährleistet, daß die Kontursensoren stets aus

dem Inneren der LV-Wand starten. Ansonsten beginnen sie unverändert an ihren nach Phase 2 erreichten Positionen.

Rotationsfähigkeit (frei / gesamt / inaktiv): Dieser Parameter entspricht dem gleichnamigen Parameter in Phase 2, jedoch kommt noch eine weitere Option hinzu, namentlich das Deaktivieren der Rotationsfähigkeit bei sonst freier Beweglichkeit aller Sensoren, die in Phase 2 ja bereits ihre Ausrichtung angepaßt haben sollten.

Intensitätssensoren (fest / frei): Hier kann entschieden werden, ob die Intensitätssensoren, die sich in Phase 2 bereits angepaßt haben sollten, in Phase 3 an ihren Position fest verankert werden oder weiterhin frei beweglich bleiben.

Richtungsgewichtung (aktiv / inaktiv): Hier steht zur Wahl, ob in Phase 3, in der die Konturanpassung erfolgen soll, die Kontursensoren auf richtungsgewichteten Gradientenkarten, wie in Abschnitt 4.4, arbeiten sollen, um verstärkt auf Konturen im Datensatz lokal parallel zur Modellkontur an den entsprechenden Kontursensorpositionen zu reagieren, oder auf den normalen Gradientenkarten ohne diese Wichtung.

Um den Einfluß der verschiedenen Parameter auf den automatischen Segmentierungsprozeß zu untersuchen, wird der automatische Segmentierungsprozeß mit allen möglichen Kombinationen der Optionen der sieben eben beschriebenen Parameter ($2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 \cdot 2 \cdot 2 = 192$ an der Zahl) durchgeführt.

Datenmaterial

Es konnte dabei auf Grund der großen Anzahl der Parameterkombinationen und der damit nötigen Segmentierungen pro Datensatz nur auf drei speziell ausgewählten Datensätzen segmentiert werden, die im folgenden beschrieben werden.

Datensatz 1, der kleinere Infarkte (also kleinere Lücken in der LV-Wand) aufweist und bei dem die Ausrichtung der LV-Hauptachse sichtbar (ca. 30 Grad) von der des Modells abweicht, der LV also deutlich gedreht bezüglich des Modells im Datensatz liegt

Datensatz 2, der einen großen Infarkt zeigt, also wieder die entsprechende große Lücke hat, und der im Vergleich zum Modell einen etwas gestauchten LV besitzt, der eher rundlich als länglich ist, also eine deutliche Formabweichung bezüglich des Modells aufweist

Datensatz 3, der wiederum eine große infarktbedingte Lücke in der LV-Wand besitzt und auf dessen vorliegender Handsegmentierung das benutzte Modell erstellt wurde (siehe Abschnitt 8.4), dessen LV also in Form und Ausrichtung dem Modell nahe kommt

Diese drei Datensätze wurden gezielt so ausgewählt, daß große Form- und Lagevarianzen des LV bereits in ihnen vorhanden waren. So kann man davon ausgehen, daß die hier zu ermittelten geeigneten Parameter besser an das allgemeine Problem der LV-Segmentierung statt nur an einen speziellen Datensatz angepaßt sind.

Experimente

Es wurden 192 Einzelsegmentierungen auf jedem der Datensätze 1 bis 3 durchgeführt, so daß jede mögliche Parameterkombination des automatischen Segmentierungsprozesses einmal pro Datensatz benutzt wurde.

Die Simulation wurde mit den im Anhang A in Tabelle A.1 aufgeführten Modellparametern und den in Tabelle A.2 gezeigten Simulationsparametern durchgeführt. Diese stellten sich bei den Experimenten im Verlauf dieser Arbeit als günstig heraus. Es soll jedoch betont werden, daß das Modellverhalten sehr robust in Bezug auf die Wahl dieser Parameter ist, da selbst Multiplikation einzelner Parameter mit bis zu 10 oder Division einzelner Parameter durch bis zu 10 das Modellverhalten nicht grundlegend zu ändern schien, sondern sich primär auf die Anpassungsgeschwindigkeit auswirkte.

Wie in der Tabelle der Modellparameter zu sehen ist, wird jeweils kein Konturversatz (siehe Abschnitt 4.5) verwendet, da er im eigentlichen Sinn keinen Einfluß auf die Art und Güte der Segmentierung hat, sondern eine Segmentierung gleichmäßig vergrößern bzw. verkleinern kann, um so einer systematischen Unter- bzw. Übersegmentierung entgegenzuwirken. Zudem wurden stets die in Abschnitt 4.1 eingeführten Torsionskräfte benutzt, da ohne sie das Modell im allgemeinen teilweise kollabiert (siehe Abschnitt 8.3).

Die erzielten Ergebnisse wurden für jeden Datensatz mittels seines jeweiligen Goldstandards evaluiert. Unter der empirischen Annahme einer gewissen Unabhängigkeit der Parameter, wurden für jeden Parameter jeweils die Ergebnisse aller Segmentierungen nach seinen möglichen Optionen gruppiert, gemittelt und einander gegenübergestellt. Auf diese Weise kann der Einfluß einzelner Parameteroptionen auf den gesamten Segmentierungsprozeß abgeschätzt werden.

In Tabelle 8.5 sind diese Ergebnisse zusammengefaßt. Neben der relativen Falschsegmentierung in Prozent sind dabei jeweils der Hausdorff-Abstand und der mittlere Oberflächenabstand in Voxeln (vox) angegeben, die sich allesamt auf den Goldstandard beziehen.

Diskussion

Es ist gut zu erkennen, daß sowohl die Parameter **Modellkompression**, als auch **Konturkompression** im Prinzip keinen Einfluß auf die Qualität der Segmentierung haben. Das ist überraschend, mag aber darin begründet liegen, daß die Modellanpassung so stabil und zielstrebig ist, daß die vermeintlichen Verbesserungen der Startpositionen einzelner Sensoren im Endeffekt keinen Gewinn bringen.

Parameter Phase 2	Status	Falschseg.	Hausdorff.	Mittl. Abst.
Modellkompression	aktiv	31.0 %	2.50 vox	0.447 vox
Modellkompression	inaktiv	30.9 %	2.49 vox	0.446 vox
Rotationsfähigkeit	frei	31.1 %	2.67 vox	0.454 vox
Rotationsfähigkeit	gesamt	30.8 %	2.32 vox	0.440 vox
Kontursensoren	inaktiv	31.8 %	2.59 vox	0.453 vox
Kontursensoren	offline	30.1 %	2.40 vox	0.441 vox
Parameter Phase 3	Status	Falschseg.	Hausdorff.	Mittl. Abst.
Konturkompression	aktiv	31.0 %	2.51 vox	0.447 vox
Konturkompression	inaktiv	30.9 %	2.47 vox	0.446 vox
Rotationsfähigkeit	frei	30.9 %	2.51 vox	0.450 vox
Rotationsfähigkeit	gesamt	30.8 %	2.43 vox	0.443 vox
Rotationsfähigkeit	inaktiv	31.2 %	2.54 vox	0.447 vox
Intensitätssensoren	fest	31.0 %	2.54 vox	0.447 vox
Intensitätssensoren	frei	30.9 %	2.44 vox	0.446 vox
Richtungsgewichtung	aktiv	29.0 %	2.49 vox	0.436 vox
Richtungsgewichtung	inaktiv	32.9 %	2.49 vox	0.457 vox

Tabelle 8.5: Gegenüberstellung der Auswirkungen aller Optionen jeweils eines Parameters des automatischen Segmentierungsprozesses auf die über alle untersuchten Datensätze (1 bis 3) gemittelte Segmentierungsgüte in Bezug auf den Goldstandard

Schaut man speziell auf Phase 2, so ist zu sehen, daß eine für das Modell einheitliche Rotationsfähigkeit leichte Verbesserungen zu bringen scheint. Das macht in sofern Sinn, als daß bei stimmiger Modellinformation keine interne Verdrehung von Teilen der Modells nötig scheint, zumal auch bei einer Gesamtmodellrotation noch genug Spielraum für kleinere lokale Anpassungen ist. Die Freiheit eines jeden Sensors zu individuellen Rotation könnte also mehr ungewollte Unruhe als beabsichtigten zusätzlichen Freiraum bringen, der auf diesen Daten nicht nötig scheint.

Ebenfalls in Phase 2 fällt das Verhalten der nicht beteiligten Kontursensoren. Auch hier scheint es so, als ob offline geschaltete Kontursensoren zu kleinen Verbesserungen gegenüber völlig inaktiven Kontursensoren führen. Auch das erscheint stimmig, da erstere sich so schon in Phase 2 aktiv an der Modellbewegung beteiligen, also sich, durch die innere Energie bedingt, mitbewegen und auch mitrotieren. Somit hat das Modell sie zum Ende der Phase 2 bereits an den Stellen positioniert und auch entsprechend rotiert, an dem es nur auf Grund seiner Modellinformation die Konturen erwartet. Das verbessert natürlich ihre Chancen zur sauberen Segmentierung dieser Konturen, da sie die während der Grobanpassung in Phase 2 gewonnen Informationen im Gegensatz zu inaktiven Kontursensoren direkt aufnehmen können.

Die Rotationsfähigkeit der Sensoren in Phase 3 scheint wie auch die Kom-

pressionsparameter keinen großen Einfluß auf die durchschnittliche Segmentierungsgüte zu haben. Man kann hier mutmaßen, daß auf Grund der bereits in Phase 2 erreichten guten Lage der Intensitätssensoren und auch der damit verbundenen bereits feststehenden ungefähren Ausrichtung der in Phase 3 benutzten Kontursensoren keine großen Veränderungen in der Ausrichtung der einzelnen Sensoren mehr nötig sind, so daß hier eine eingeschränkte bzw. deaktivierte Rotationsfähigkeit ausreicht, eine freie jedoch auf Grund der stabilen, bereits gut durch die Daten gestützten Lage nicht schadet.

Auch eine eventuelle feste Verankerung der Intensitätssensoren in Phase 3 hat den Werten nach interessanterweise keinen Einfluß auf die letztendliche Qualität des Segmentierungsergebnisses. Die Modellanpassung in Phase 3 scheint so stabil und robust zu sein, daß die aus einer festen Verankerung der bereits in Phase angepaßten Intensitätssensoren resultierende Stabilität nicht benötigt wird. Sie ist aber auch nicht schädlich, was die These einer bereits vollständigen Anpassung der Intensitätssensoren in Phase 2 stützt, wie sie in Abschnitt 6.1.3 entwickelt wurde.

Zuletzt bleibt der Parameter **Richtungsgewichtung** zu betrachten. Hier ist die Verbesserung der Segmentierungsgüte bei Verwendung richtungsgewichteter Kontursensoren wieder leicht, aber doch etwas deutlicher als bei anderen Parametern festzustellen. Das ist auch nicht verwunderlich, da schon Abschnitt 8.3 die Vorzüge dieser Sensorart gerade bei der lokalen Anpassung unterstreicht.

Man kann also zusammenfassen, daß einige Parameter keinen nachweisbaren Einfluß auf die Segmentierungsgüte haben, und daher nicht berücksichtigt werden brauchen. Im Falle von **Modellkompression** und **Konturkompression** heißt das, daß nicht extra Kompressionen durchgeführt werden sollten. Für den Parameter **Rotationsfähigkeit** in Phase 3 ist die gewählte Option ebenfalls den Ergebnissen nach im Prinzip egal, deshalb soll hier aus den gleichen Gründen wie in Phase 2 dem Modell eine Gesamtrotation ermöglicht werden. So kann sich das Modell nicht ineinander verdrehen, hat aber trotzdem noch etwas Spielraum zur letztendlichen Anpassung. Auch für den Parameter **Intensitätssensoren** ist kein großer Einfluß auf die Güte des Gesamtsegmentierungsergebnisses auszumachen, so daß die Intensitätssensoren hier nicht fest verankert werden sollen, da so nur eine gewisse Unbeweglichkeit in der letzten Anpassungsphase erzeugt wird, deren resultierende Stabilität den Experimenten nach aber nicht notwendig erscheint.

Fazit

Tabelle 8.6 faßt die resultierende günstige und effiziente Parameterwahl für den automatischen Segmentierungsprozeß zusammen, die auch die Grundlage für die automatische Segmentierung der Datensätze bilden soll. Diese Parameter ermöglichen zwar keine grundlegenden Verbesserungen, können aber in ihrer Summe doch etwas die erreichte Segmentierungsgüte steigern. Eine perfekte Wahl gibt es nicht, da schon in den Tests auf den drei Datensätzen jeweils verschiedene Parameterkonfigurationen optimal waren, teilweise sogar verschiedene für die verschiedenen Evaluierungsmaße. Die hier in Tabelle 8.6 gewählte

Parameter Phase 2	Geeigneter Wert
Modellkompression	inaktiv
Rotationsfähigkeit	gesamt
Kontursensoren	offline
Parameter Phase 3	Geeigneter Wert
Konturkompression	inaktiv
Rotationsfähigkeit	gesamt
Intensitätssensoren	frei
Richtungsgewichtung	aktiv

Tabelle 8.6: geeignete Parameterkonfiguration des automatischen Segmentierungsprozesses

Parameterkonfiguration war dabei immer unter den besten.

Weitere Ergebnisse

Neben der Bestimmung einer geeigneten Parameterkonfiguration im vorhergehenden Unterabschnitt konnten noch weitere Erkenntnisse durch die Experimente zu diesen Konfigurationen gewonnen werden.

Modellgleichgewicht

Bei den Experimenten zur Bestimmung der geeigneten Phasenkonfiguration fiel auf, daß das in Abschnitt 6.2 vorgestellte Stopkriterium der Modellbewegung zur Feststellung des Eintritts des Modellgleichgewichtes im Laufe der gesamten Experimente ausnahmslos reibungsfrei funktionierte und die Phase 2 im Schnitt nach 1450 Simulationsschritten und die Phase 3 durchschnittlich nach 624 Simulationsschritten jeweils korrekt nach Einstellung des Modellgleichgewichtes die Modellbewegung beendet hat. 1000 Schritte wurden dabei jeweils ohne Grafikausgabe in ca. 5 bis 15 Sekunden je nach Parameterkonfiguration simuliert, so daß eine komplette automatische Simulation bei diesen Experimenten im Schnitt nur ca. 20 Sekunden dauerte.

2-Phasen-Ansatz

Weiterhin wurde untersucht, ob 3 Phasen, wie sie in Kapitel 6 motiviert wurden, überhaupt nötig sind, oder 2 Phasen ausreichen, indem Phase 2 und 3, die ja beide das dynamische Modell anpassen, quasi verschmolzen werden. In Tabelle 8.7 sind gesondert für jeden der drei schon im vorigen Abschnitt untersuchten Datensätze die jeweils besten erreichten Werte der einzelnen Evaluierungsmaße unter allen Parameterkombinationen für automatische 2- bzw. 3-Phasensegmentierungen aufgeführt. Bei den 2-Phasen-Ansätzen

Datensatz	Phasen	Falschseg.	Hausdorff.	Mittl. Abst.
1	2	28.8 %	2.24 vox	0.462 vox
1	3	27.5 %	2.00 vox	0.448 vox
2	2	26.0 %	1.41 vox	0.401 vox
2	3	25.4 %	1.41 vox	0.384 vox
3	2	27.2 %	1.73 vox	0.366 vox
3	3	26.2 %	1.73 vox	0.363 vox

Tabelle 8.7: Bestwerte der einzelnen Evaluierungsmaße der automatischen Segmentierungsprozesse aller Parameterkonfigurationen auf allen untersuchten Datensätzen (1 - 3) gruppiert nach 2- bzw. 3-Phasen-Ansätzen

wurden dabei die Parameter entfernt, die nur bei 3 Phasen Sinn machen, also *Kontursensoren*, *Konturkompression*, *Rotationsfähigkeit* (aus Phase 3) und *Intensitätssensoren*.

Es ist zu sehen, daß der 3-Phasen-Ansatz leicht überlegen ist, jedoch nicht so deutlich, wie vielleicht nach den Motivationen in Kapitel 6 zu erwarten war. Eine mögliche Erklärung für dieses Verhalten ist, daß das stabile und zielstrebige Modellverhalten die stabilisierenden und die die Zielstrebigkeit unterstützenden Effekte des 3-Phasen-Aufbaus, der zuerst die Konzentration auf die Intensitätssensoren lenkt und sich nach deren Anpassung auf die Kontursensoren konzentriert, praktisch kaum erfordert. Das mag unter anderem auch daran liegen, daß der 3-Phasen-Ansatz eine direkte Anpassung vor allem aus schlechten Ausgangspositionen durch anfängliche Nichtbenutzung der Kontursensoren fördern soll, hier aber durchweg durch Phase 1 nur sehr gute Ausgangspositionen für die Modellanpassung erzeugt wurden, so daß sich eventuelle Vorteile des 3-Phasen-Ansatzes nicht deutlich herauskristallisieren konnten.

Zwei weitere Vorteile des 3-Phasen-Ansatzes bleiben jedoch davon unberührt. Der erste ist die Möglichkeit, die richtungsgewichteten Kontursensoren bei deaktivierter Rotation besonders effizient in Phase 3 einsetzen zu können (siehe Abschnitt 7.2.1), da ja die zur Anpassung nötige Rotation des Modells bereits in Phase 2 nur mittels der Intensitätssensoren stattgefunden hat. Der zweite Vorteil liegt in der klareren Strukturierung des gesamten Anpassungsprozesses gegenüber einer 2-Phasen-Segmentierung. Auf diese Weise kann zwischen Phase 2 und 3 sozusagen mitten in der Modellanpassung leicht eine Zwischeneinschätzung der Benutzers mit eventueller Korrektur vorgenommen. Da nach dieser Korrektur das Modell weiter angepaßt wird, findet die Interaktion an dieser Stelle besonders frühzeitig im Segmentierungsprozeß statt und muß nicht sonderlich präzise sein. Auf Grund all dessen soll zur endgültigen Segmentierung der gegebenen Daten auch der 3-Phasenansatz gewählt werden.

Datensatz	Torsion.	Falschseg.	Hausdorff.	Mittl. Abst.
1	nein	148 %	5.74 vox	1.08 vox
1	ja	28.6 %	2.24 vox	0.457 vox
2	nein	153 %	7.48 vox	1.34 vox
2	ja	25.7 %	1.73 vox	0.390 vox
3	nein	53.4 %	3.74 vox	0.585 vox
3	ja	28.0 %	2.45 vox	0.390 vox

Tabelle 8.8: Segmentierungsgüte der automatischen Segmentierung mit nach Abschnitt 8.5.2 geeigneter Parameterkonfiguration durch Modelle ohne und mit Torsionskräfte

8.5.3 Torsionskräfte

Alle Experimente in Abschnitt 8.5.2 wurden mit Modellen durchgeführt, die Torsionskräfte benutzen, da ansonsten im allgemeinen mit Kollapsen zu rechnen war. Diese Annahme wird in Tabelle 8.8 durch einen Vergleich der nach Abschnitt 8.5.2 parametrisierten Segmentierungsprozesse mit und ohne Torsionskräfte auf allen drei dort untersuchten Datensätzen deutlich belegt.

Dabei ist auf allen drei untersuchten Datensätzen eine extreme Verschlechterung der Segmentierungsqualität der Ansätze mit Modellen ohne Torsionskräfte gegenüber den Ansätzen mit Modellen mit Torsionskräften zu verzeichnen. Gerade in intensitätsschwachen Bereichen des LV kollabiert das Modell dabei komplett und zieht sich unter starker Formveränderung in intensitätsstarke Bereiche zurück. Eine korrekte Segmentierung dieser intensitätsschwachen Bereiche (gewöhnlich also der infarktbedingten Lücken im LV) findet nicht statt. Weiterhin kann beobachtet werden, daß sichtbare Winkelunterschiede in der Ausrichtung (z. B. die besagten 30 Grad in Datensatz 1), die das Modell durch eine Art Rotation ausgleichen sollte, ohne Torsionskräfte ebenfalls zu teilweisen Zusammenbrüchen der Modellstruktur führen.

Insgesamt kann hier also ohne die Verwendung der Torsionskräfte nicht mehr von einer wirklichen modellbasierten Segmentierung gesprochen werden, da ohne sie die Modellinformation praktisch kaum noch einen kontrollierbaren Einfluß auf die Segmentierung hat. Auch wenn bei den hier verwendeten, extra dünn vernetzten Modelle (siehe Abschnitt 5.2) dieses Verhalten sicher besonders deutlich zu Tage tritt, ist auch bei sehr stark vernetzten Modell nicht mit einem kompletten Verschwinden dieses Effektes zu rechnen.

8.5.4 Explizite Rotationsfähigkeit

In Abschnitt 8.5.2 wurden nur Parameterkonfigurationen untersucht, in denen das Modell in der Grobanpassungsphase (Phase 2) die Möglichkeit zur Rotation hatte. Daß diese Annahme sinnvoll ist, ergibt sich aus der Tatsache, daß sonst keine gute Anpassung an etwas anders als das Modell ausgerichtete LVs

Datensatz	Rot.-fähig	Falschseg.	Hausdorff.	Mittl. Abst.
1	nein	47.4 %	4.69 vox	0.685 vox
1	ja	28.6 %	2.24 vox	0.457 vox
2	nein	26.4 %	2.24 vox	0.404 vox
2	ja	25.7 %	1.73 vox	0.390 vox
3	nein	25.6 %	1.73 vox	0.342 vox
3	ja	28.0 %	2.45 vox	0.390 vox

Tabelle 8.9: Segmentierungsgüte der automatischen Segmentierung mit nach Abschnitt 8.5.2 geeigneter Parameterkonfiguration bzw. der bis auf die in Phase 2 und 3 deaktivierte Rotationsfähigkeit gleichen automatischen Segmentierung

vorgenommen werden kann, wie z. B. der LV in Datensatz 1 zeigt. Der Unterschied der Hauptachsen zwischen ihm und dem verwendeten Modell beträgt hier ungefähr 30 Grad.

Tabelle 8.9 untermauert diese These zudem durch einen Vergleich der Segmentierungsergebnisse des 3-phasigen Ansatz mit und ohne Rotationsfähigkeit auf den drei schon für die Bestimmung der Segmentierungsprozeßparameter untersuchten Datensätzen. Dabei finden die in Abschnitt 8.5.2 empfohlenen Parameterwerte Verwendung, wobei gegebenenfalls die Rotationsfähigkeit der Sensoren in Phase 2 und 3 deaktiviert ist.

Es ist unübersehbar, wie stark die Qualität der Segmentierungsergebnisse ohne Rotationsfähigkeit auf Datensatz 1 absinkt, dessen LV ja besagte Abweichung in seiner Ausrichtung zu der des Modells besitzt. Die Fähigkeit der erweiterten dynamischen Modelle zur Rotation ist also eine wichtige Eigenschaft für eine erfolgreiche Segmentierung von Objekten im Datensatz, deren Ausrichtung nicht genau der des verwendeten Modells entspricht. Ohne sie können nur Objekte durch Modelle, die Torsionskräfte verwenden, erfolgreich segmentiert werden, deren Ausrichtung der des verwendeten Modells entspricht, wie ein Blick auf die Segmentierungsergebnisse von Datensatz 2 zeigt, dessen LV ungefähr die Ausrichtung des benutzten Modells hat. Bei Datensatz 3 stimmen natürlich LV- und Modellausrichtung perfekt überein, so daß erstere nicht aus den Daten durch Anpassung ermittelt werden muß, sondern bereits gegeben vorliegt, was sich in einer entsprechend besseren Segmentierungsgüte des Ansatzes ohne Rotationsfähigkeit der Sensoren gegenüber dem mit dieser niederschlägt.

8.6 Segmentierung der Datensätze

Mit dem schon im vorigen Abschnitt verwendeten Modell, das aus der Handsegmentierung von Datensatz 3 automatisch generiert wurde, sollen nun die sieben Datensätze (1 bis 7), zu denen es einen Goldstandard gibt, automatisch segmentiert werden. Die Modell-, Simulations- und Segmentierungsprozeßparameter sind dabei die, die in den Tabellen A.1, A.2 und 8.6 aufgeführt, da sie

Eval.-Maß	Auto. Seg.	Seg. von A	Seg. von B
Dauer	25.3 sec	15.6 min	21.4 min
Falschseg.	29.4 %	36.5 %	30.6 %
Überseg.	15.3 %	32.3 %	25.8 %
Unterseg.	14.1 %	4.20 %	4.88 %
Tanimoto.	0.743	0.650	0.708
Hausdorff.	2.21 vox	2.81 vox	2.39 vox
mittl. Abst.	0.436 vox	0.667 vox	0.534 vox

Tabelle 8.10: durchschnittliche Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf allen Datensätzen mit Goldstandard

laut dem vorigen Abschnitt eine geeignete Wahl für diese Aufgabe darstellen.

Bis auf Datensatz 4, der intensitätsstarke Artefakte (ca. 15 mal intensiver als der LV) aufweist, kommt dabei immer die faltungsbasierte Mustererkennung in Phase 1 zum Einsatz. Lediglich in diesem Datensatz wird das Verfahren der Mustererkennung durch Vektorähnlichkeiten verwendet. Diese manuelle Auswahl des auf solche artefaktbelasteten Datensätze angepaßten, alternativen automatischen Verfahrens ist dabei die einzige aufgewendete Benutzerinteraktion, während der Segmentierung der besagten sieben Datensätze. Sie stellt jedoch keine Interaktion innerhalb der Verfahren da, da diese komplett autonom ablaufen können und das bei der Segmentierung dieser Datensätze auch tun. Abschließend ist zu den praktischen Testbedingungen noch zu sagen, daß außer den in Abschnitt 3.2 beschriebenen Normierungen keine Vorverarbeitungen auf den Datensätzen stattfanden.

Die Ergebnisse der automatischen Segmentierungen sind im Durchschnitt in Tabelle 8.10 und einzeln für alle Datensätze im Anhang B aufgeführt. Zur Einordnung dieser Ergebnisse sind jeweils auch die Ergebnisse der manuellen Segmentierungen zweier auf diesem Gebiet erfahrener und kompetenter Benutzer (A und B) auf den gleichen Datensätzen mit angegeben. Zusätzlich zu den im vorigen Abschnitt betrachteten Evaluationsmaßen, den Fehlermaßen in Bezug auf den jeweiligen Goldstandard, sind nun auch noch die relative Über- und Untersegmentierung in Prozent und der Tanimoto-Koeffizient mit verzeichnet.

Da das verwendete Modell auf Datensatz 3 erzeugt wurde und die verwendeten Parameter des automatischen Segmentierungsprozesses mittels Experimenten auf den Datensätzen 1 bis 3 optimiert wurden, können die Segmentierungsergebnisse auf diesen Datensätzen nicht als unabhängige Tests zur Bestätigung der Segmentierungsgüte mittels dieses automatischen Verfahrens dienen. Dazu sind lediglich die Ergebnisse auf den Datensätze 4 bis 7 geeignet, die in den Tabellen B.4 bis B.7 im Anhang B noch einmal gesondert aufgeführt sind.

Interessant ist, daß sich bei diesen unabhängigen Tests auf den Datensätzen 4 bis 7 die Ergebnisse der automatischen Segmentierung aber im selben Rahmen bewegen wie die Ergebnisse auf den Datensätzen 1 bis 3 (Tabellen B.1 bis B.3),

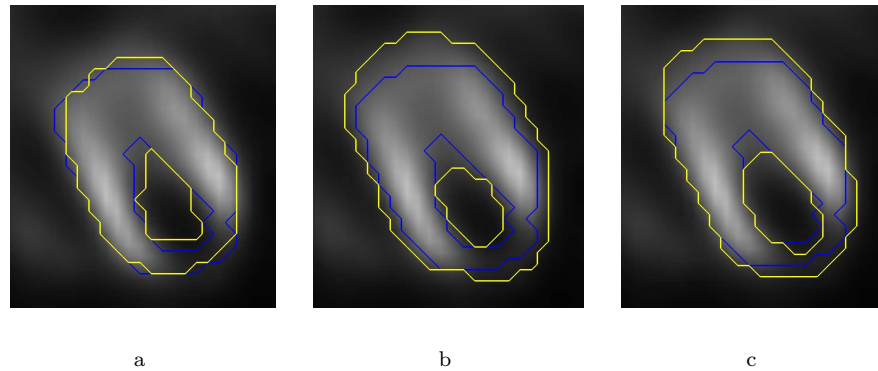


Abbildung 8.8: Ergebnisse (gelb) der verglichenen Segmentierungsprozesse in Schicht 11 des Datensatzes 3 im Vergleich zum Goldstandard (blau). **(a)** Automatische Segmentierung wie in dieser Arbeit entwickelt. **(b)** Manuelle Segmentierung von Benutzer A. **(c)** Manuelle Segmentierung von Benutzer B.

was zeigt, daß das Modell und das automatische Segmentierungsverfahren keine Überanpassung auf bestimmte Datensätze aufweisen, sondern auf allen die gleiche Fähigkeit zur Segmentierung besitzen.

Die automatische Segmentierung war auf allen Datensätzen erfolgreich. Der LV wurde jeweils gefunden und segmentiert, wobei infarktbedingt Lücken immer durch das Modell in geeigneter Weise ausgeglichen wurden (siehe für Beispiele die Abbildungen 8.8a und 8.9a). Die Zahlen in Tabelle 8.10 belegen diese Feststellung zudem auf eindrucksvolle Weise.

Ein Konturversatz wurde nicht benutzt (siehe Tabelle A.1) und war auf Grund der im Mittel ausgewogenen Über- und Untersegmentierung auch nicht vonnöten. Im Gegensatz dazu fällt die deutliche Übersegmentierung bei den beiden manuellen Vergleichsegmentierungen auf (siehe die Abbildungen 8.8b und 8.9b für Beispiele des Benutzers A und die Abbildungen 8.8c und 8.9c für Beispiele des Benutzers B), die auf etwas veränderte Modellannahmen bei den menschlichen Benutzern schließen lassen. Die Modellannahmen des automatischen Verfahrens entsprachen jedoch in dieser Hinsicht dem Goldstandard und hätten bei Bedarf durch den Konturversatz mühelos in die eine oder andere Richtung korrigiert werden können.

Ansonsten zeigt der Vergleich mit den manuellen Segmentierungen auf, daß die Qualität des automatischen Verfahrens absolut im Rahmen des Schwankens der Segmentierungsgüte zwischen durch von verschiedenen Personen durchgeführten Segmentierungen liegt, ja sogar besser ist. Zudem segmentiert es immer gleich in einer voraussehbaren und einstellbaren Art und Weise, während Personen, die ohnehin jeweils nach ihren persönlichen Modellvorstellungen segmentieren, immer (zumindest unterbewußten) Schwankungen unterliegen, selbst wenn sie sich aufeinander abstimmen.

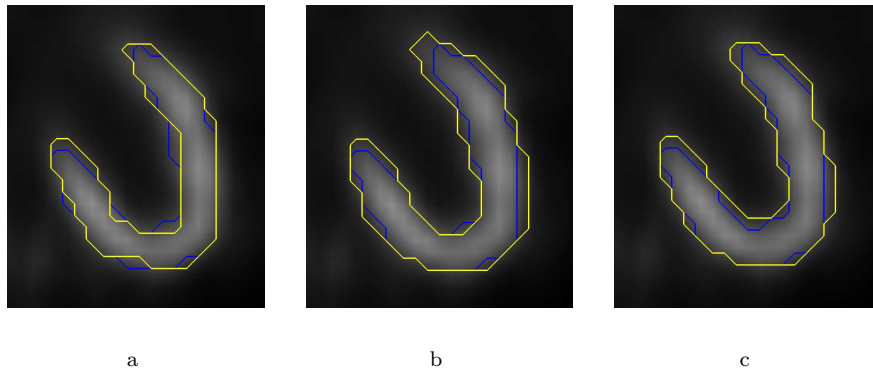


Abbildung 8.9: Ergebnisse (gelb) der verglichenen Segmentierungsprozesse in Schicht 22 des Datensatzes 7 im Vergleich zum Goldstandard (blau). **(a)** Automatische Segmentierung wie in dieser Arbeit entwickelt. **(b)** Manuelle Segmentierung von Benutzer A. **(c)** Manuelle Segmentierung von Benutzer B.

Bemerkenswert ist weiterhin die Segmentierungsdauer des automatischen Ansatzes, der im Vergleich zu den manuellen Vergleichsegmentierungen und zu den Zeiten vergleichbarer Segmentierungen unseres klinischen Partners (siehe Abschnitt 8.1) einen 30- bis fast 50-fachen Geschwindigkeitsvorteil bietet, bei dem keinerlei Interaktion während des automatischen Verfahrens notwendig ist.

Zum Abschluß noch ein Wort zu den restlichen 34 gegebenen Datensätzen, zu denen kein Goldstandard gegeben war und die zudem in anderer Ausrichtung als die sieben hier zuerst untersuchten vorlagen (siehe Abschnitt 8.1): Die Segmentierung wurde auf diesen unter sich gleich ausgerichteten Datensätzen mit einem Modell durchgeführt, das aus einer eigens auf einem dieser Datensätze manuell angefertigten Segmentierung automatisch erzeugt wurde. Das war erforderlich, da für keinen dieser Datensätze eine Handsegmentierung unseres klinischen Partners vorlag. Die automatische Segmentierung, mit diesem Modell und ansonsten unveränderten Einstellungen lieferte optisch gleichwertige Ergebnisse im Vergleich zu den automatischen Segmentierungen auf den Datensätzen 1 bis 7. Auf Grund des fehlenden Goldstandards war aber leider keine begründete quantitative Einschätzung wie bei den Datensätzen mit Goldstandard möglich.

8.7 Zusammenfassung

In diesem Kapitel wurden die Erkenntnisse und Entwicklungen der Theoriekapitel dieser Arbeit angewendet und in Zahlen und Bildern evaluiert. Dabei konnten dreidimensionale Feder-Masse-Modelle schnell und adäquat generiert werden. Weiterhin konnte die Wirksamkeit der Mustererkennung in Phase 1 des automatischen Segmentierungsprozesses zur geeigneten Startpunktfindung sowohl für normale als auch für Datensätze mit extrem intensitätsstarken Ar-

tefakten gezeigt werden. Zudem wurde eine für die gegebene Segmentierungsaufgabe geeignete Konfiguration der Parameter des Segmentierungsprozesses experimentell bestimmt.

Es wurde gezeigt, daß die Verwendung der Torsionskräfte essentiell für gute Segmentierungsergebnisse ist. Gleiches konnte für die explizite Rotationsfähigkeit des Modell bei Datensätzen mit vom verwendeten Modell deutlich abweichender LV-Ausrichtung festgestellt werden. Auch der positive Einfluß von richtungsgewichteten Kontursensoren konnte experimentell belegt werden. Letztendlich wurde auch die Tauglichkeit des in dieser Arbeit entwickelten, auf dem Modellgleichgewicht aufbauenden Stopkriteriums der Modellsimulation verifiziert.

Durch das in dieser Arbeit entwickelte Verfahren konnten alle gegebenen Datensätze erfolgreich automatisch segmentiert werden, das heißt, daß überall der LV sicher gefunden und segmentiert wurde, infarktbedingte Lücken in ihm modellgerecht überspannt wurden und keinerlei Benutzerinteraktion während des Verfahrens erforderlich war. Dabei segmentierte das automatische Verfahren in reproduzierbarer und konfigurierbarer Weise und war im Schnitt in Bezug auf einen durch Experten erstellten Goldstandard zahlenmäßig besser als manuelle Segmentierungen verschiedener erfahrener Benutzer. Damit lagen die Ergebnisse des hier entwickelten Segmentierungsverfahrens innerhalb der Varianz der Ergebnisse dieser menschlicher Experten. Darüber hinaus wurde die automatische Segmentierung um ein Vielfaches schneller durchgeführt als von jedem menschlichen Benutzer.

Kapitel 9

Zusammenfassung und Ausblick

9.1 Zusammenfassung

Durch das in dieser Arbeit entwickelte vollautomatische Segmentierungsverfahren konnte in allen vorliegenden humanen 3D-SPECT-Datensätzen der LV erfolgreich in anatomischer Hinsicht segmentiert werden. Das heißt, daß er im Datensatz jeweils korrekt gefunden und segmentiert wurde, wobei eventuelle Lücken in den funktionellen LV-Daten korrekt durch anatomisches Modellwissen ausgeglichen wurden.

Das in dieser Arbeit entwickelte Verfahren läuft vollautomatisch ab, so daß keine Benutzerinteraktion während des gesamten Segmentierungsprozesses erforderlich ist. Es segmentiert auf Grund seiner Natur den LV in einer reproduzierbaren, aber doch konfigurierbaren Weise. Die Qualität der von ihm erstellten Segmentierungen war im Durchschnitt besser als die manuell erstellter Segmentierungen von erfahrenen Benutzern. Vor allem jedoch war es um ein Vielfaches schneller beim Erstellen der Segmentierungen als alle menschlichen Benutzer.

Zur Segmentierung wurden dreidimensionale Modelle des LV verwendet, die vollautomatisch aus jeweils einer Beispielsegmentierung eines LV mittels eines ebenfalls in dieser Arbeit entwickelten Verfahrens erstmalig generiert werden konnten. Auf die durch dieses Verfahren zur allgemeinen Modellgenerierung erzeugten Modelle kann dabei mittels verschiedener Generierungsparameter (Sensordichte, Vernetzungsgrad, etc.) Einfluß genommen werden.

Zur Segmentierung selbst wurde eine neuartige, in dieser Arbeit erweiterte Form dynamischer Modelle eingesetzt. Diese um Torsionskräfte, explizite Rotationsfähigkeit und richtungsgewichtete Kontursensoren erweiterten 3D-Feder-Masse-Modelle sind die ersten expliziten dynamischen 3D-Formmodelle mit der Fähigkeit zur beliebig stabilen Formerhaltung bei gleichzeitiger hoher Anpassungsflexibilität. Sie sind daher ideal für die Segmentierung lückenhafter Daten geeignet, so wie z. B. hier für die funktionalen LV-Daten, da dort die Fähigkeit

zum modellbasierten Ausgleich dieser Lücken Voraussetzung für eine erfolgreiche Segmentierung ist.

Durch diese Arbeit wird damit zum ersten mal eine allgemeine automatische Segmentierung mit dynamischen Formmodellen auf dreidimensionalen Daten ermöglicht, denn sie können nun automatisch generiert, plaziert und angepaßt werden. Die dabei verwendete neue Klasse von Modellen schafft dafür die Voraussetzungen. Ihre Modelle sind stabil, trotzdem flexibel, effizient simulierbar, einfach kontrollierbar und zudem intuitiv verständlich in ihrem Verhalten.

9.2 Weiterführende Ergebnisse

Im Laufe dieser Arbeit wurde ein Prototyp erstellt, der die entwickelten theoretischen Verfahren implementiert und mit dem die Experimente in Kapitel 8 durchgeführt wurden. Diese Software wurde hinsichtlich Stabilität und Funktionalität so weit entwickelt, daß sie bereits zur alltäglichen LV-Segmentierung bzw. -Vorsegmentierung bei unserem klinischen Partner eingesetzt werden könnte.

Zudem ist mit dem Prototyp ein Programm entstanden, welches die entwickelten Verfahren effizient implementiert und vielfältige Darstellungs- und Steuerungsoptionen bietet, so daß es ebenfalls für Forschungs- und Lehrzwecke von Interesse ist. Dabei ist vor allem seine flüssig darstellbare Modellsimulation und die Fähigkeit zur Interaktion in Echtzeit interessant, die dem Benutzer ein lebendiges Gefühl für die Bedeutung der einzelnen Parameter und Auswirkungen seiner Interaktionen vermittelt.

9.3 Ausblick

Die in dieser Arbeit entwickelten theoretischen Grundlagen werfen eine Menge interessanter, weiterführender Fragestellungen auf. So ist für die Modellerzeugung nicht nur von Interesse, inwiefern die bei ihr verwendeten Parameter von einander abhängen und vielleicht zusammengefaßt werden könnten, sondern auch, wie das Modellwissen aus verschiedenen Beispielen der gleichen Klasse von Objekten in einem automatisch generierten Modell zusammengefaßt werden kann, so daß neben der grundlegenden Form der Klasse auch die Varianzen der Form der Objekte innerhalb der Klasse modelliert werden.

Weitere interessante Punkte ergeben sich direkt aus dem automatischen Segmentierungsverfahren. So ist es denkbar, die in Abschnitt 8.1 erwähnte halbautomatische Prozedur unseres klinischen Partners zur Vorsegmentierung vielleicht als datensatzspezifisches Vorwissen in den modellbasierten Segmentierungsprozeß einfließen zu lassen, um so die Qualität der automatischen Segmentierung gemessen am Goldstandard, also den Segmentierungen unseres klinischen Partners, zu erhöhen.

Generell ist es natürlich auch von Interesse, die automatische Modellerzeugung und die erweiterten dynamischen Modelle in der Segmentierung anderer dreidimensionaler Strukturen zu testen. Dabei könnten auf Grund des formsta-

bilien Charakters dieser Modelle besonders die modellgestützte Ergänzung nicht datengestützter Bereiche und die stabile Modellierung von Kontextregionen eines Objektes durch Teile des Modells, wie es z. B. durch das LV-Innere in dieser Arbeit der Fall war, eine sinnvolle Einsatzrichtung darstellen.

Anhang A

Parameter der Modellsimulation

Modellparameter	Wert
Sensormasse	1
Sensortorsionskonstante	1
Sensorkraftkonstante	1
Federkonstante	reziproke Ruhelänge
Federkraftwichtung	1
Torsionskraftwichtung	1
Intensitätssensorkraftwichtung	10
Kontursensorkraftwichtung	10
Konturversatz	0

Tabelle A.1: beim automatischen Segmentierungsprozeß verwendete Modellparameter

Simulationsparameter	Wert
Federkräfte	aktiv
Torsionskräfte	aktiv
Sensorkräfte	aktiv
Quarternionenrotationsstückelung	100
Simulationsschrittweite	0.05
Dämpfungsfaktor	0.02
Modellgleichgewichtsumgebung	0.01
Modellgleichgewichtssimulationsschritte	10
Datensatznormalisierung	aktiv
Dateninterpolation	trilinear
Gradientenoperatorgröße	$5 \times 5 \times 5$

Tabelle A.2: beim automatischen Segmentierungsprozeß verwendete Parameter der Simulationsumgebung

Anhang B

Ergebnistabellen der Segmentierungen

Eval.-Maß	Auto. Seg.	Seg. von A	Seg. von B
Dauer	21 sec	17 min	16 min
Falschseg.	28.6 %	37.2 %	25.8 %
Überseg.	14.7 %	32.7 %	15.8 %
Unterseg.	13.8 %	4.50 %	10.0 %
Tanimoto.	0.749	0.644	0.765
Hausdorff.	2.24 vox	3.00 vox	2.00 vox
mittl. Abst.	0.457 vox	0.700 vox	0.416 vox

Tabelle B.1: Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 1

Eval.-Maß	Auto. Seg.	Seg. von A	Seg. von B
Dauer	26 sec	16 min	21 min
Falschseg.	25.7 %	43.4 %	34.7 %
Überseg.	20.5 %	42.5 %	33.8 %
Unterseg.	5.16 %	0.824 %	0.847 %
Tanimoto.	0.756	0.570	0.656
Hausdorff.	1.73 vox	2.45 vox	2.00 vox
mittl. Abst.	0.390 vox	0.806 vox	0.605 vox

Tabelle B.2: Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 2

Eval.-Maß	Auto. Seg.	Seg. von A	Seg. von B
Dauer	17 sec	16 min	26 min
Falschseg.	28.0 %	35.0 %	27.0 %
Überseg.	9.04 %	33.8 %	22.8 %
Unterseg.	19.0 %	1.15 %	4.17 %
Tanimoto.	0.765	0.654	0.741
Hausdorff.	2.45 vox	3.00 vox	2.24 vox
mittl. Abst.	0.390 vox	0.674 vox	0.448 vox

Tabelle B.3: Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 3

Eval.-Maß	Auto. Seg.	Seg. von A	Seg. von B
Dauer	55 sec	19 min	25 min
Falschseg.	32.9 %	46.5 %	41.2 %
Überseg.	13.5 %	34.5 %	29.1 %
Unterseg.	19.4 %	12.0 %	12.1 %
Tanimoto.	0.725	0.585	0.632
Hausdorff.	2.24 vox	3.00 vox	2.45 vox
mittl. Abst.	0.502 vox	0.829 vox	0.720 vox

Tabelle B.4: Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 4

Eval.-Maß	Auto. Seg.	Seg. von A	Seg. von B
Dauer	18 sec	15 min	23 min
Falschseg.	42.2 %	31.3 %	24.1 %
Überseg.	22.8 %	26.5 %	20.7 %
Unterseg.	19.3 %	4.79 %	3.41 %
Tanimoto.	0.647	0.701	0.767
Hausdorff.	3.16 vox	2.24 vox	1.73 vox
mittl. Abst.	0.580 vox	0.541 vox	0.389 vox

Tabelle B.5: Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 5

Eval.-Maß	Auto. Seg.	Seg. von A	Seg. von B
Dauer	18 sec	16 min	20 min
Falschseg.	28.5 %	42.6 %	38.1 %
Überseg.	18.0 %	39.8 %	37.0 %
Unterseg.	10.6 %	2.82 %	1.12 %
Tanimoto.	0.742	0.586	0.623
Hausdorff.	2.24 vox	4.58 vox	4.58 vox
mittl. Abst.	0.416 vox	0.779 vox	0.715 vox

Tabelle B.6: Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 6

Eval.-Maß	Auto. Seg.	Seg. von A	Seg. von B
Dauer	22 sec	10 min	19 min
Falschseg.	20.2 %	19.7 %	23.6 %
Überseg.	8.60 %	16.4 %	21.1 %
Unterseg.	11.6 %	3.31 %	2.45 %
Tanimoto.	0.819	0.809	0.770
Hausdorff.	1.41 vox	1.41 vox	1.73 vox
mittl. Abst.	0.319 vox	0.340 vox	0.441 vox

Tabelle B.7: Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 7

Literaturverzeichnis

- [AL-ZUBI 2002] AL-ZUBI, STEPHAN (2002). *Survey of Deformable Models*. Technischer Bericht, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg.
- [ALGORRI und SCHMITT 1995] ALGORRI, MARÍA-ELENA und F. SCHMITT (1995). *Deformable Models For Reconstructing Unstructured 3D Data*. In: *CVRMed*, Bd. 905 d. Reihe *Lecture Notes Series on Computing*, S. 420–426. Springer.
- [BARDINET et al. 1996] BARDINET, ERIC, L. D. COHEN und N. AYACHE (1996). *Tracking and Motion Analysis of the Left Ventricle with Deformable Superquadrics*. *Medical Image Analysis*, 1(2):129–149.
- [BARDINET et al. 1998] BARDINET, ERIC, L. D. COHEN und N. AYACHE (1998). *A Parametric Deformable Model To Fit Unstructured 3D Data*. *CVIU*, 71(1):39–54.
- [BENEDIX 2004] BENEDIX, PETER (2004). *Automatisierungsprozesse zur Parametrisierung und Generierung dynamischer Modelle*. Diplomarbeit, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg.
- [BERGNER 2003] BERGNER, STEVEN (2003). *Structural Deformable Models for Robust Object Recognition*. Diplomarbeit, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg.
- [BERGNER et al. 2004] BERGNER, STEVEN, S. AL-ZUBI und K. TÖNNIES (2004). *Deformable Structural Models*. In: *ICIP*.
- [COHEN 1991] COHEN, LAURENT D. (1991). *On Active Contour Models and Ballons*. *CVGIP: Image Understanding*, 53(2):211–218.
- [COOTES et al. 1995] COOTES, T. F., C. J. TAYLOR, D. H. COOPER und J. GRAHAM (1995). *Active Shape Models - Their Training and Application*. *CVIU*, 61(1):38–59.
- [COOTES et al. 1998] COOTES, TIMOTHY F., G. J. EDWARDS und C. J. TAYLOR (1998). *Active Appearance Models*. In: *ECCV*, S. 484–498.

- [DELINGETTE 1994a] DELINGETTE, HERVE (1994a). *Adaptive and Deformable Models based on Simplex Meshes*. In: *IEEE Workshop of Non-Rigid and Articulated Objects*.
- [DELINGETTE 1994b] DELINGETTE, HERVE (1994b). *Simplex Meshes: a General Representation for 3D Shape Reconstruction*. In: *CVPR*, S. 856–857.
- [DELINGETTE 1998] DELINGETTE, HERVE (1998). *Initialization of Deformable Models from 3D Data*. In: *ICCV*, S. 311–316.
- [ENGEL 2004] ENGEL, KARIN (2004). *Ein integriertes strukturell-morphologisches Modell zur Erkennung des Heschl'schen Gyrius in planaren Kortextkarten*. Diplomarbeit, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg.
- [FERNANDEZ-MALOIGNE et al. 2000] FERNANDEZ-MALOIGNE, CHRISTINE, R. H. RAKOTOBÉ, F. LANGEVIN und M. FAUCHET (2000). *3D Segmentation and Visualization of Cardiac SPECT Studies*. In: *28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*, Bd. 3905 d. Reihe *Proceedings of SPIE*, S. 222–231.
- [FOLEY et al. 1996] FOLEY, JAMES D., A. VAN DAM, S. K. FEINER und J. F. HUGHES (1996). *Computer Graphics - Principles and Practice*. Addison-Wesley.
- [GARLAND 1999] GARLAND, MICHAEL (1999). *Quadric-Based Polygonal Surface Simplification*. Doktorarbeit, School of Computer Science, Carnegie Mellon University.
- [GARLAND und HECKBERT 1997] GARLAND, MICHAEL und P. S. HECKBERT (1997). *Surface Simplification Using Quadric Error Metrics*. In: *SIGGRAPH*, S. 209–216.
- [GONZALEZ und WOODS 1992] GONZALEZ, RAFAEL C. und R. E. WOODS (1992). *Digital Image Processing*. Addison-Wesley.
- [HAMARNEH et al. 2001] HAMARNEH, GHASSAN, T. MCINERNEY und D. TERZOPOULOS (2001). *Deformable Organisms for Automatic Medical Image Analysis*. In: *MICCAI*, S. 66–76.
- [HEIMANN et al. 2004] HEIMANN, TOBIAS, M. THORN, T. KUNERT und H.-P. MEINZER (2004). *Empirische Vergleichsmaße für die Evaluation von Segmentierungsergebnissen*. In: *Bildverarbeitung für die Medizin*.
- [JAIN et al. 1998] JAIN, ANIL K., Y. ZHONG und M.-P. DUBUISSON-JOLLY (1998). *Deformable Template Models: A Review*. *Signal Processing*, 71:109–129.
- [JÄHNE 2002] JÄHNE, BERND (2002). *Digitale Bildverarbeitung*. Springer.

- [JÄHNE et al. 1999] JÄHNE, BERND, H. SCHARR und S. KÖRKELE (1999). *Signal Processing and Pattern Recognition*, Bd. 2 d. Reihe *Handbook of Computer Vision and Applications*, Kap. Principles of Filter Design, S. 125–151. Academic Press.
- [KASS et al. 1988] KASS, M., A. WITKIN und D. TERZOPOULOS (1988). *Snakes: Active Contour Models*. IJCV, 1(4):321–331.
- [KEREN et al. 1992] KEREN, DANIEL, J. SUBRAHMONIA und D. B. COOPER (1992). *Robust Object Recognition Based on Implicit Algebraic Curves and Surfaces*. In: *ICPR*, S. 791–794.
- [KOHONEN 1987] KOHONEN, TEUVO (1987). *Self-Organization and Assoziative Memory*. Springer.
- [KONRAD-VERSE 2004] KONRAD-VERSE, OLAF (2004). *Interactive Refinement of a Ballon Segmentation on Medical Data*. Diplomarbeit, Fakultät für Informatik, Universität Bremen.
- [KRATZER 1990] KRATZER, KLAUS PETER (1990). *Neuronale Netze*. Hanser.
- [KUIPERS 2002] KUIPERS, JACK B. (2002). *Quarternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press.
- [LORENSEN und CLINE 1987] LORENSEN, WILLIAM E. und H. E. CLINE (1987). *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*. In: *SIGGRAPH*, Bd. 21, S. 163–169.
- [MCINERNEY und TERZOPOULOS 1993] MCINERNEY, TIM und D. TERZOPOULOS (1993). *A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking*. In: *ICCV*, S. 518–523.
- [OTTMANN und WIDMAYER 1996] OTTMANN, THOMAS und P. WIDMAYER (1996). *Algorithmen und Datenstrukturen*. Spektrum Akademischer Verlag, Heidelberg.
- [POHLE 2004] POHLE, REGINA (2004). *Computerunterstützte Bildanalyse zur Auswertung medizinischer Bilddaten*. Habilitationsschrift, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg.
- [POHLE et al. 2004] POHLE, REGINA, M. WEGNER, K. RINK, K. TÖNNIES, A. CELLER und S. BLINDER (2004). *Segmentation of the Left Ventricle in 4D-dSPECT Data using Free Form Deformation of Super Quadrics*. In: *Medical Imaging: Image Processing*, Bd. 5370 d. Reihe *Proceedings of SPIE*, S. 1388–1394.
- [PRUSINKIEWICZ und LINDENMAYER 1990] PRUSINKIEWICZ, P. und A. LINDENMAYER (1990). *The Algorithmic Beauty of Plants*. Springer.

- [SCLAROFF und ISIDORO 1998] SCLAROFF, STAN und J. ISIDORO (1998). *Active Blobs*. In: *ICCV*, S. 1146–1153.
- [SIDDIQI und KIMIA 1996] SIDDIQI, KALEEM und B. B. KIMIA (1996). *A Shock Grammar for Recognition*. In: *CVPR*, S. 507–513.
- [TERZOPOULOS und FLEISCHER 1988] TERZOPOULOS, DEMETRI und K. W. FLEISCHER (1988). *Deformable Models*. *The Visual Computer*, 4(6):306–331.
- [TERZOPOULOS und VASILESCU 1991] TERZOPOULOS, DEMETRI und M. VASILESCU (1991). *Sampling and Reconstruction with Adaptive Meshes*. In: *CVPR*, S. 70–75.
- [WEGNER 2003] WEGNER, MELANIE (2003). *Segmentierung von Leber und linkem Herzventrikel in 4D-SPECT-Daten*. Diplomarbeit, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg.

Tabellenverzeichnis

8.1	günstige Parameter zur automatischen Generierung eines dynamischen Modells auf den gegebenen Daten	64
8.2	Anzahl der einzelnen Elemente des erzeugten dynamischen Modells	65
8.3	Anzahl und mittlerer Vernetzungsgrad der verschiedenen Sensortypen des erzeugten dynamischen Modells	65
8.4	Richtigkeit der Startpunktbestimmung mittels Mustererkennung durch Faltung (Faltung) bzw. Vektorähnlichkeiten (Vektor) . .	67
8.5	Gegenüberstellung der Auswirkungen aller Optionen jeweils eines Parameters des automatischen Segmentierungsprozesses auf die über alle untersuchten Datensätze (1 bis 3) gemittelte Segmentierungsgüte in Bezug auf den Goldstandard	70
8.6	geeignete Parameterkonfiguration des automatischen Segmentierungsprozesses	72
8.7	Bestwerte der einzelnen Evaluierungsmaße der automatischen Segmentierungsprozesse aller Parameterkonfigurationen auf allen untersuchten Datensätzen (1 - 3) gruppiert nach 2- bzw. 3-Phasen-Ansätzen	73
8.8	Segmentierungsgüte der automatischen Segmentierung mit nach Abschnitt 8.5.2 geeigneter Parameterkonfiguration durch Modelle ohne und mit Torsionskräfte	74
8.9	Segmentierungsgüte der automatischen Segmentierung mit nach Abschnitt 8.5.2 geeigneter Parameterkonfiguration bzw. der bis auf die in Phase 2 und 3 deaktivierte Rotationsfähigkeit gleichen automatischen Segmentierung	75
8.10	durchschnittliche Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf allen Datensätzen mit Goldstandard	76
A.1	beim automatischen Segmentierungsprozeß verwendete Modellparameter	86
A.2	beim automatischen Segmentierungsprozeß verwendete Parameter der Simulationsumgebung	86

B.1	Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 1	88
B.2	Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 2	88
B.3	Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 3	88
B.4	Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 4	89
B.5	Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 5	89
B.6	Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 6	89
B.7	Segmentierungsergebnisse des automatischen Segmentierungsprozesses und der manuellen Segmentierung durch zwei erfahrene Benutzer (A und B) auf Datensatz 7	90

Abbildungsverzeichnis

1.1	Schichtbild einer 3D-SPECT-Aufnahme eines menschlichen Brustkorbes. Die helle Struktur ist der linke Ventrikel.	2
1.2	SPECT-Schichtbildaufnahme des linken Ventrikels, mit einer deutlich sichtbaren infarktbedingten Lücke in der hellen LV-Wand (Pfeil). In Bild (b) ist eine manuell erstellte Segmentierung dieses Organs zu sehen.	2
4.1	Über den Gradientenbetrag erzeugte Konturbilder eines Schichtbildes. Bild (b) wurde zusätzlich einer Richtungsgewichtung in X-Richtung (Pfeil) unterzogen. Die hellen Bereiche sind jeweils die unscharfen Konturen des LV.	27
4.2	LV mit zugehöriger Handsegmentierung unseres klinischen Partners. Es ist deutlich zu sehen, daß etwas weiter als bis zur idealen Kontur segmentiert wurde.	28
5.1	Prozeß des Markierens des inneren LV-Bereiches einer gegebenen Handsegmentierung. (a) ist die ursprüngliche Segmentierung. (b) zeigt die Differenz (grau) zur konvexen Hülle des LV. (c) zeigt das Ergebnis nach Entfernung der Artefakte (Pfeil)	30
6.1	Aus dem Modell generiertes Muster (a) und Ergebnis der faltungsbasierten Erkennung dieses Musters auf einem Datensatz (b) . Helle Bereiche stehen dort für hohe Ähnlichkeitswerte des Musters.	38
6.2	(a) zeigt ein Schnittbild in Y-Richtung des Ergebnisses der faltungsbasierten Mustererkennung eines LV-Musters. (b) zeigt dieselbe Ansicht des Ergebnisses der Mustererkennung mittels Vektorähnlichkeiten desselben Musters. Helle Bereiche stehen jeweils für hohe Ähnlichkeitswerte des Musters. Deutlich ist zu sehen, daß in (a) ein Artefakt (roter Pfeil) bedeutend heller ist als die Mitte des LV (gelber Pfeil), es in (b) jedoch keine Rolle spielt. Dort ist die LV-Mitte am hellsten ist.	40
6.3	die Kontur eines Modells (gelb) und seine daraus erzeugte voxelbasierte Repräsentation (blau)	45

7.1	Verschiedene mögliche Darstellungen eines 3D-SPECT-Datensatzes im Prototypen. (a) Schichtbild. (b) Volume Rendering. (c) Iso-Oberfläche. (d) Verschiedene Schnittebenen in der 3D-Ansicht.	51
7.2	Verschiedene mögliche Darstellungen eines 3D-Modells im Prototypen. (a) 2D-Schnitt in Höhe der angezeigten Schicht des Datensatzes durch die Modellstruktur mit IDs der geschnittenen Federn. (b) 2D-Projektion des Modells innerhalb eines Toleranzbereichs von 2 Voxeln um die angezeigte Schicht des Datensatzes mit IDs der Sensoren. (c) 3D-Ansicht der Modellkontur mit Sensor-IDs. (d) 3D-Ansicht der gesamten Modellstruktur.	52
8.1	SPECT-Schichtbildaufnahme des linken Ventrikels, mit einer deutlich sichtbaren infarktbedingten Lücke in der hellen LV-Wand (Pfeil). In Bild (b) ist eine halbautomatisch erstellte, schwellwertbasierte Vorsegmentierung dieses Organs zu sehen. . .	56
8.2	Anpassung eines LV-Modells an den LV in 2D- und 3D-Ansicht. (a) , (c) Ohne Verwendung von Torsionskräften. (b) , (d) Mit der Verwendung von Torsionskräften.	59
8.3	Anpassung eines LV-Modells an einen LV abweichender Ausrichtung unter Verwendung von Torsionskräften. (a) Ohne explizite Rotationsfähigkeit des Modells. (b) Mit expliziter Rotationsfähigkeit des Modells.	60
8.4	Anpassungsprozeß eines LV-Modells an einen LV aus sehr schlechter, weil ferner Startposition heraus. Torsionskräfte, explizite Rotationsfähigkeit und richtungsgewichtete Kontursensoren wurden benutzt. Bild (a) bis (c) zeigen, wie das Modell unter Verformung in Richtung des LV gezogen wird. Bild (d) bis (f) zeigen den Rest der Anpassung, bei dem sich das Modell mittels eines durchgängigen Kontaktes zur intensitätsstarken LV-Wand richtig auf dem LV ausrichtet.	61
8.5	Einfluß der Richtungsfilterung der Kontursensoren auf die LV-Modellanpassung. (a) Ohne Richtungsfilterung. (b) Mit Richtungsfilterung. Die Anpassung wurde zur besseren Sichtbarkeit des Effektes dabei jeweils mit 5-fach stärker gewichteten Kontursensorkräften durchgeführt.	62
8.6	Einfluß eines Konturversatzes auf die LV-Modellanpassung. (a) Negativer Konturversatz. (b) Kein Konturversatz. (c) Positiver Konturversatz.	63
8.7	3D-Ansichten des mittels der Handsegmentierung zu Datensatz 3 generierten Modells	65
8.8	Ergebnisse (gelb) der verglichenen Segmentierungsprozesse in Schicht 11 des Datensatzes 3 im Vergleich zum Goldstandard (blau). (a) Automatische Segmentierung wie in dieser Arbeit entwickelt. (b) Manuelle Segmentierung von Benutzer A. (c) Manuelle Segmentierung von Benutzer B.	77

8.9 Ergebnisse (gelb) der verglichenen Segmentierungsprozesse in Schicht 22 des Datensatzes 7 im Vergleich zum Goldstandard (blau). **(a)** Automatische Segmentierung wie in dieser Arbeit entwickelt. **(b)** Manuelle Segmentierung von Benutzer A. **(c)** Manuelle Segmentierung von Benutzer B. 78

Selbständigkeitserklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbstständig und nur unter Zuhilfenahme der angegebenen Quellen erstellt habe.

Lars Dornheim

Magdeburg, den 28. Januar 2005