



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

**Evaluierung von 3D
Segmentierungsapplikationen sowie
Entwicklung und Implementierung
einer Segmentierungsmethode für
den Voxel-Man Chirurgie-Simulator**

Teresa Schiffner

(Fachrichtung Medizintechnik)

18. März 2014

Erstgutachter: Prof. Dr. -Ing Thomas Schiemann

Zweitgutachter: Dr. rer. nat. Ulf Tiede

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Masterarbeit unterstützt und motiviert haben.

Zunächst möchte ich mich bei meinem Betreuer und Professor, Herrn Prof. Thomas Schiemann bedanken, der mich auf die Arbeitsgruppe Voxel-Man aufmerksam gemacht und mir somit die Möglichkeit gegeben hat, die Masterarbeit in meinen Interessenbereich schreiben zu können. Darüber hinaus wäre ohne seine interessanten und anschaulichen Vorlesungen wohl nicht mein besonderes Interesse an der Informatik geweckt worden.

Weiterhin möchte ich mich bei meinen Kollegen der Arbeitsgruppe Voxel-Man für die interessante und lehrreiche Zeit bedanken. Besonderer Dank gilt meinem Betreuer, Herrn Dr. Ulf Tiede, der mir mit seinem fachlichen Rat und zahlreichen Anregungen stets zur Seite stand. Herrn Bastian Dittmar möchte ich explizit für die algorithmische Unterstützung, die zahlreichen Diskussionen und Anregungen danken. Seine Unterstützung war nicht nur während der Einarbeitung, sondern während der gesamten Zeit eine große Hilfe. Weiterhin danke ich Herrn Prof. Dr. Karl Heinz Höhne und Herrn Dr. Andreas Pommert für die konstruktiven strukturellen Ratschläge.

Zu guter Letzt möchte ich mich bei meinen Eltern und meinem Bruder bedanken, die mich während meines gesamten Studiums stets unterstützt und motiviert haben. Ich danke ihnen zudem für die zahlreichen Stunden des Korrekturlesens meiner Arbeit.

Zusammenfassung

Der im Jahr 2005 am Markt eingeführte Voxel-Man Chirurgie-Simulator dient als Trainingssystem angehender Chirurgen während der Ausbildung. Zudem ermöglicht er die Vorbereitung auf eine Operation eines individuellen Patienten. Um aus den individuellen Daten einen virtuellen Patienten zu erstellen, ist zunächst eine Segmentierung der Daten erforderlich. Dies war bisher nur grob mit einem globalen Schwellwert möglich.

Das Ziel der Arbeit war es daher, eine geeignete Segmentierungsapplikation zu finden und gegebenenfalls ein eigenes Verfahren zu entwickeln und in das Voxel-Man System zu integrieren. Um die für den Simulator „Voxel-Man Tempo“ relevanten Risikostrukturen des Innenohrs segmentieren zu können, lag der Fokus der Arbeit auf der Segmentierung kleiner Strukturen.

Zunächst werden verschiedene Segmentierungsapplikationen im Hinblick auf die Eignung für den Simulator untersucht. Dabei waren eine schwellwertbasierte Segmentierung kleiner Risikostrukturen sowie eine intuitive Bedienung von besonderer Wichtigkeit.

Keine der untersuchten Applikationen erfüllte alle Anforderungen, weshalb ein eigenes Segmentierungsverfahren entwickelt und in das Voxel-Man System integriert wurde. Das Verfahren arbeitet mit einem Region Growing auf Schwellwertbasis. Durch Setzen des Startpunktes für das Region Growing wird der Schwellwertbereich bestimmt. Auch eine Schwellwertanpassung ist möglich, indem weitere Regionen des Datensatzes ausgewählt werden, an deren Intensitäten sich der Schwellwertbereich anpasst.

Für die Trennung von Objekten, deren Schwellwertbereiche sich überschneiden, wurde eine Methode entwickelt, die nach Auswahl einer ungewünschten Region automatisch eine Trennung an der dünnsten Verbindung herbeiführt. Zudem ist eine Segmentierung über Teilobjekte möglich, um für jede Region einen optimalen Schwellwertbereich festzulegen und somit eine möglichst glatte Oberfläche zu erzielen.

Das im Rahmen dieser Arbeit entwickelte Verfahren bietet eine interaktive und intuitive Bedienung. Durch die semi-automatische Trennung, die bildverarbeitende Algorithmen vor dem Benutzer verbirgt, kann die Segmentierung ohne aufwendige Einarbeitung und mit geringem Zeitaufwand erfolgen. Die Segmentierung aller Risikostrukturen ist jeweils in wenigen Minuten möglich und führt zu glatten Oberflächen in der 3D Ansicht.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ziel der Arbeit	3
1.2	Aufbau der Arbeit	4
2	Segmentierung	7
2.1	Pixelorientierte Verfahren	7
2.2	Regionenorientierte Verfahren	9
2.3	Kantenbasierte Verfahren	11
2.4	Modellbasierte Verfahren	12
3	Der Voxel-Man Chirurgie-Simulator	13
3.1	Manuelle Segmentierung	14
3.2	3D-Visualisierung	17
4	Segmentierungsapplikationen	21
4.1	MeVisLab	21
4.1.1	Material und Methoden	22
4.1.2	Ergebnisse	29
4.2	3D Slicer	31
4.2.1	Material und Methoden	31
4.2.2	Ergebnisse	36
4.3	ITK Snap	37
4.3.1	Material und Methoden	37
4.3.2	Ergebnisse	41
4.4	MIPAV	43
4.4.1	Material und Methoden	43
4.4.2	Ergebnisse	47
4.5	Turtle Seg	47
4.5.1	Material und Methoden	48
4.5.2	Ergebnisse	50

4.5.3	Weitere Segmentierungsapplikationen	51
4.6	Schlussfolgerung	52
5	Eigener Ansatz für die Segmentierung kleiner Risikostrukturen	53
5.1	Lösungsansatz	53
5.2	Beschreibung der implementierten Methoden	54
5.2.1	Setzen eines Startpunktes für das Region Growing	54
5.2.2	Grow	55
5.2.3	Shrink	55
5.2.4	Manuelle Trennung	57
5.3	Beschreibung der neu entwickelten Methoden	57
5.3.1	Split-Methode	58
5.3.2	Objektmanagement	62
5.3.3	Manuelle Schwellwertanpassung	63
6	Ergebnisse	65
6.1	Segmentierung der Risikostrukturen des Innenohrs	65
7	Schlussfolgerungen	75
7.1	Fazit	75
7.2	Ausblick	76
	Literatur	79

1 Einleitung

Die Segmentierung medizinischer Bilddaten ist ein sehr wichtiges und gleichzeitig problematisches Thema in der medizinischen Bildverarbeitung. Sie bildet die Grundlage für die weitergehende Analyse, Vermessung und 3D-Visualisierung der Daten. Soll beispielsweise das Volumen eines Tumors ermittelt werden, ist zunächst eine Segmentierung der Tumorstruktur nötig. Das bedeutet, dass alle zur Tumorstruktur gehörenden Elemente ermittelt und gekennzeichnet werden müssen.

Der Mensch ist mit einem visuellen System ausgestattet, welches ihm ermöglicht, Strukturen voneinander zu unterscheiden, obwohl sie überlappen und sich von ihren Intensitätswerten nur geringfügig unterscheiden oder aber nur teilweise umrissen sind. In der maschinellen Bildgebung hingegen ist es bis heute nicht gelungen, ein solches System nachzubilden, da sich die Bedingungen und Parameter für eine allgemeingültige Segmentierung nicht formulieren lassen. Dies ist durch teils stark schwankende Bildqualität sowie Rauscheinflüsse und Bildartefakte bedingt. Zwar wurden im Laufe der Jahre verschiedene Verfahren zur manuellen Segmentierung entwickelt, diese sind jedoch nur eingeschränkt einsetzbar. Häufig sind manuelle Verfahren mit einem hohen Zeitaufwand verbunden, da verschiedene Algorithmen durchgeführt werden müssen, um ein Objekt zu segmentieren. Durch Anpassung der Bilddaten an einen Atlas (Registrierung) kann der Zeitaufwand erheblich verringert werden. Medizinische Bilddaten unterscheiden sich jedoch in vielerlei Hinsicht. Gründe dafür sind beispielsweise, dass die Daten an unterschiedlichen Orten von unterschiedlichen Ärzten mit unterschiedlichen Geräten aufgenommen wurden. Vor allem ist aber die Lage menschlicher Organe sehr unterschiedlich. Die Lage des Herzens im menschlichen Thorax ist beispielsweise keineswegs eindeutig. Es kann je nach Patient um seine Längsachse gedreht oder vertikal gestaucht sein. All diese Gründe führen dazu, dass es in der medizinischen Bildgebung extrem schwierig ist, Standardatlanten zu erstellen [Maintz und Viergever, 1998].

Das Institut für Medizinische Informatik - heute die Arbeitsgruppe Voxel-Man -

am Universitätsklinikum Hamburg-Eppendorf begann im Jahre 2000 unter der Leitung von Prof. Dr. Höhne mit der Entwicklung eines Chirurgie-Simulators. Dieser computergestützte Simulator dient vor allem der Ausbildung angehender Chirurgen, denen während ihrer Ausbildung eine Operation an einem virtuellen Patienten ermöglicht wird. Der Chirurg arbeitet mit virtuellen Instrumenten, die über Kraftrückkopplungsgeräte ein haptisches Gefühl vermitteln.

Die erste Linux-basierte Generation des Simulators „Voxel-Man Tempo“ dient dem Trainieren der chirurgischen Eröffnung des Felsenbeins als Zugang zum Innenohr und ist schon seit 2005 auf dem Markt. In der zweiten Generation (Abbildung 1.1) wurde das System auf Windows umgestellt und beinhaltet zusätzlich ein Modul, um Blutungen zu simulieren und ein Modul für den Dentalbereich. Auch für die Nasennebenhöhlenchirurgie und die Arthroskopie sind Module in der Entwicklung.

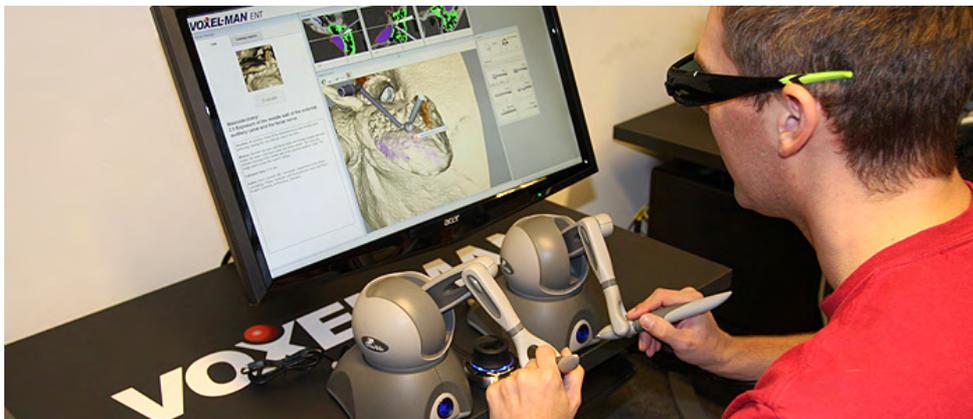


Abbildung 1.1: Eröffnung des Felsenbeins an dem Voxel-Man Simulator

Um aus CT-Daten einen virtuellen Patienten erstellen zu können, wurde ein Segmentierungsverfahren entwickelt, welches die Anforderungen der Voxel-Man Visualisierung erfüllt, sowie eine interaktive und schnelle Segmentierung ermöglicht, die ohne Verfälschung der Originalbilddaten einhergeht. Die interaktive Bedienung zusammen mit einer 3D-Visualisierung ermöglicht die Erzeugung glatter Oberflächen, was für den Simulator von besonderer Bedeutung ist. Dieses Verfahren wurde allerdings für die erste Generation entwickelt und ist somit im Gegensatz zum aktuellen Windows-basierten Voxel-Man Simulator auf Linux-Basis geschrieben. Die Segmentierung erfolgt über einen globalen Schwellwert. Das bedeutet, es wird ein Intensitätsbereich festgelegt, der auf das gesamte Volumen angewandt wird. Zudem kommen form- gestalt- und strukturverändernde

Algorithmen - sogenannte morphologische Operatoren - zum Einsatz, welche in Kapitel 3.1 noch näher erläutert werden. Diese stellen eine Hürde für einen Anwender dar, der nicht über Kenntnisse der Bildverarbeitung verfügt. Zudem ist die Segmentierung kleiner Strukturen sehr zeitaufwendig, da keine lokalen Operatoren vorhanden sind. Die aktuelle Windows-basierte Version beinhaltet ausschließlich die Möglichkeit einer Segmentierung über einen globalen Schwellwert. Es sind keine weiteren Operatoren vorhanden, weshalb dieser Segmentierer aktuell lediglich für die Segmentierung von Hartgewebe verwendet wird. Eine Verbesserung zum Linux-basierten Segmentierer bietet die 3-dimensionale Visualisierung in Echtzeit. Zu jedem Zeitpunkt einer Schwellwertänderung wird die 3D-Visualisierung aktualisiert, was das Einstellen eines optimalen Schwellwerts erheblich erleichtert.

1.1 Ziel der Arbeit

Ziel der Arbeit ist es, verschiedene Segmentierungsapplikationen hinsichtlich ihrer Bedienbarkeit und ihrer Eignung für den Voxel-Man Chirurgie-Simulator zu untersuchen. Die Software soll die Anforderungen an die Visualisierung des Voxel-Man Chirurgie-Simulators erfüllen, sowie eine intuitive und interaktive Bedienung gewährleisten. Dabei sollen die folgenden Punkte als Anforderungen gelten:

- Das Segmentierungsverfahren sollte schwellwertbasiert sein, da die Oberflächendarstellung im Simulator auf homogenen Intensitätsgebieten beruht.
- Die Software sollte eine intuitive Methode für einen Benutzer darstellen, der nicht mit den Algorithmen der Bildverarbeitung vertraut ist. Kenntnisse über Segmentierungsalgorithmen sollten somit keine Voraussetzung für die Benutzung der Software sein.
- Die 3-dimensionale Ansicht während des Segmentierungsprozesses sollte in Echtzeit erfolgen, um mit einem geringen Zeitaufwand eine möglichst glatte Oberfläche zu erzielen.
- Es sollte sich um frei verfügbare Software handeln, da anderenfalls mögliche Lizenzkosten eine Verwendung im Simulator verhindern.
- Die Software muss stabil laufen. Ständige Programmabstürze gelten als Ausschlusskriterium.

- Eine Segmentierung kleiner Strukturen sollte möglich sein. Der Fokus wurde dabei auf die Risikostrukturen des Innenohrs (Cochlea, Bogengänge, Gehörknöchelchen und Gesichtsnerv (Abbildung 1.2)) gelegt.

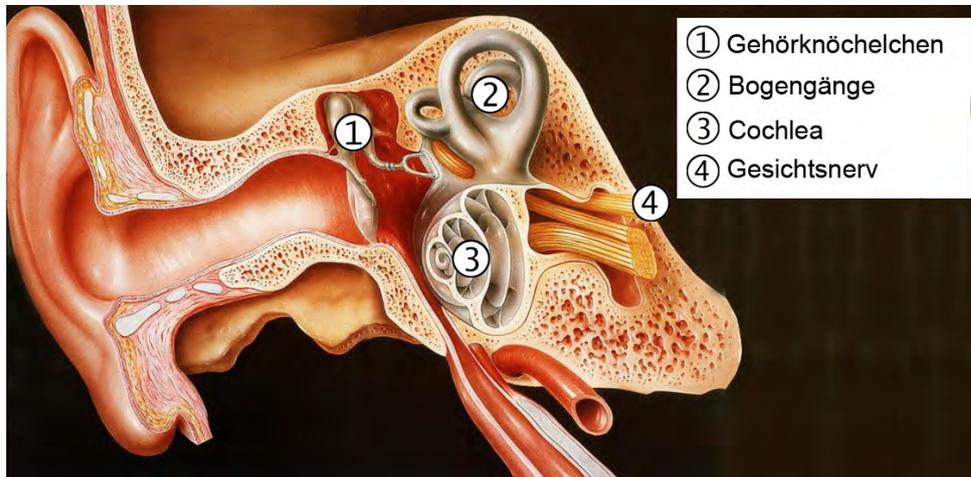


Abbildung 1.2: Das menschliche Innenohr mit den für den Simulator „Voxel-Man Tempo“ relevanten Risikostrukturen [Regensburg, 2014]

Sollte keine geeignete Software auf dem freien Markt verfügbar sein, die diese Anforderungen erfüllt, soll ein eigenes Verfahren entwickelt und in das System integriert werden.

1.2 Aufbau der Arbeit

Im folgenden Kapitel werden zunächst Grundlagen erläutert, die zum Verständnis der Arbeit nötig sind. Der Begriff der Segmentierung wird erklärt und es wird ein Einblick in die verschiedenen Segmentierungsverfahren und Algorithmen gegeben.

Um ein Verständnis für die speziellen Anforderungen des Voxel-Man Chirurgie-Simulators an das Segmentierungsverfahren zu vermitteln, befasst sich Kapitel 3 mit dem Visualisierungs- und Linux-basierten Segmentierungsverfahren des Simulators.

Anschließend werden in Kapitel 4 die untersuchten Segmentierungsapplikationen auf die in Kapitel 1.1 genannten Anforderungen hin untersucht und deren Ergebnisse präsentiert.

Kapitel 5 befasst sich mit der Entwicklung und Implementierung eines neuen Voxel-Man Segmentierers. Ein Lösungsansatz wird formuliert und die verwendeten Algorithmen werden erläutert.

Die Ergebnisse des implementierten Segmentierers werden in Kapitel 6 anhand eines Beispiels präsentiert.

Abschließend wird in Kapitel 7 ein Fazit gezogen, sowie mögliche Verbesserungen des implementierten Segmentierers angesprochen.

2 Segmentierung

In der Einleitung wurde bereits erwähnt, dass im Laufe der Jahre verschiedene Verfahren zur manuellen Segmentierung entwickelt wurden. In diesem Kapitel sollen nun die wichtigsten Verfahren beschrieben werden.

Eine maschinelle Segmentierung lässt sich durch eine Unterteilung der einzelnen Elemente in zusammenhängende Regionen realisieren. Bei den Elementen kann es sich dabei je nach Dimension sowohl um Bildelemente (Pixel), als auch um Volumenelemente (Voxel) handeln. Alle Elemente einer Region erhalten dasselbe Label, wodurch sie eindeutig einer Region zugeordnet sind. Um dies zu erreichen, wurden im Laufe der Jahre verschiedene Methoden entwickelt, die sich grob in 4 Kategorien einteilen lassen [Jähne, 2002]:

- Pixelorientierte Verfahren
- Regionalorientierte Verfahren
- Kantenbasierte Verfahren
- Modellbasierte Verfahren

2.1 Pixelorientierte Verfahren

Bei pixelorientierten Verfahren wird für jedes einzelne Element anhand eines Attributs untersucht und entschieden, ob es zu einem bestimmten Objekt gehört. Das Schwellwertverfahren ist das verbreitetste pixelorientierte Verfahren, welches an dieser Stelle näher erläutert werden soll. Das einfachste Schwellwertverfahren ist das des globalen Schwellwerts. Dabei wird durch einen unteren Schwellwert S_1 und einen oberen Schwellwert S_2 ein Intensitätsbereich festgelegt. Das Label $L(x, y, z)$ eines jeden Elements wird anhand seiner Intensität $I(x, y, z)$ wie folgt bestimmt:

$$L(x, y, z) = \begin{cases} l & \text{falls } I(x, y, z) \in [S_1, S_2] \\ 0 & \text{sonst} \end{cases},$$

wobei l für einen Labelwert steht, der das Objekt charakterisiert. Ein Element wird durch sein Label somit einer Region und einem Intensitätsbereich zugeordnet. Die Bestimmung der Schwellwerte kann sowohl automatisch - beispielsweise über das Histogramm - als auch manuell erfolgen. Dieses Verfahren lässt sich auf weitere Schwellwertbereiche ausweiten, denen jeweils ein Label zugeordnet wird.

Auch eine Überlappung der Schwellwertbereiche ist möglich, welche in Kombination mit einer ungleichmäßigen Beleuchtung bzw. Rauscheinflüssen bei CT-Daten zu Problemen führen kann. Ein Beispiel dazu ist in Abbildung 2.1 zu sehen. Die linke Abbildung zeigt das Originalbild und die rechte Abbildung das Ergebnis der Anwendung eines globalen Schwellwerts. Bedingt durch die ungleichmäßige Beleuchtung überschneiden sich die Schwellwertbereiche von Objekten und Hintergrund. Während die Punkte in der linken oberen Ecke, die stärker beleuchtet sind, nicht vollständig segmentiert sind, liegt der rechte obere Teil des Hintergrundes noch im Schwellwertbereich. Möchte man die Kreise korrekt darstellen, fällt ein immer größerer Teil des Hintergrundes in den Schwellwertbereich. Soll hingegen der Hintergrund aus der Segmentierung ausgeschlossen werden, würden die stark beleuchteten Punkte immer weiter verschwinden.

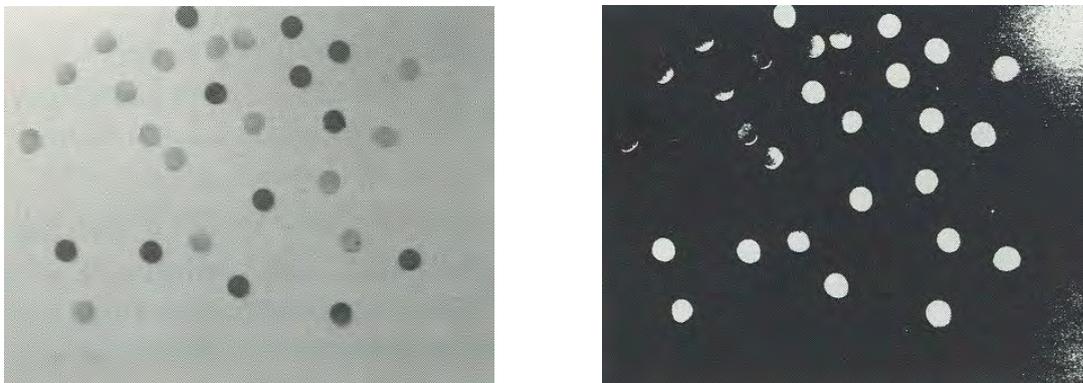


Abbildung 2.1: *Segmentierung über einen globalen Schwellwert bei ungleichmäßiger Beleuchtung. Die Punkte konnten über einen globalen Schwellwert nicht komplett vom Hintergrund getrennt werden [Jähne, 2002]*

Pixelorientierte Verfahren sind schnelle Verfahren und führen bei homogenen Intensitätsgebieten zu guten Ergebnissen. Sie sind einfach zu implementieren und führen zu einer vollständigen Binarisierung der Daten. Ein Nachteil dieser Verfahren ist, dass nicht zusammenhängende Regionen entstehen können, da jedes Element unabhängig von seiner Nachbarschaft untersucht wird [Jähne, 2002;

Handels, 2009]. Pixelorientierte Verfahren eignen sich daher beispielsweise, um in CT-Bildern zwischen Knochen, Weichteilen und Hintergrund unterscheiden zu können.

2.2 Regionenorientierte Verfahren

Regionenorientierte Verfahren betrachten Punktmenge, indem benachbarte Elemente bei der Segmentierung berücksichtigt werden [Jähne, 2002]. Ein Verfahren, welches häufig Anwendung findet, ist das Region Growing. Dabei wird zunächst ein Startpunkt (Seed) spezifiziert. Anschließend wird die Nachbarschaft betrachtet und auf ein für die Region spezifisches Attribut untersucht. Als Beispiel sei hier die Intensität genannt. Befinden sich die Intensitäten der Nachbarelemente im Schwellwertbereich, werden sie mit einem Label versehen und ihre Nachbarn untersucht. Dieser Prozess wird fortgesetzt, bis kein neues Element mit Intensität innerhalb des Schwellwertbereichs mehr existiert [Withey und Koles, 2004].

Zur Veranschaulichung soll das Beispiel eines 2-dimensionalen Region Growings in Abbildung 2.2 und 2.3 dienen. Für jedes Element ist die Intensität eingetragen und die aktuell untersuchten Elemente sind durch einen Rahmen gekennzeichnet. Der Schwellwertbereich sei $S = [25, \infty)$. In Abbildung 2.2 wurde bereits ein Startpunkt gesetzt und dessen 4er-Nachbarschaft untersucht. Alle Nachbarn befinden sich im Schwellwertbereich, sodass für sie ein Label gesetzt wurde. Abbildung 2.3 (links) zeigt das Ergebnis nach 2 weiteren Wachstumsiterationen. Für die drei Elemente wurde kein Label gesetzt, da sich deren Intensitäten außerhalb des Schwellwertbereichs befinden. Das Endergebnis ist in Abbildung 2.3 (rechts) zu sehen. Keines der Elemente besitzt ein ungelabeltes Nachbarelement, dessen Intensität im Schwellwertbereich liegt, sodass die Region nicht mehr weiter wachsen kann.

Ein Nachteil der meisten regionenorientierten Verfahren ist, wie auch bei pixelorientierten Verfahren, die Abhängigkeit der Größe eines Objektes vom gewählten Schwellwert. Wäre für das eben erläuterte Beispiel nur ein geringfügig anderer Schwellwertbereich $S = [20, \infty)$ gewählt worden, wäre das Objekt um einige Elemente ergänzt worden.

0	0	2	7	3	4	1	0	0	0
3	8	9	14	22	18	2	10	0	0
4	19	18	29	35	27	19	8	4	0
7	22	25	70	78	68	38	10	6	0
19	63	30	75	79	47	42	19	4	2
20	29	28	56	70	63	33	24	5	0
9	18	45	38	42	23	22	10	8	0
0	3	17	33	39	27	47	19	20	6
0	0	5	24	22	25	29	30	27	22
0	0	9	8	19	20	23	20	23	10

- Startpunkt
- gelabelte Randelemente
- betrachtete Nachbarn der aktuellen Iteration

Abbildung 2.2: Region Growing unter Verwendung des Schwellwerts $S = [25, \infty)$ und Betrachtung einer 4er-Nachbarschaft nach einer Iteration. Die Intensitäten aller Nachbarn befinden sich im Schwellwertbereich.

0	0	2	7	3	4	1	0	0	0
3	8	9	14	22	18	2	10	0	0
4	19	18	29	35	27	19	8	4	0
7	22	25	70	78	68	38	10	6	0
19	63	30	75	79	47	42	19	4	2
20	29	28	56	70	63	33	24	5	0
9	18	45	38	42	23	22	10	8	0
0	3	17	33	39	27	47	19	20	6
0	0	5	24	22	25	29	30	27	22
0	0	9	8	19	20	23	20	23	10

- Startpunkt
- gelabelte Randelemente
- betrachtete Nachbarn der aktuellen Iteration

0	0	2	7	3	4	1	0	0	0
3	8	9	14	22	18	2	10	0	0
4	19	18	29	35	27	19	8	4	0
7	22	25	70	78	68	38	10	6	0
19	63	30	75	79	47	42	19	4	2
20	29	28	56	70	63	33	24	5	0
9	18	45	38	42	23	22	10	8	0
0	3	17	33	39	27	47	19	20	6
0	0	5	24	22	25	29	30	27	22
0	0	9	8	19	20	23	20	23	10

- Startpunkt
- gelabelte Randelemente

Abbildung 2.3: Links ist das Ergebnis nach 2 weiteren Iterationen zu sehen. Für 3 Nachbarelemente wurde kein Label gesetzt, da ihre Intensitäten nicht im Schwellwertbereich liegen. Die rechte Abbildung zeigt das Endergebnis des Region Growings.

2.3 Kantenbasierte Verfahren

Kantenbasierte Verfahren arbeiten mit maximalen Gradienten, um die Kante zwischen Hintergrund und Objekt möglichst genau bestimmen zu können. Mathematisch bedeutet dies, dass der maximale Wert der Ableitung erster Ordnung oder der Nulldurchgang der Ableitung zweiter Ordnung bestimmt wird. Diese Verfahren werden somit nicht, wie punktorientierte und die meisten regionenorientierten Verfahren, durch eine ungleichmäßige Beleuchtung beeinflusst. Zudem lässt sich der Größenfehler, der bei pixelorientierten Verfahren auftritt, durch Verwendung von Gradienten vermeiden.

Ein wesentlicher Nachteil kantenbasierter Verfahren ist, dass keine Segmentierung möglich ist, wenn die Kanten nicht vollständig vorhanden sind. Abbildung 2.4 zeigt ein Beispiel eines sehr verbreiteten kantenbasierten Verfahrens - der Canny-Algorithmus (auch: Canny edge detector) [Marr und Hildreth, 1980]. Das Ergebnis (rechts) zeigt, dass die Kontur der rechten Niere einige Lücken aufweist, weshalb eine Segmentierung nicht möglich ist.

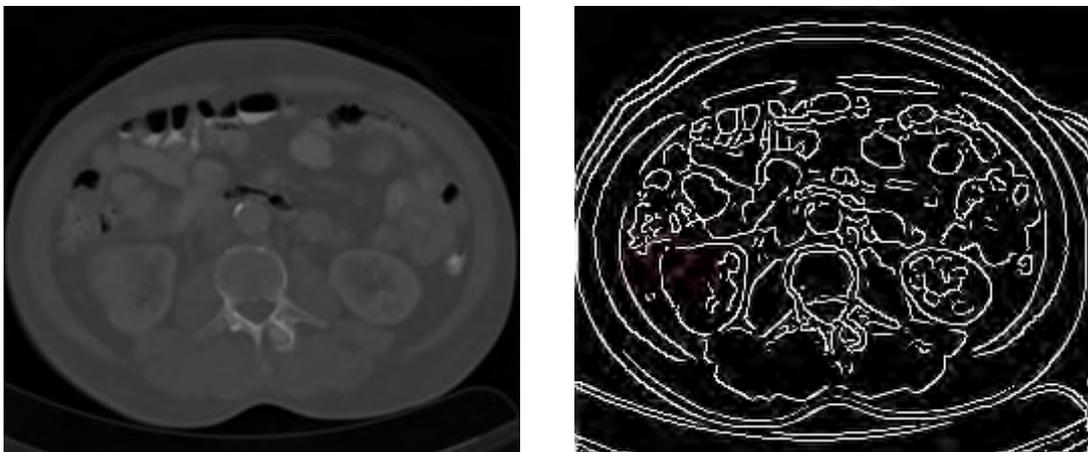


Abbildung 2.4: Anwendung des Canny-Algorithmus. An einigen Stellen sind die Konturen nicht geschlossen, weshalb keine Segmentierung der rechten Niere möglich ist.

2.4 Modellbasierte Verfahren

Sind Kanten jedoch nicht vollständig vorhanden, stoßen sowohl kantenbasierte als auch regionenorientierte Verfahren an ihre Grenzen. Daher wurden Verfahren entwickelt, die sich an der menschlichen visuellen Fähigkeit orientieren, Formen zu erkennen, selbst wenn diese nicht vollständig umrissen sind. In Abbildung 2.5 kann das menschliche visuelle System ein Dreieck erkennen, obwohl nur die Ecken tatsächlich dargestellt sind. Dies ist allerdings nur möglich, weil dem Menschen die Form eines Dreiecks bekannt ist und genau nach diesem Prinzip arbeiten auch modellbasierte Verfahren. Möchte man in einem Bild beispielsweise Kreise segmentieren, so wird in dem Datensatz genau nach dieser Form gesucht. Ähnlich lassen sich die Verfahren auf andere Formen anwenden. Ein Beispiel modellbasierter Verfahren ist die Houghtransformation [Jähne, 2002].

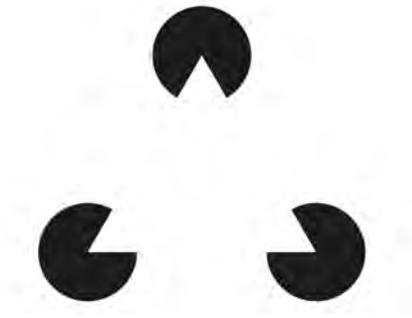


Abbildung 2.5: *Illusion eines Dreiecks [Meier, 2014]*

3 Der Voxel-Man Chirurgie-Simulator

Wie schon in der Einleitung erwähnt, ist eine schwellwertbasierte Segmentierung eine wichtige Voraussetzung, da die Visualisierung des Voxel-Man Simulators mit Schwellwerten arbeitet. Um diesen Zusammenhang deutlich zu machen, wird im Folgenden der Linux-basierte Segmentierer (Abbildung 3.1) sowie die Visualisierung des Voxel-Man Chirurgie-Simulators erläutert. Beide Verfahren arbeiten mit zwei verschiedenen Volumina - dem Grauwertvolumen und dem Objektvolumen. Das Grauwertvolumen enthält die Intensitätswerte der CT-Daten, während das Objektvolumen dazu dient, die Labelwerte der einzelnen Objekte zu setzen.

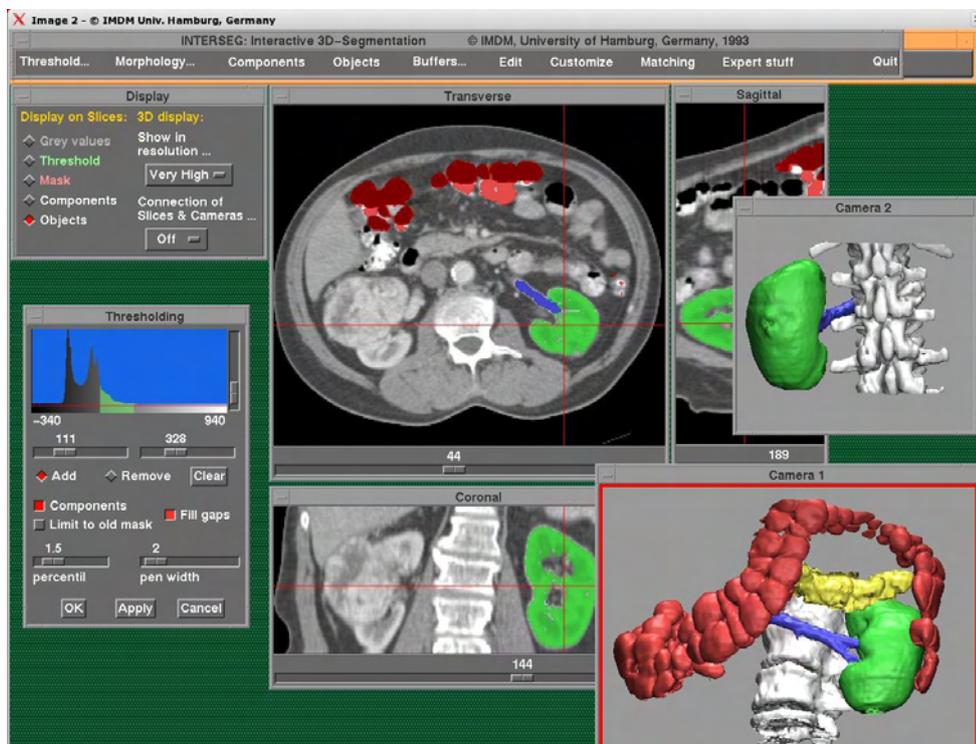


Abbildung 3.1: Voxel-Man Segmentierer auf Linux-Basis

3.1 Manuelle Segmentierung

Das Segmentierungsverfahren des Simulators basiert auf einem 1992 von Höhne und Hanson [1992] entwickelten interaktiven Ansatz, der mit homogenen Intensitätsoberflächen arbeitet. Die Intensität eines Voxels bestimmt also die Zugehörigkeit zu einem morphologischen und funktionellen Bereich. Das Grundgerüst des Verfahrens besteht aus den folgenden 4 Schritten:

1. Durch Mausklick auf das zu segmentierende Objekt spezifiziert der Benutzer einen Intensitätsbereich. Dieser wird vom System ermittelt, indem ein Intensitätsbereich um die Intensität des ausgewählten Voxels gebildet wird. Der Vorgang kann beliebig oft wiederholt werden, um den Bereich zu erweitern. Das Ergebnis der Segmentierung kann dabei zu jedem Zeitpunkt 3-dimensional visualisiert werden.
2. Wurde der Schwellwert soweit angepasst, dass das Objekt charakterisiert ist und eine glatte Oberfläche bildet, stellt man häufig fest, dass weitere Objekte in denselben Intensitätsbereich fallen. Um das zu segmentierende Objekt zu selektieren, bietet das System die Möglichkeit Zusammenhangskomponenten aller Voxel des Intensitätsbereichs zu erzeugen. Dies wird durch eine Connected-Components Analyse realisiert, wodurch zusammenhängende Voxel dasselbe Attribut erhalten. Abbildung 3.2 zeigt ein 2-dimensionales Beispiel, in dem ein Stern segmentiert werden soll. Um Objekte desselben Schwellwertbereichs, die nicht mit dem Stern verbunden sind, zu trennen, wird eine Connected-Components Analyse durchgeführt. Das Ergebnis ist in Abbildung 3.2a zu sehen. Alle anderen Objekte werden anschließend automatisch eliminiert (3.2b).
3. Einige Strukturen ähnlicher Intensitäten sind jedoch durch dünne Strukturen wie beispielsweise Blutgefäße miteinander verbunden oder liegen so dicht beieinander, dass sie vom System nicht als räumlich voneinander getrennte Strukturen zu erkennen sind. Für solche Fälle bietet der Voxel-Man Simulator die Möglichkeit, die Strukturen durch eine vorangestellte Erosion voneinander zu trennen. Der Benutzer spezifiziert die Größe eines Strukturelements, welches als Filter fungiert und somit Randvoxel eliminiert. Ein 2-dimensionales Beispiel eines solchen Strukturelements der Größe 1 ist in Abbildung 3.3 zu sehen. Überdeckt jedes Element des Filters ein gelabeltes Element, so wird nur das Element in der Mitte des Filters in einem neuen Bild oder Volumen gesetzt. Durch erneute Anwendung der

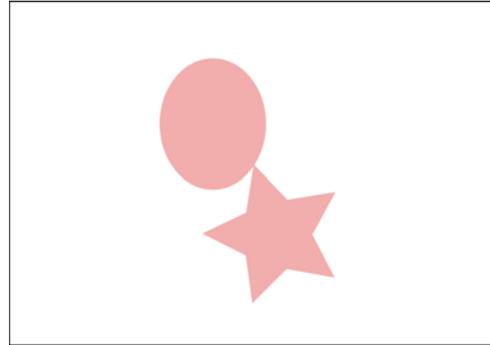
Zusammenhangskomponenten wird anschließend geprüft, ob die Objekte voneinander getrennt werden konnten. War die Trennung erfolgreich, wird das zu segmentierende Objekt ausgewählt. Im anderen Fall muss der Vorgang gegebenenfalls mit einem größeren Strukturelement wiederholt werden. Abbildung 3.2c zeigt das Ergebnis der Erosion. Der Stern ist nun von dem anderen Objekt getrennt und kann durch eine Connected-Components Analyse separiert werden.

4. Wurde das Objekt durch die Erosion erfolgreich segmentiert, werden die entfernten Voxel durch eine Dilatation wieder hinzugefügt. Auch bei der Dilatation wird ein Strukturelement verwendet, welches den gesamten Datensatz durchläuft. Überdeckt das mittlere Element ein gelabeltes Element, so werden alle Elemente des Strukturelements im neuen Bild oder Volumen gesetzt. Auch für die Dilatation ist ein 2-dimensionales Beispiel in Abbildung 3.3 zu sehen. Wie die Abbildung weiterhin zeigt, sind die beiden Operationen nicht invers zueinander, weshalb das Strukturelement für die Dilatation in der Regel ein wenig größer gewählt wird. Um keine Elemente zu der Region hinzuzufügen, die nicht zur ursprünglichen Region gehörten, wird die Dilatation auf die Elemente beschränkt, deren Intensitäten im Schwellwertbereich liegen. Dies ist ein sehr wichtiger Punkt für die anschließende Visualisierung, da ein Element, dessen Label gesetzt ist, aber dessen Intensitätswert nicht im entsprechenden Schwellwertbereich liegt, nicht korrekt dargestellt werden kann. Das Ergebnis ist in Abbildung 3.2d zu sehen.

Häufig kommt es bei der Segmentierung zu Abweichungen der oben beschriebenen Punkte. Die Operationen werden in einer anderen Reihenfolge ausgeführt oder es werden weitere Methoden angewandt, die das System zur Verfügung stellt. Hierzu gehören vor allem Bool'sche Operatoren, Fülloperatoren zur Segmentierung innerer Strukturen und das Einbringen von Schnitten zur Trennung, bei der die Erosion nicht ausreicht.



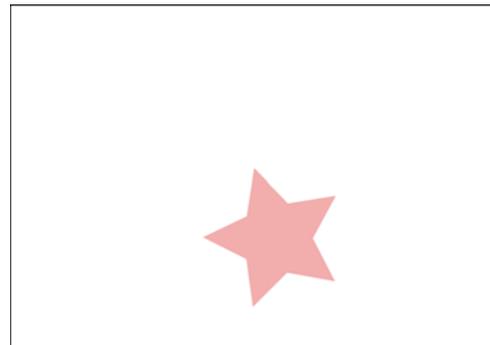
(a) Segmentierte Objekte nach Anwendung einer Connected-Components Analyse



(b) Nach Auswahl eines Objektes wurden alle anderen Objekte eliminiert



(c) Die zusammenhängenden Objekte konnten durch die Erosion getrennt werden. Nach Anwendung einer Connected-Components Analyse erhalten die getrennten Objekte unterschiedliche Label.



(d) Nach Auswahl des gewünschten Objektes wird eine Dilatation mit Begrenzung auf den Schwellwertbereich durchgeführt.

Abbildung 3.2: abstraktes Beispiel einer Segmentierung mit dem Linux-basierten Voxel-Man Segmentierer

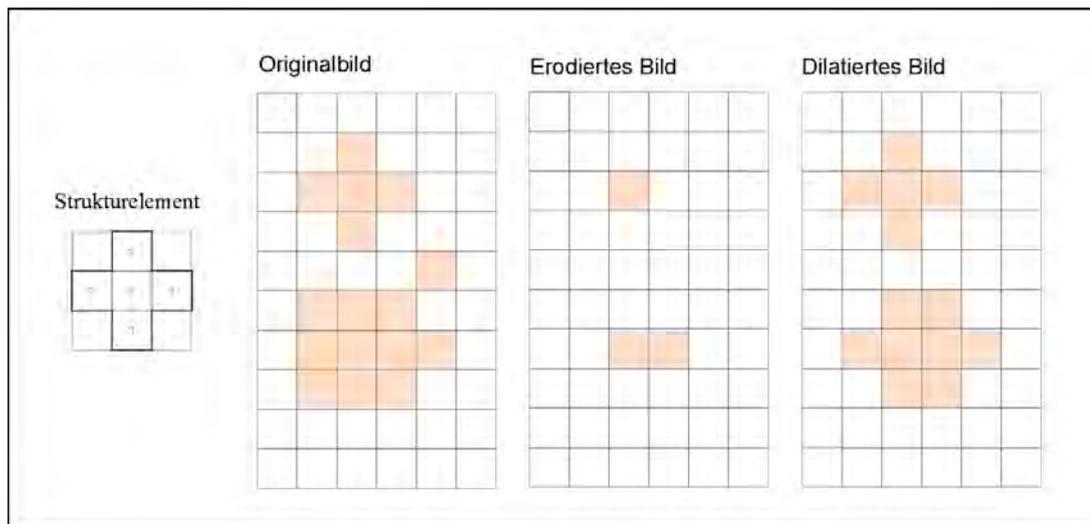


Abbildung 3.3: Die Grafik zeigt das Originalbild, das Ergebnis einer Erosion und einer anschließenden Dilatation. Das Strukturelement fungiert als Filter, der über das gesamte Bild läuft. Es ist deutlich zu sehen, dass durch eine Dilatation eines erodierten Bildes nicht die Originaldaten wiederhergestellt werden können.

3.2 3D-Visualisierung

Bei der Visualisierung im Voxel-Man Simulator handelt es sich um eine volumenbasierte Visualisierung. Der wesentliche Vorteil gegenüber der oberflächenbasierten Visualisierung ist die Erhaltung der Originalbilddaten. Während bei der oberflächenbasierten Visualisierung ein Objekt durch eine leere Hülle aus Polygonen dargestellt wird, wird bei der volumenorientierten Visualisierung das komplette Volumen erhalten. Dies ist für den Voxel-Man Simulator von besonderer Bedeutung, da das Volumen während einer virtuellen Operation durch Bohren, Schneiden oder andere chirurgische Eingriffe manipuliert wird. Für eine polygonale Darstellung hieße das eine komplett neue Berechnung der Polygonstruktur, was einen erheblichen Zeitaufwand bedeutet.

Für die Visualisierung des Voxel-Man Chirurgie-Simulators wurde das Strahlverfolgungsverfahren (engl. Ray-Tracing oder Ray-Casting) [Foley u. a., 1990] genutzt. Dabei wird ein Sehstrahl durch das Volumen geschickt, der es in regelmäßigen Abständen abtastet. Für jeden Abtastpunkt werden Farb- und Transparenzwerte, sowie die Schattierung bestimmt. Trifft der Strahl auf eine Oberfläche, wird die Oberflächennormale und die zum Betrachter reflektierte Intensität berechnet.

Für die voxelbasierte Visualisierung wurde lange Zeit mit binären Volumeninformationen gearbeitet. In Höhne und Bernstein [1986] wurde erstmals ein

Verfahren beschrieben, um Oberflächennormalen aus Grauwertgradienten zu berechnen. Durch Verwendung des Partial-Volumen-Effekts tomographischer Volumendaten konnte eine deutlich größere Variabilität der Oberflächennormalen und somit eine qualitativ hochwertigere Oberflächendarstellung gegenüber binären Volumina erzielt werden.

Eine weitere Entwicklung, auf die die Visualisierung des Voxel-Man Simulators aufbaut, ist die der Attributverwendung. Diese wurde unabhängig voneinander von Höhne und Tiede [1987] und Stiehl [1987] entwickelt. Dabei wird jedem Voxel ein Attribut zugewiesen, welches es eindeutig einer Region zuordnet. Jedem Objekt werden zusätzlich Darstellungsparameter wie Farbe und Reflexionseigenschaften zugewiesen.

Für eine realitätsnahe und glatte Oberfläche reicht eine Vergabe von Attributen an einzelne Voxel nicht aus. Es ist notwendig, den Subvoxelkoordinaten der Abtastpunkte die korrekten Attribute zuzuweisen. Um die Objektgrenzen zwischen den Regionen verschiedener Attribute möglichst genau zu bestimmen, wurde eine Visualisierung multiattributierter und multiparametrischer Volumendaten entwickelt [Tiede, 1999], welche im Folgenden näher erläutert wird.

Zunächst wird der Intensitätswert der Abtastposition durch trilineare Interpolation erzeugt. Anschließend wird ein Zähler mit 0 initialisiert und die Zugehörigkeit der Abtastposition zu einem Objekt in seiner $2 \times 2 \times 2$ -Umgebung bestimmt. Dazu wird der interpolierte Intensitätswert der Abtastposition mit dem Schwellwertbereich der dem Nachbarvoxel zugeordneten Objekt verglichen. Liegt der Intensitätswert innerhalb der Schwellwertgrenzen, wird der Zähler um 1 erhöht. Kommen mehrere Voxel desselben Objektes in der Nachbarschaft vor, werden diese nur einmal berücksichtigt.

Ist dieser Vorgang beendet, kann anhand des Zählers bestimmt werden, ob die Abtastposition einem Objekt eindeutig zuzuordnen ist. Wurden mehrere Objekte gefunden, dessen Schwellwertbereiche zum Intensitätswert passen, konnte keine eindeutige Zugehörigkeit bestimmt werden. Dies ist beispielsweise bei überlappenden Schwellwertbereichen möglich. In diesem Fall findet eine heuristische Zuordnung der Position zu einem Objekt statt, auf die hier nicht näher eingegangen werden soll [Tiede, 1999].

Nachdem die Objektzugehörigkeit des Abtastwertes geklärt ist, bleibt nach wie vor die Frage der genauen Lage der Oberfläche. Trifft der Strahl auf eine Position P_i , die dargestellt werden soll, muss die Objektgrenze zwischen dieser und der vorherigen Abtastposition P_{i-1} liegen. Um die Grenze genau zu bestimmen,

wurde von Tiede [1998] ein Verfahren entwickelt, welches sich auf die Annahme stützt, der Intensitätsverlauf zwischen zwei Abtastpunkten sei linear. Um den Schnittpunkt B der Objektoberfläche mit dem Abtaststrahl zu bestimmen, wurde daher die Strecke zwischen beiden Punkten in demselben Verhältnis geteilt, wie der Schwellwert die Intensitätswerte teilt. Mathematisch kann dies durch die folgende Formel beschrieben werden:

$$B = P_i - t(P_i - P_{i-1}) \quad \text{mit} \quad \frac{S - g_{i-1}}{g_i - g_{i-1}},$$

„wobei S den unteren Schwellwert angibt, falls $g_{i-1} < S$ ist. Anderenfalls ist S der obere Schwellwert des Intensitätsbereichs, der das Objekt charakterisiert“ [Tiede, 1999].

Da es sich jedoch um eine lineare Interpolation in drei Dimensionen handelt, ist der Intensitätsverlauf zwischen zwei Punkten nicht linear. Er wird ebenfalls durch die Intensitätswerte der anderen Dimensionen beeinflusst. Aufgrund dieser Erkenntnis wurde ein Bisektionalgorithmus entwickelt [Tiede, 1999], welcher anhand eines Beispiels erläutert werden soll.

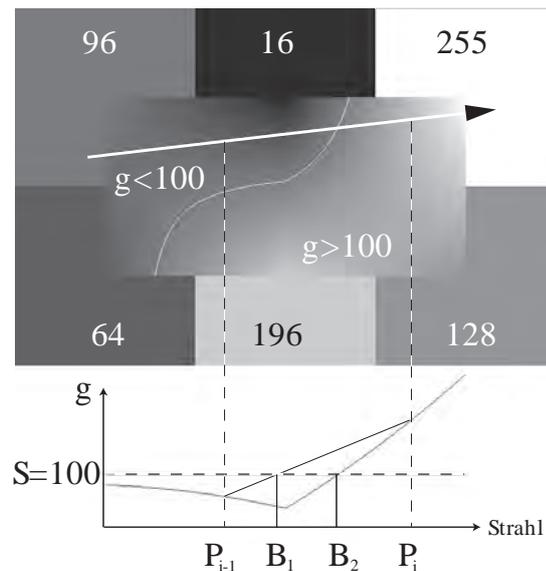


Abbildung 3.4: Die Grafik zeigt die deutliche Abweichung eines durch Linearisierung ermittelten Intensitätsverlauf zu dem durch Bisektion ermittelten. Lineare Interpolation an den Abtastpositionen erzeugt eine falsche Objektgrenze an der Stelle B_1 . Durch Bisektion wird die korrekte Objektgrenze an der Stelle B_2 ermittelt [Tiede, 1999].

Gegeben seien die Abtastpunkte P_i und P_{i-1} mit den interpolierten Werten $g_i = 120$ und $g_{i-1} = 68$. Der untere Schwellwert, der das Objekt vom Hintergrund trennt, sei $S = 100$. Nun wird die Strecke $\overline{P_{i-1}P_i}$ in der Mitte geteilt und an dieser Position M_1 durch trilineare Interpolation die Intensität $g = 79$ bestimmt. Da dieser Wert $< S$ ist, muss die Oberfläche zwischen M_1 und P_i liegen. Es wird also die Strecke $\overline{M_1P_i}$ in der Mitte geteilt und für diese Position (M_2) die Intensität durch trilineare Interpolation bestimmt. Dieser Vorgang wird fortgesetzt bis $S < \varepsilon$ ist. Abbildung 3.4 zeigt den durch Bisektion und durch Linearisierung approximierten Intensitätsverlauf. Durch die Linearisierung wird die eine falsche Objektgrenze an der Position B_1 ermittelt, während Bisektion zur korrekten Objektgrenze an der Stelle B_2 führt. Die sprunghafte Änderung des Intensitätsverlaufs ist durch die lineare Interpolation bedingt. Bei Anwendung höherer Interpolation (z. B. Splines) wäre auch die Differenzierbarkeit in der ersten Ableitung und damit die Kontinuität des Grauwertverlaufs gewährleistet.

4 Segmentierungsapplikationen

Nachdem durch Kapitel 3 auch die Wichtigkeit der Segmentierung über Schwellwerte deutlich gemacht wurde, werden in diesem Kapitel die untersuchten Segmentierungsapplikationen präsentiert und deren Ergebnisse ausgewertet. Für jede der sechs untersuchten Applikationen wurde die Segmentierung der Cochlea durchgeführt. Die Cochlea wurde aus den Risikostrukturen des Innenohrs ausgewählt, da aufgrund der Schwellwertüberschneidung mit den Bogengängen Trennoperatoren benötigt werden. Da ein geeigneter Schwellwertbereich für die Cochlea bekannt ist, ist eine Trennung über morphologische Operatoren nicht notwendig und soll nur zur Demonstration durchgeführt werden. Als Datensatz wurde von der Arbeitsgruppe ein CT-Datensatz eines Schädels zur Verfügung gestellt. Der Schwellwertbereich, der sich bei dem CT-Datensatz für die Cochlea als geeignet erwiesen hat, ist 693 - 1251 und soll im Folgenden für die Segmentierung verwendet werden.

4.1 MeVisLab

MeVisLab wurde von der MeVis Medical Solution AG und der Fraunhofer Research GmbH in Bremen für die Verarbeitung und Visualisierung medizinischer Bilddaten entwickelt. Es wurde in C++ geschrieben und verwendet Qt für das graphische Userinterface. MeVisLab ist für Microsoft Windows, Linux und MacOS X verfügbar und wurde in dieser Arbeit in der Windows-Version verwendet. Die MeVis Image Processing Library (ML) bietet ein generisches Framework für die Bildanalyse. Es enthält zusätzlich die Insight Segmentation und das Insight (ITK) und Visualization Toolkit (VTK). Jeder der über 300 Algorithmen wird durch ein eigenes Modul repräsentiert, welches vom Anwender mit anderen Modulen zu einem Netzwerk verbunden werden kann. Außerdem bietet es die Möglichkeit, eigene Module auf der Basis von Java Script oder Python zu schreiben [MeVisLab 2014].

4.1.1 Material und Methoden

Im Rahmen der vorliegenden Arbeit wurden unter Verwendung der ML verschiedene Netzwerke aufgebaut, um eine schwellwertbasierte Segmentierung vorzunehmen. Eines dieser Netzwerke wurde entsprechend des alten Voxel-Man Systems entworfen (Abbildung 4.1), bei dem eine Trennung von Objekten über morphologische Operatoren erzielt wird.

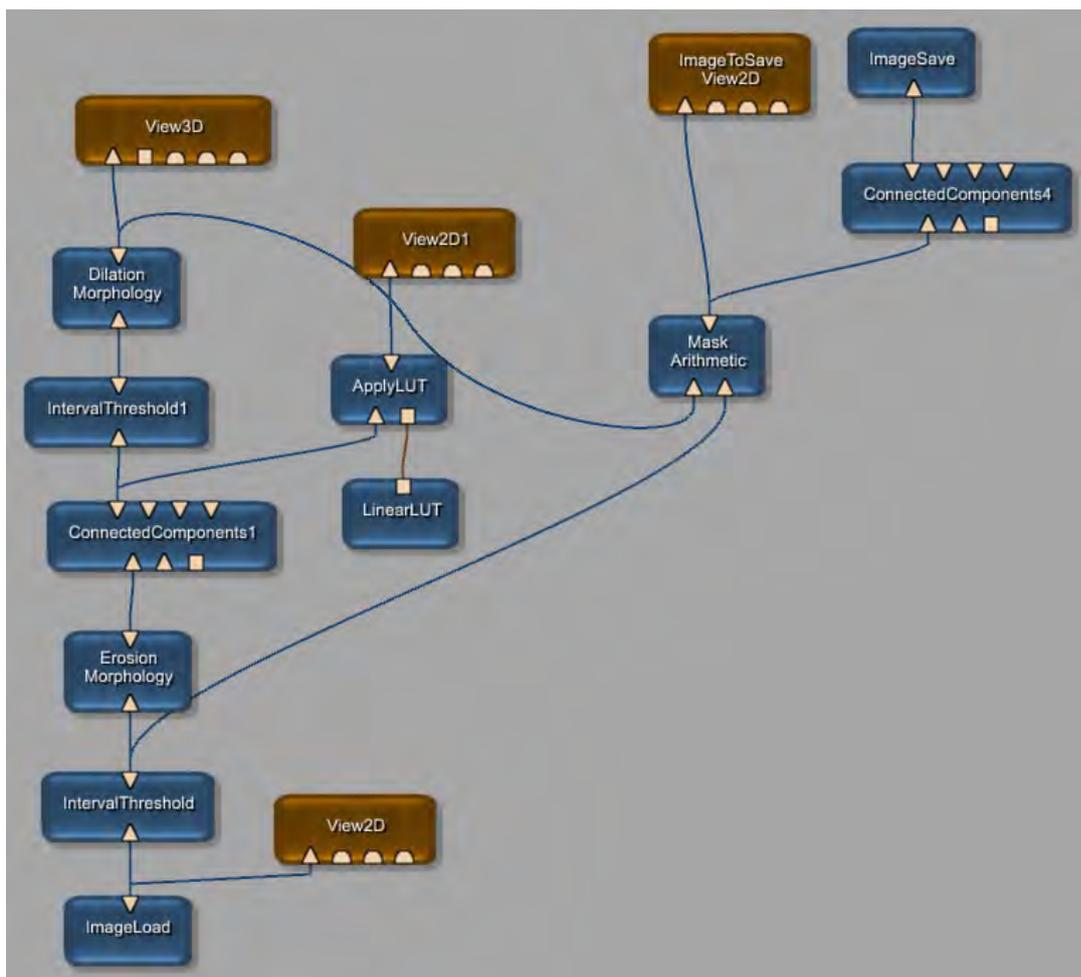


Abbildung 4.1: Netzwerk einer Segmentierung über einen globalen Schwellwert in Anlehnung an den Linux-basierten Voxel-Man Segmentierer

Zur Segmentierung der Cochlea wurde zunächst mittels des „ImageLoad“-Moduls ein Datensatz im RAW-Format eingelesen. Das Modul „View2D“ ermöglicht eine 2-dimensionale Ansicht des Datensatzes. Das Scrollen durch verschiedene Schichten realisiert die Ansicht der dritten Dimension.

Durch das „IntervallThreshold“-Modul können Schwellwerte entsprechend der zu segmentierenden Struktur eingestellt werden. Abbildung 4.2 zeigt die Schaltfläche sowie die 2-dimensionale Ansicht nach der Schwellwertsetzung. Nach Einstellen des schon erwähnten Schwellwertbereichs von 693 - 1251 wurden alle Voxel mit Intensitäten innerhalb dieses Bereichs auf den eingestellten „Inner Value“ 1 gesetzt. Alle anderen Voxel erhalten den Wert 0. Abbildung 4.2 zeigt weiterhin, dass eine Reihe anderer Strukturen in diesem Schwellwertbereich liegen.

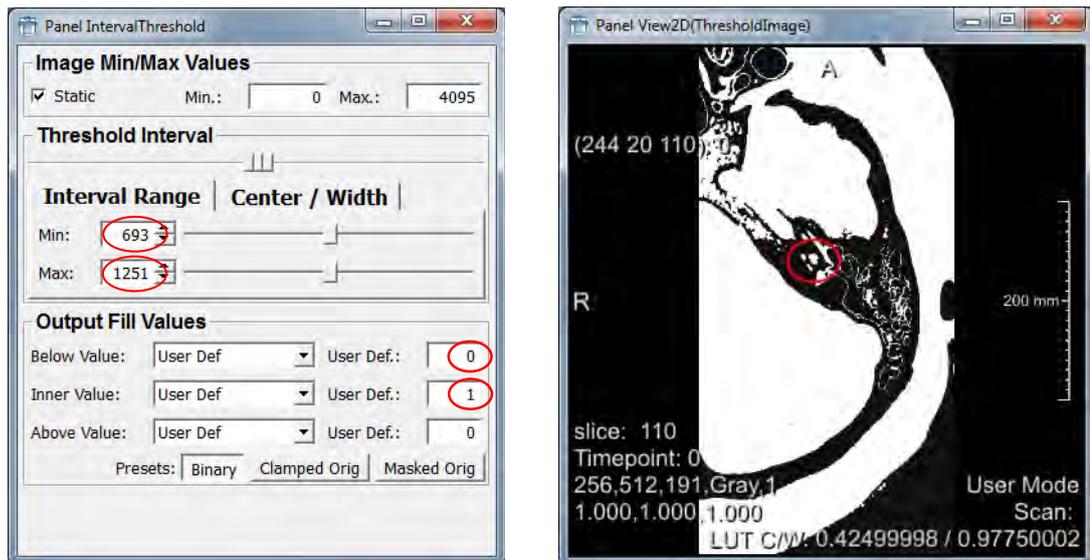


Abbildung 4.2: Schaltfläche und Ergebnis des „IntervallThreshold“-Moduls. Die rote Markierung in der rechten Abbildung zeigt die Position der Cochlea.

In Folge einer Erosion, die das Modul „Morphology“ anbietet, werden Strukturen gleicher Intensität voneinander getrennt. Wie Abbildung 4.3 zeigt, ist anhand der 2D-Ansicht schwer festzustellen, ob Verbindungen zwischen den einzelnen Objekten bestehen. Daher wurde ein „ConnectedComponents“-Modul sowie eine Lookup-Tabelle (LUT) und ein „View2D“-Modul angeschlossen. Für die Connected-Components Analyse wurde die kleinste 3-dimensionale Nachbarschaft (6er-Nachbarschaft) ausgewählt, da somit nicht mehr Voxel als nötig erodiert werden müssen, um eine Trennung zu erzielen. Wie Abbildung 4.4 zeigt, wurde für die minimale Cluster-Größe 0,1 ml gewählt, um auch kleine Strukturen wie die Cochlea zu identifizieren. Dies führte zu einer Segmentierung von 26 Objekten. Die Objekte erhalten dadurch Labelwerte zwischen 1 und 26 und werden durch die LUT in unterschiedlichen Farben dargestellt. Verändert man nun die Kernel-Werte der Erosion, kann anhand des „View2D“-Moduls eine Trennung

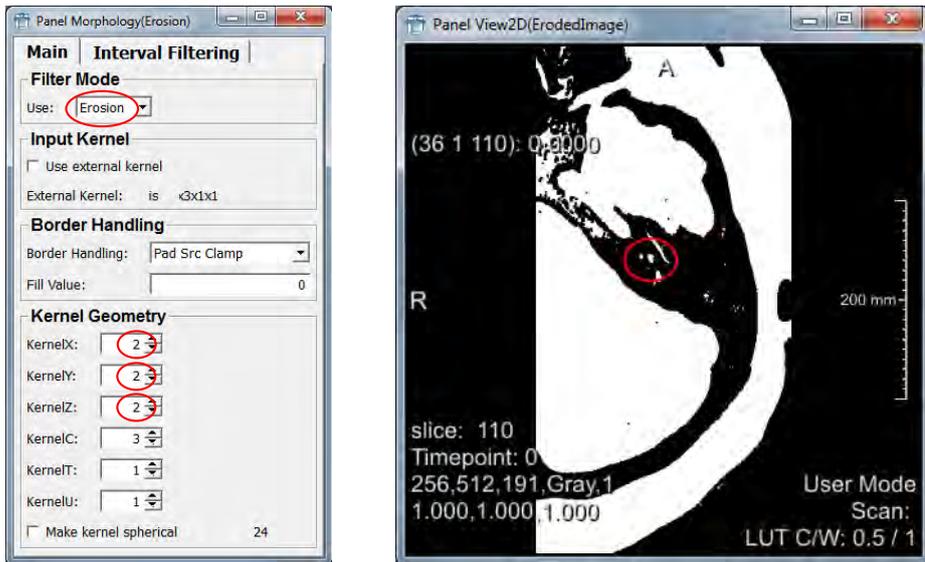


Abbildung 4.3: Schaltfläche und Ergebnis des Erosion-Moduls

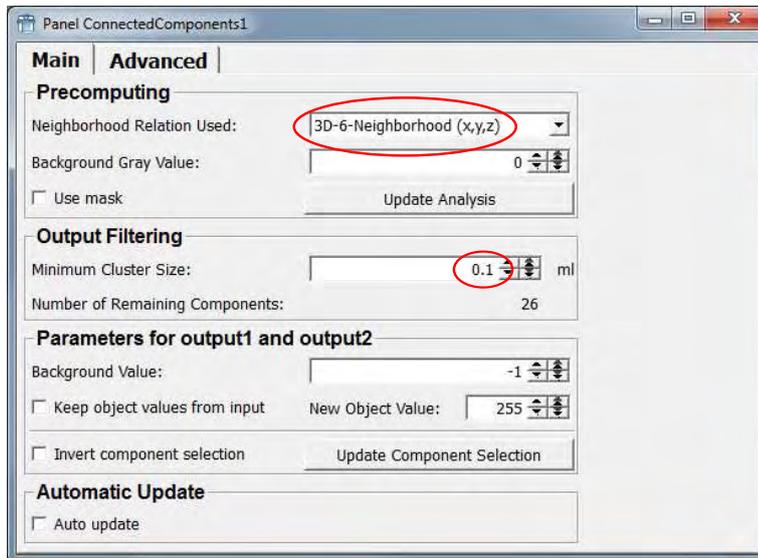


Abbildung 4.4: Schaltfläche des „ConnectedComponents“-Moduls

der einzelnen Objekte detektiert werden (Abbildung 4.5).

Für die Eliminierung der anderen Strukturen wurde erneut ein „IntervalThreshold“-Modul verwendet. Sowohl für den unteren als auch für den oberen Schwellwert wurde der Intensitätswert der segmentierten Cochlea gewählt, der anhand des „View2D“-Moduls abgelesen werden kann.

Nun konnte eine Dilatation durchgeführt werden, um dem Voxelverlust der Erosion entgegenzuwirken. Die entsprechende Schaltfläche und das Ergebnis sind

in Abbildung 4.6 dargestellt. Mittels des Moduls „Morphology“, welches auch eine Dilatation anbietet, ist eine Begrenzung auf den Schwellwertbereich nicht möglich. Um diese Einschränkung zu realisieren, wurde das Ergebnis der Dilatation mit dem Schwellwertvolumen maskiert. Umgesetzt wurde dies durch Verwendung des „Arithmetic“-Moduls, welches eine voxelweise Multiplikation der beiden Volumina anbietet.

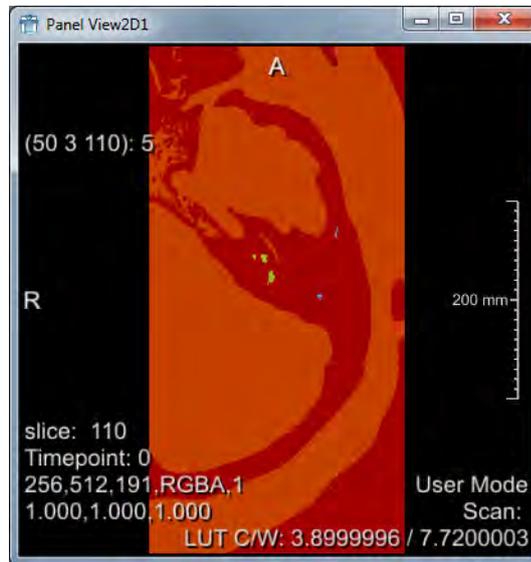


Abbildung 4.5: Ergebnis der Connected-Components Analyse und nachfolgender Anwendung einer LUT. Die Cochlea ist als grüner Bereich in der Mitte des Bildes auszumachen.

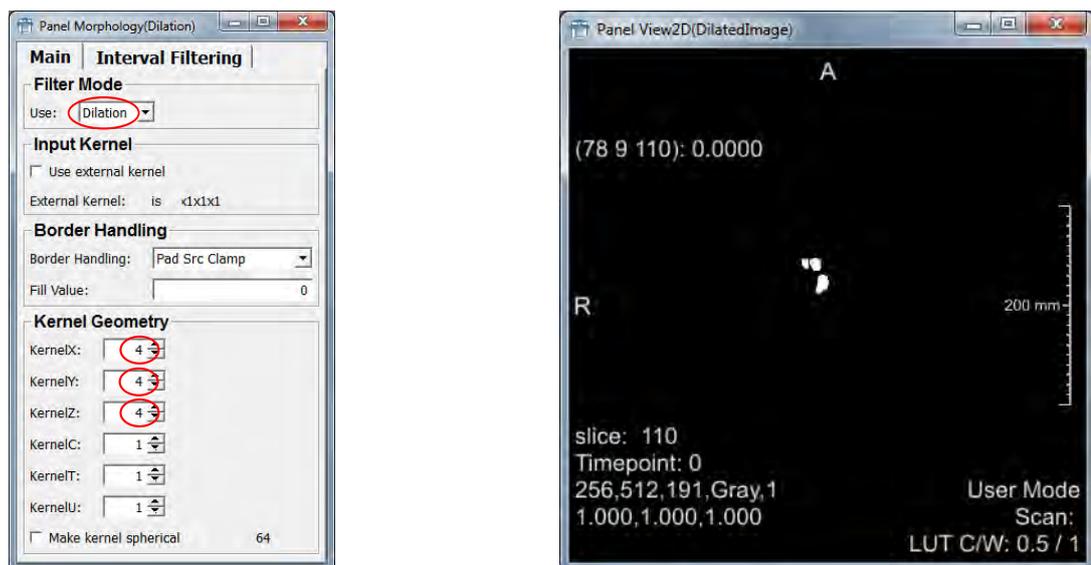


Abbildung 4.6: Schaltfläche und Ergebnis des Dilatation-Moduls

Eine weitere Segmentierung wurde über ein Region Growing realisiert. Der Aufbau des entsprechenden Netzwerkes ist in Abbildung 4.7 zu sehen.

Zunächst wurden die Daten im RAW-Format eingelesen und über einen globalen Schwellwert segmentiert. Um ein Wachsen in angrenzende Regionen von Beginn an zu unterbinden, wurde schon vor dem Region Growing eine Erosion durchgeführt. Für den Schwellwertbereich wurde 693 - 1251 und für die Kernel-Werte der Erosion 3 gewählt.

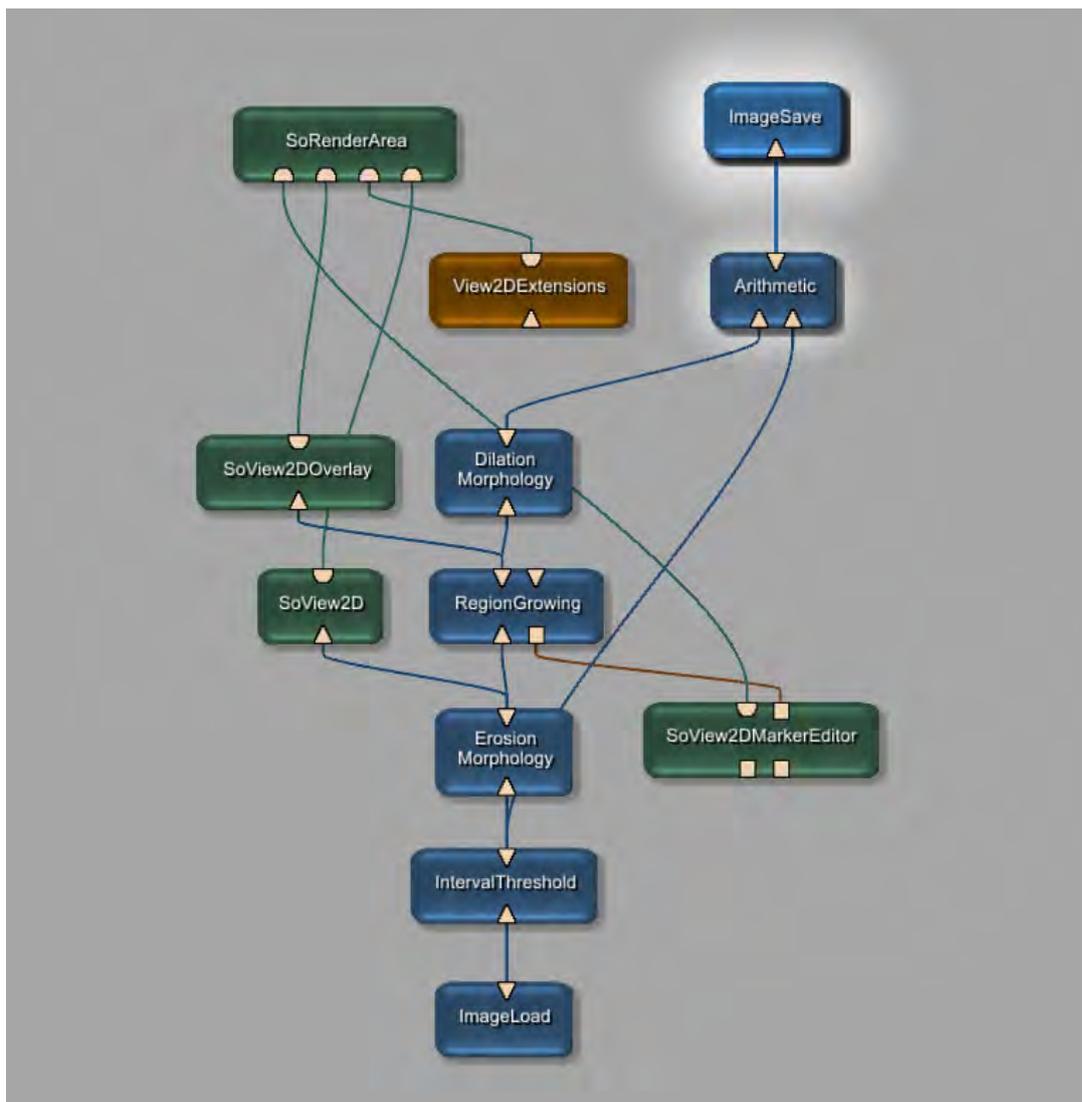


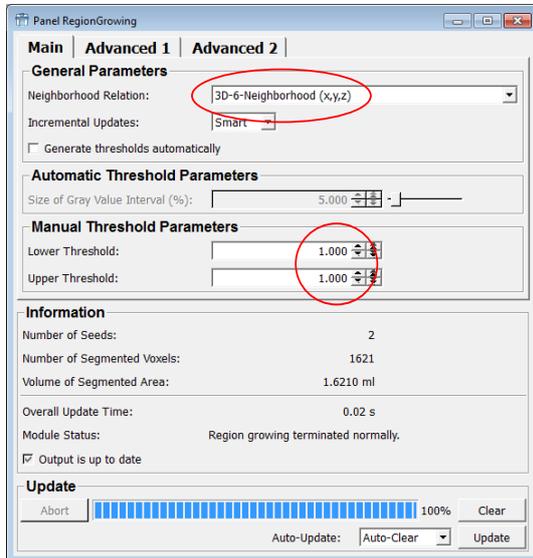
Abbildung 4.7: MeVisLab-Netzwerk zur Segmentierung über ein Region Growing

Das Region Growing selbst setzt sich aus mehreren Modulen zusammen. Über das Modul „RegionGrowing“ lassen sich ein unterer und ein oberer Schwellwert, sowie eine Nachbarschaft einstellen, in der die Region wachsen soll. Da das

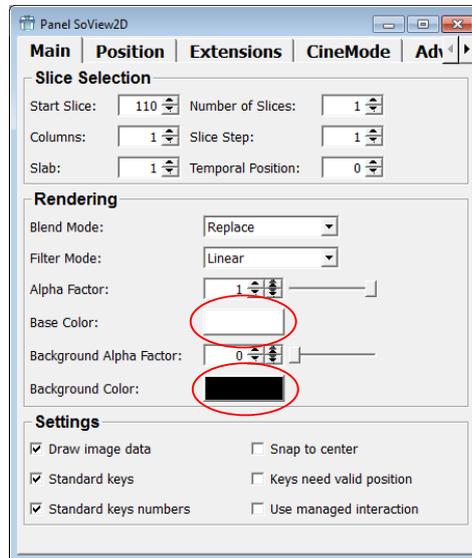
Region Growing auf der bereits durch das „Threshold“-Modul segmentierten Region des Labels 1 arbeitet, wurde dieser Wert für beide Schwellwertgrenzen des „RegionGrowing“-Moduls gewählt (Abbildung 4.8a). Die Kombination der Module „SoRenderArea“ und „SoView2D“ ermöglicht eine 2-dimensionale Darstellung der Daten, auf welches das Region Growing angewandt werden soll. Die Einstellung weiß für „Base Color“ und schwarz für „Background Color“ resultiert in einer weißen Darstellung der segmentierten Region auf schwarzem Hintergrund (Abbildung 4.8b). Eine gemeinsame Darstellung dieser Daten und dem Ergebnis des Region Growings wird über das Modul „SoView2DOverlay“ realisiert (Abbildung 4.8c), während der „SoViewMarkerEditor“ die Setzung des Startpunktes für das Region Growing ermöglicht (Abbildung 4.8d). Durch den „Blend Mode“ wird eine Überlagerung der Daten gewährleistet und als Farbe für die segmentierte Region wurde die grün gewählt. Für die Größe der Startpunkte wurde die kleinst mögliche Größe (2) und als Farbe der Startpunkte gelb eingestellt.

Abbildung 4.9 zeigt das Ergebnis des Region Growings in der „SoRenderArea“. Zwei Startpunkte wurden in der Region der Cochlea gesetzt, was zu einer Segmentierung des grün markierten Bereichs führte.

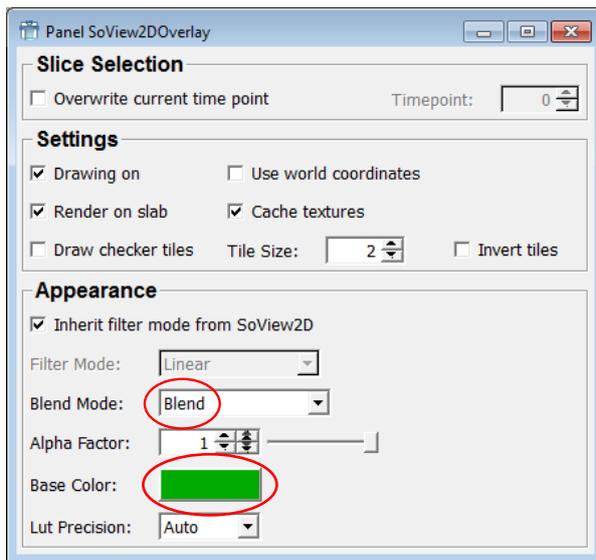
Anschließend wurde eine Dilatation mit den Kernel-Werten 4 durchgeführt. Die Einschränkung auf den Schwellwertbereich wurde, wie schon bei der Segmentierung über einen globalen Schwellwert, durch eine Maskierung mit dem Ergebnis der Schwellwertsetzung realisiert.



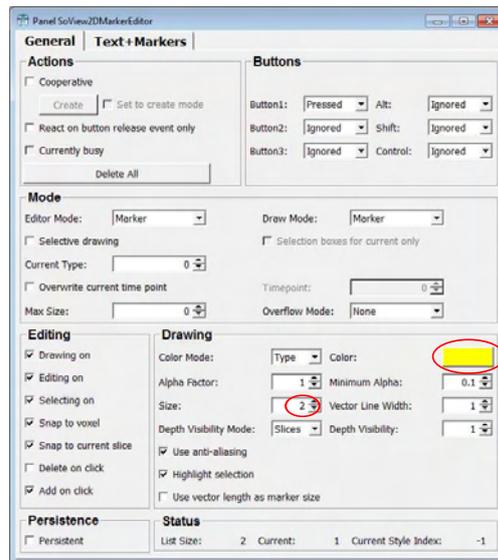
(a) "RegionGrowing"-Modul



(b) "SoView2D"-Modul



(c) "SoView2DOverlay"-Modul



(d) "SoView2DMarkerEditor"-Modul

Abbildung 4.8: Schaltflächen für das Region Growing

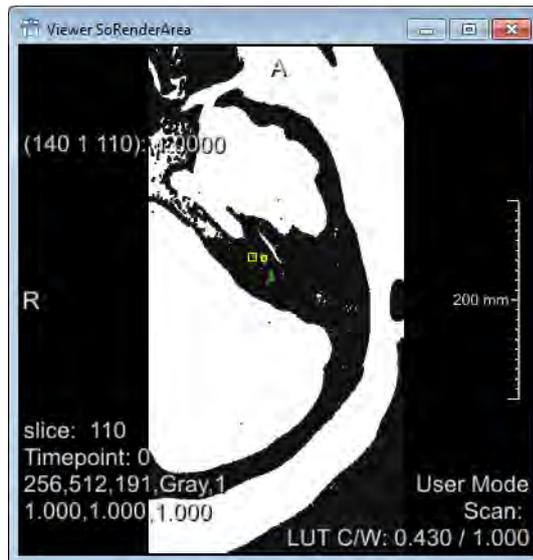


Abbildung 4.9: Ergebnis des Region Growings in der „SoRenderArea“. Durch das Setzen zweier Startpunkte (gelb) wurde die grün markierte Region segmentiert.

4.1.2 Ergebnisse

Die Segmentierung mit MeVisLab hat gezeigt, dass eine schwellwertbasierte Segmentierung sowohl über einen globalen Schwellwert als auch über ein Region Growing möglich ist. MeVisLab bietet also globale und lokale Operatoren zur schwellwertbasierten Segmentierung an. Nach jedem Segmentierungsschritt kann das Ergebnis im 2- wie auch im 3-Dimensionalen visualisiert werden. Durch die über 300 verschiedenen Algorithmen, die zu einem Netzwerk zusammengeschaltet werden können, ist MeVisLab eine sehr flexible Software und bietet somit diverse Möglichkeiten zur Segmentierung medizinischer Bilddaten. Während der Segmentierung mit MeVisLab kam es nicht zu Programmabstürzen.

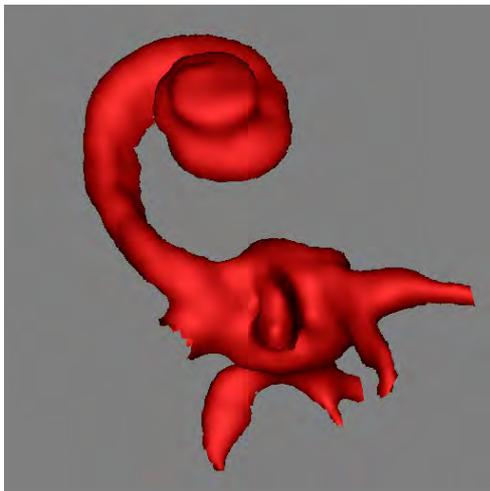
Ein Nachteil zeigte sich in der 3D-Visualisierung, dessen Aktualisierung je nach vorgeschaltetem Modul unterschiedlich lange dauert und bei der Einstellung des Schwellwerts nicht in Echtzeit möglich war.

Ein weiterer Nachteil ist, dass die Dilatation des Moduls „Morphology“ nicht auf den Schwellwertbereich begrenzt werden kann. Alternativ wurde dies über eine Maskierung mit dem ursprünglichen Schwellwertdatensatz realisiert.

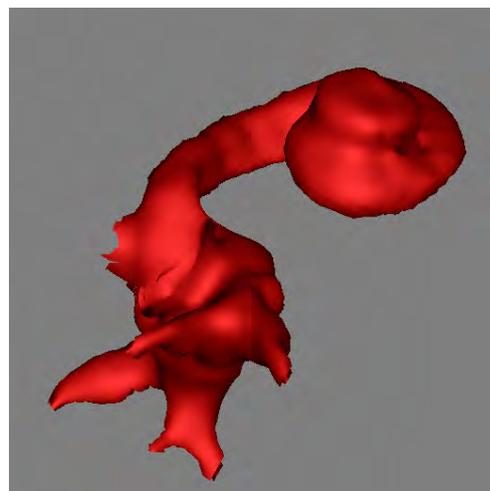
Die Bedienung stellte das größte Problem dar. Zunächst erfordert das Aufbauen eines Netzwerkes sowohl eine Einarbeitung in die Bildverarbeitung, als auch in MeVisLab selbst. Da die Algorithmen völlig frei zusammengestellt werden können, kann eine Segmentierung auf unterschiedlichste Weise erfolgen. Allerdings

setzt dies auch voraus, dass man mit den Algorithmen vertraut ist, um sie gezielt auswählen und Parameter einstellen zu können. Eine weitere Einschränkung stellt die ständige Aktualisierung der einzelnen Module und Verbindungen dar. Soll beispielsweise ein neuer Datensatz geladen werden, so muss jedes Modul aktualisiert und die Verbindung teilweise neu erstellt werden. Auch die Kontraste sind in der 2D-Ansicht oft nicht automatisch an den Datensatz angepasst, sodass zunächst nur ein schwarzes Bild erscheint.

Die segmentierten Daten können im DICOM-Format abgespeichert werden. Es stellte sich jedoch heraus, dass die Daten nicht DICOM-konform sind und somit nicht ohne Weiteres vom Simulator importiert werden konnten. Daher wurde von der Arbeitsgruppe Voxel-Man ein Importer für die MeVisLab DICOM-Daten zur Verfügung gestellt. MeVisLab bietet keine Möglichkeit, die für die Visualisierung notwendigen Schwellwerte zu exportieren. Daher wurden diese im Nachhinein mittels des in MeVisLab entstandenen Objektvolumen und der Grauwertdaten ermittelt. Als Schwellwertgrenzen wurden der minimale und maximale in der segmentierten Region vorkommende Intensitätswert verwendet. Dies führte sowohl für das Verfahren eines globalen Schwellwerts als auch für das Region Growing zu einem Schwellwertbereich von 689 - 1246, welcher sich nur minimal von den eingestellten Schwellwerten 693 und 1251 unterscheidet.



(a) Segmentierung über einen globalen Schwellwert



(b) Segmentierung über ein Region Growing

Abbildung 4.10: Voxel-Man Visualisierung der mit MeVisLab segmentierten Cochlea

Wie Abbildung 4.10a zeigt, führte die Voxel-Man Visualisierung mit nachträglich ermitteltem Schwellwertbereich zu einer glatten Oberfläche der Cochlea. Durch die Überschneidung der Intensitätsbereiche mit den Bogengängen, wurde ein Teil dieser in die Segmentierung eingeschlossen.

Abschließend lässt sich festhalten, dass ein Benutzer seine mit MeVisLab segmentierten Daten in den Voxel-Man Simulator importieren kann, solange auf Schwellwertbasis gearbeitet wird. Es erfüllt jedoch nicht alle Anforderungen, sodass es nicht als Standardsegmentierer für den Voxel-Man Simulator verwendet werden kann.

4.2 3D Slicer

Eine weitere Segmentierungsapplikation, die untersucht wurde, ist 3D Slicer. Die Entwicklung dieser Software begann 1998 im Rahmen eines Masterarbeitsprojekts zwischen dem Surgical Planning Laboratory des Brigham and Women's Hospital in Boston und dem MIT Artificial Intelligence Laboratory in Cambridge. Die erste komplett neu überarbeitete Version wurde 2007 veröffentlicht. Die aktuellste Version 3D Slicer 4.3, die im Rahmen dieser Arbeit verwendet wurde, ist seit September 2013 auf dem freien Markt. Sie ist vorwiegend in C++ geschrieben und für verschiedene Betriebssysteme, einschließlich Windows, Linux und Mac OS X verfügbar [3D Slicer 2014].

4.2.1 Material und Methoden

Die Bilddaten wurden im NIFTI-Format eingelesen. Abbildung 4.11 zeigt das User Interface von 3D Slicer. Die Daten werden auf allen drei Schnittebenen dargestellt und ein bereits ausgewählter Segmentierungs-Editor bietet die Möglichkeit, verschiedene Algorithmen anzuwenden.

Zunächst wurde der Schwellwertbereich von 639 - 1251 gewählt (Abbildung 4.12). 3D Slicer stellt verschiedene Label bereit, die einem Namen und einer Farbe zugeordnet sind. Für alle Voxel, die sich im Schwellwertbereich befinden, wird dann ein solches Label gesetzt. Wie Abbildung 4.12 zeigt, wurde dem aktuell segmentierten Objekt die Farbe dunkelgrün und der Name „tissue“ zugewiesen. Da in der Liste kein Label für die Cochlea vorhanden ist, soll der Labelname nicht weiter berücksichtigt werden.

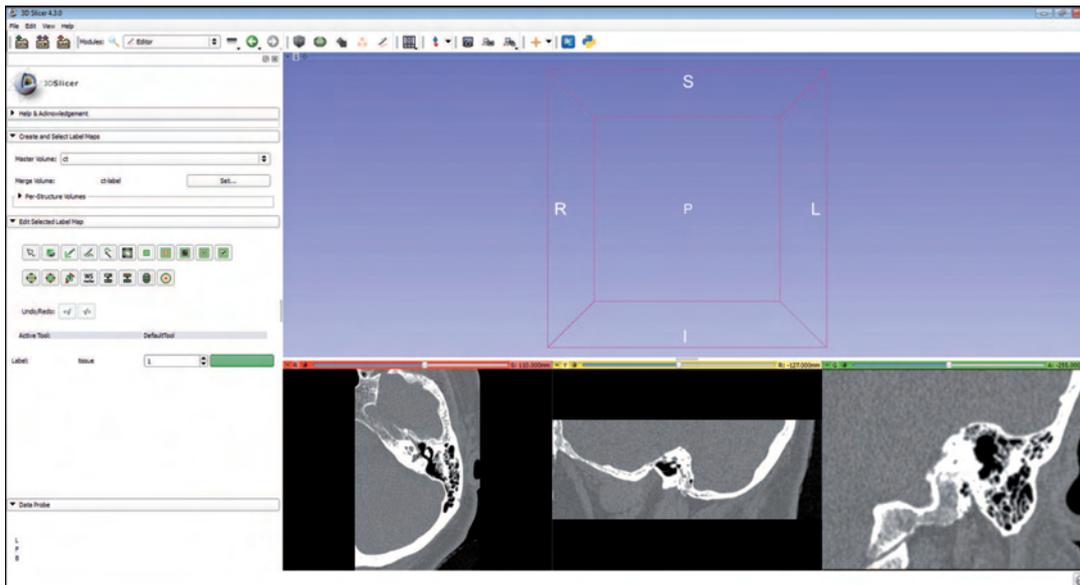


Abbildung 4.11: *3D Slicer User Interface*

Anschließend wurde die Methode „Erode“ ausgewählt, in dem eine Nachbarschaft (4er- oder 8er-Nachbarschaft) sowie ein Label ausgewählt werden kann, auf das die Methode angewandt wird. Für die Nachbarschaft wurde aufgrund des geringen Volumens der Cochlea die 4er-Nachbarschaft gewählt. Die Schaltfläche sowie das Ergebnis der Erosion ist in Abbildung 4.13 zu sehen.

Über die Methode „Identify Islands“ wird eine Connected-Components Analyse durchgeführt und den segmentierten Strukturen jeweils ein Label zugewiesen (Abbildung 4.14). Über „Safe Islands“ kann ein Objekt ausgewählt werden, welches gespeichert werden soll, während die anderen Objekte wieder verworfen werden. (Abbildung 4.15).

Anschließend wurde eine Dilatation durchgeführt. Da das Objekt durch die „Identify Island“-Methode ein anderes Label erhalten hat, muss dieses für die Anwendung der Dilatation entsprechend angepasst werden (Abbildung 4.16). Da für die Dilatation keine Einschränkung auf den Schwellwertbereich vorhanden ist, wurde, wie schon für die Erosion, eine 4er-Nachbarschaft gewählt. Dies entspricht einem „Opening“ und gewährleistet, dass keine Voxel gesetzt werden, die nicht zum Schwellwertbereich gehören.

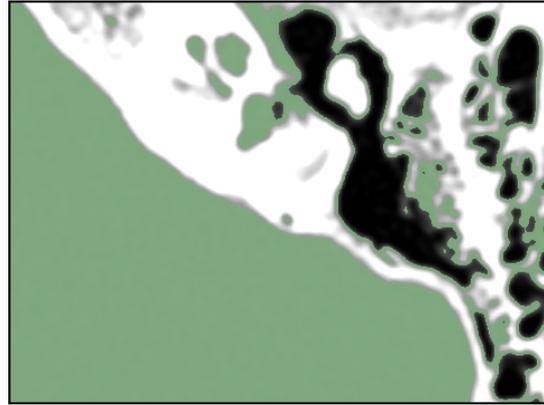
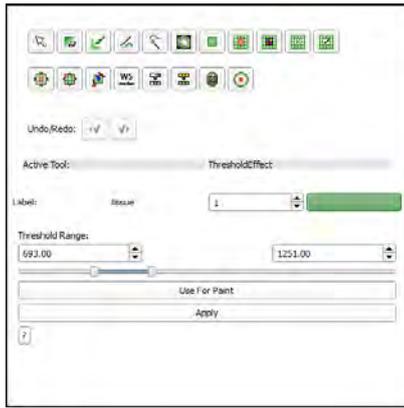


Abbildung 4.12: Schaltfläche und Ergebnis der Schwellwertsetzung

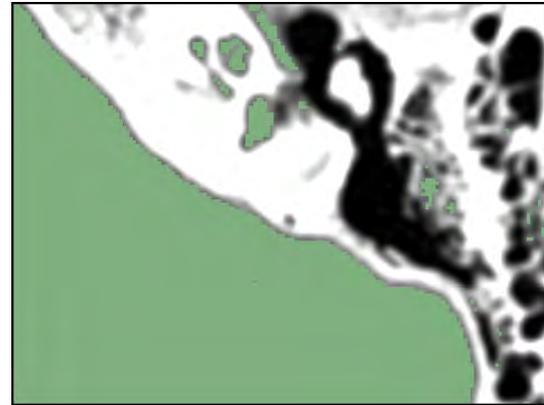


Abbildung 4.13: Schaltfläche und Ergebnis der Erosion



Abbildung 4.14: Die „Identify Islands“-Methode (Connected-Components) vergibt zusammenhängenden Voxeln ein gemeinsames Label und stellt sie durch Verwendung einer LUT farblich dar.

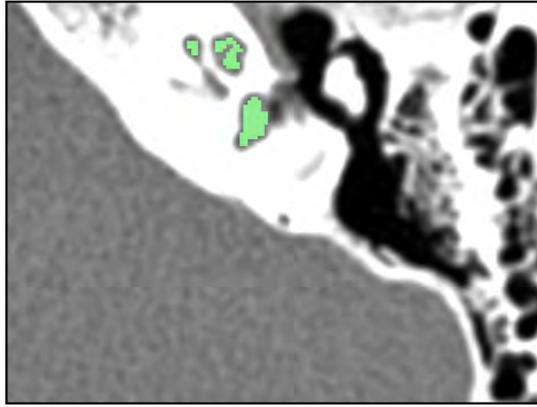


Abbildung 4.15: *Nach Auswählen der Cochlea im „Safe Islands“-Modus werden alle anderen Objekte gelöscht*

Über die Schaltfläche „Split Merge Volume“ wird das segmentierte Objekt in die Liste übernommen und 3-dimensional visualisiert (Abbildung 4.17). Das Label könnte an dieser Stelle noch mit einem anderen Label getauscht werden, um den entsprechenden Namen zu erhalten. Da für die Cochlea, wie schon erwähnt, kein Label vorhanden ist, wird darauf nicht näher eingegangen. In Abbildung 4.17 ist deutlich zu erkennen, dass, wie schon bei MeVisLab, nicht nur die Cochlea, sondern auch ein Teil der Bogengänge segmentiert wurde. Dies ist erneut darauf zurückzuführen, dass die Verbindung zwischen den beiden Objekten zu dick ist, um durch eine Erosion voneinander getrennt zu werden.

Für eine manuelle Trennung bietet 3D Slicer die Schaltfläche „Erase Label“. Nach Aktivierung dieser Schaltfläche kann der Benutzer auf einer der drei orthogonalen Schichten mit gedrückter Maustaste alle Voxel auswählen, die wieder auf das Hintergrundlabel zurückgesetzt werden sollen. Dieses Verfahren wurde auf mehreren Schichten angewandt, bis alle Teile der Bogengänge entfernt werden konnten. Um das Ergebnis in 3D zu visualisieren, wurde dann die Schaltfläche „Merge and Build“ erneut aktiviert. Abbildung 4.19 zeigt das abschließende Ergebnis der Segmentierung. Die Cochlea wurde vollständig segmentiert und die Bogengänge abgetrennt.

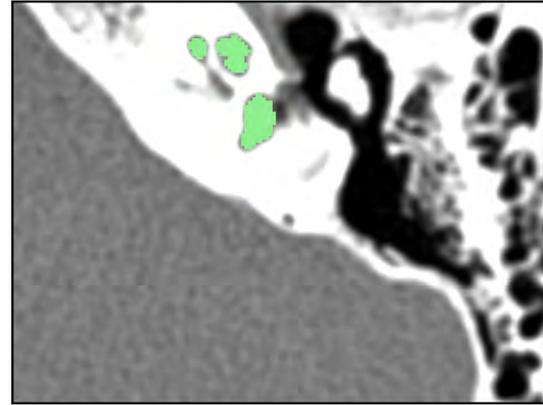
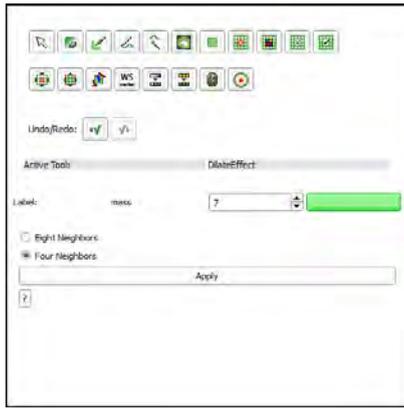


Abbildung 4.16: Über eine Dilatation werden erodierte Voxel wiederhergestellt.

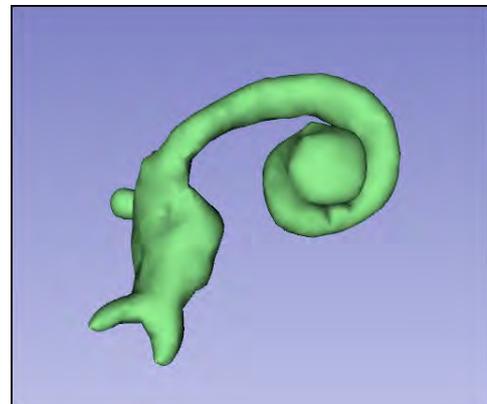
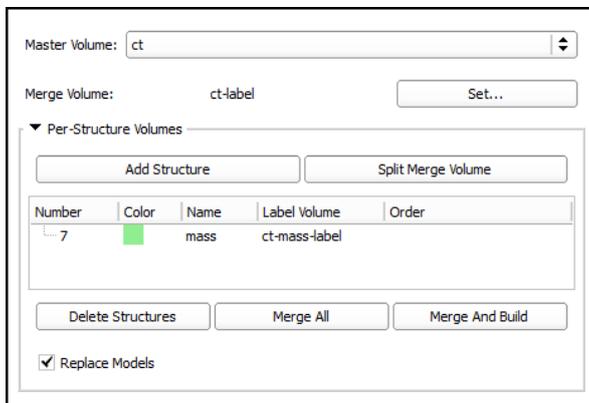


Abbildung 4.17: Labelliste (links) sowie 3D-Visualisierung des segmentierten Objekts (rechts)

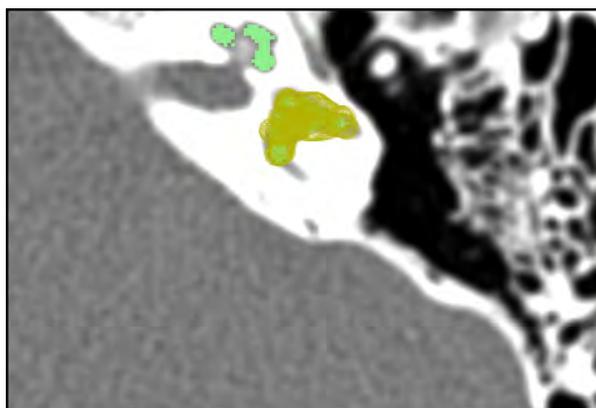


Abbildung 4.18: Auswahl der zu löschenden Voxel einer Schicht nach Aktivierung der Methode „Erase Label“

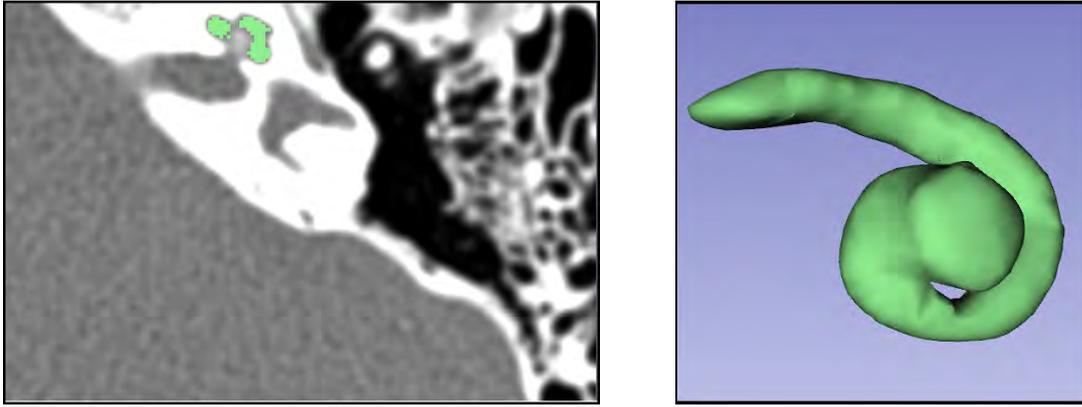


Abbildung 4.19: Endergebnis der segmentierten Cochlea auf einer der orthogonalen Schichten (links) sowie in der 3D Visualisierung (rechts). Die unter Verwendung der „Erase Label“-Methode ausgewählten Voxel wurden gelöscht, wodurch die Bogengänge verschwunden sind.

4.2.2 Ergebnisse

Die Untersuchung von 3D Slicer hat gezeigt, dass eine Segmentierung auf Schwellwertbasis durchgeführt werden kann. Zudem wird sowohl eine Trennung über morphologische Operatoren, als auch eine manuelle Trennung ermöglicht. Durch die begrenzt möglichen Algorithmen zur Segmentierung ist es einem Benutzer möglich, sich schnell in die Software einzuarbeiten. Auch die Segmentierung kleiner Strukturen ist umsetzbar, wie am Beispiel der Cochlea gezeigt wurde.

Eine Kenntnis über Algorithmen wie Erosion oder Dilatation ist allerdings, wie auch schon bei MeVisLab, Voraussetzung für die Bedienung. Zudem kann die 3D-Ansicht während der Schwellwertanpassung nicht in Echtzeit realisiert werden, was das Einstellen eines geeigneten Schwellwerts sehr schwierig macht. Des Weiteren sind nicht alle Strukturen des menschlichen Körpers in der Labelliste vorhanden, sodass für die Cochlea ein anderer Name gewählt werden musste.

Ein weiterer Nachteil von 3D Slicer zeigte sich in der Anwendung der Dilatation. Da sie nicht auf den Schwellwertbereich begrenzt werden kann, ist es möglich, Voxel mit einem Label zu versehen, dessen Intensitätswerte sich nicht im Schwellwertbereich befinden. Dadurch stimmt der Schwellwertbereich nicht mehr mit den Schwellwertgrenzen an den Kanten der segmentierten Cochlea überein, was zu Problemen der Voxel-Man Visualisierung führt. Eine anschließende Maskierung, wie bei MeVisLab in Kapitel 4.1 erläutert, könnte die entsprechenden Voxel zwar wieder zurücksetzen, ist im Segmentierungs-Editor jedoch nicht vorhanden. Zusätzlich zu der Kenntnis über morphologische Operatoren muss dem Benutzer

also bewusst sein, dass der Kernel der Dilatation den der Erosion nicht übersteigen darf. Auch wenn dieser Umstand berücksichtigt wird, können nicht alle Voxel wiederhergestellt werden, da die beiden Operationen nicht invers zueinander sind.

Bei 3D Slicer gab es keine Möglichkeit, die Schwellwerte zu speichern. Schon der Import der MeVisLab-Daten zeigte, dass eine Visualisierung mit nachträglich ermitteltem Schwellwertbereich zu einem guten Ergebnis führte. Da keines der von 3D Slicer angebotenen Formate mit den existierenden Importern des Voxel-Man Simulators korrespondiert, wurde auf einen Import an dieser Stelle verzichtet. Durch das Ergebnis der MeVisLab-Daten wird angenommen, dass auch diese schwellwertbasierte Segmentierung zu einer glatten Oberfläche führt, wenn für die Dilatation dieselben Parameter wie für die Erosion eingestellt werden.

4.3 ITK Snap

Bei ITK Snap handelt es sich um eine Software-Applikation, die speziell für die 3D-Segmentierung medizinischer Bilddaten entwickelt wurde. Es bietet sowohl semi-automatische Segmentierung über Active Contour Methoden als auch manuelle Delinearisierung und Bild Navigation an. Seit den 1990er Jahren beschäftigen sich diverse Universitäten der Vereinigten Staaten mit der Weiterentwicklung von ITK Snap. An der Entwicklung der letzten veröffentlichten Version ITK Snap 2.X war die University of Pennsylvania und die University of Utah beteiligt. Unterstützt wurde es durch das U.S National Institute of Biomedical Imaging and BioEngineering und NIH Blueprint for Neuroscience Research [ITK Snap 2014]. In der vorliegenden Arbeit wurde mit der Version ITK Snap 2.4.2 gearbeitet.

4.3.1 Material und Methoden

Die Bilddaten wurden im NIFTI-Format importiert und zunächst in transversaler, sagittaler und koronarer Schmittichtung dargestellt (Abbildung 4.20). Wie die Abbildung weiterhin zeigt, wurde die zu segmentierende Region durch ein Snake Region of Interest (ROI)-Tool eingegrenzt.

Nachdem ein Label mit zugehöriger Farbe ausgewählt wurde, gelangt der Anwender über die Schaltfläche „Segment 3D“ in den Segmentierungsmodus, in

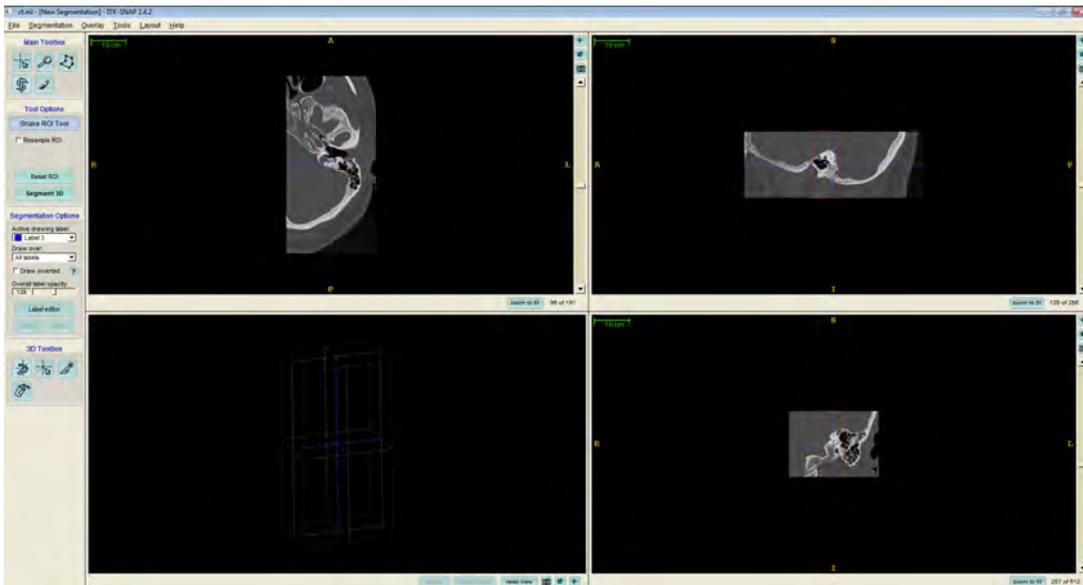


Abbildung 4.20: *ITK Snap User Interface. Es wurde bereits ein ROI um die zu segmentierende Region gesetzt.*

dem er zwischen „Intensity regions“ und „Image edges“ wählen kann. Da für den Voxel-Man Simulator die Schwellwertsegmentierung von besonderem Interesse ist, wird hier nur auf diese Methode näher eingegangen. Abbildung 4.21 zeigt das Segmentierungsfenster, sowie die Schaltfläche zur Schwellwerteinstellung, die sich nach Auswahl der Schaltfläche „Preprocess Image“ öffnet. Es wurde, wie bei den vorherigen Segmentierungsapplikationen, ein Schwellwertbereich von 693 - 1251 gewählt. Das Ergebnis der Schwellwertsetzung zeigt Abbildung 4.22.

Nachdem die Schwellwerte eingestellt sind, kann ein Region Growing durchgeführt werden. Dazu müssen Startpunkte in Form einer Kugel (hier: „Bubbles“) in die Schwellwertregion gesetzt werden (Abbildung 4.23). Die Größe der „Bubbles“ können entsprechend der Strukturgröße angepasst werden. Der Anwender hat die Möglichkeit, die Schritte des Region Growings sowohl manuell als auch automatisch über eine „Play“-Schaltfläche auszuführen. Die Anzahl der Iterationen wird dabei immer angezeigt (Abbildung 4.24). ITK Snap bietet außerdem die Möglichkeit einzelne Iterationen wieder rückgängig zu machen. Über die Schaltfläche „Update mesh“ wird das segmentierte Objekt 3-dimensional dargestellt (Abbildung 4.25). Zudem kann die Farbe sowie der Name des Objekts geändert werden. Dafür wird ein Label Editor bereitgestellt.

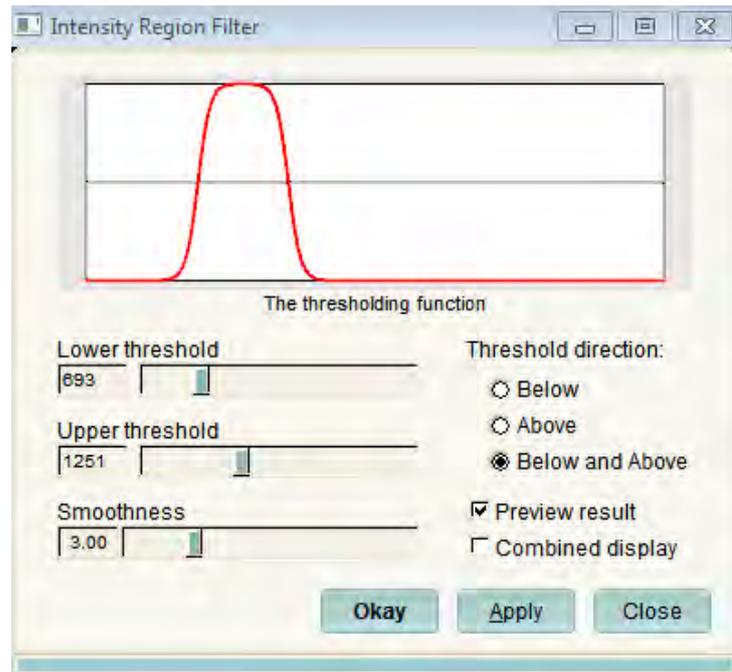


Abbildung 4.21: Segmentierungsfenster (links) sowie die Schaltfläche zur Einstellung des Schwellwerts (rechts)

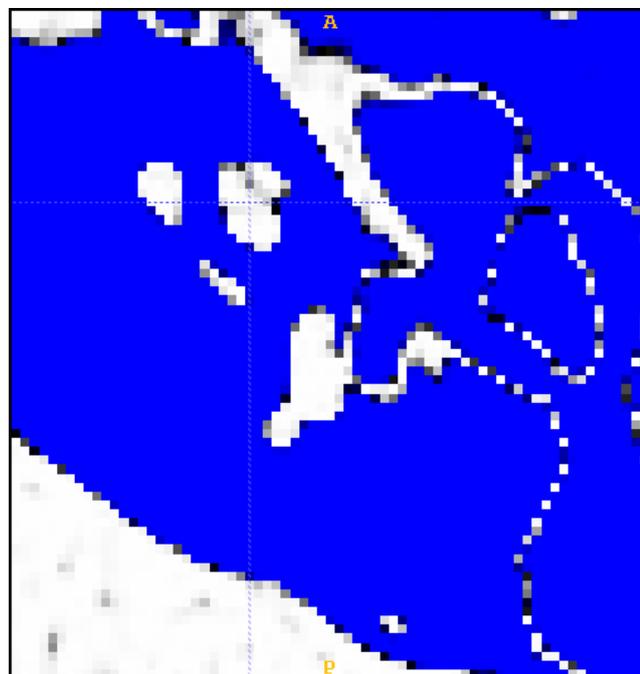


Abbildung 4.22: Ergebnis der Schwellwertsetzung

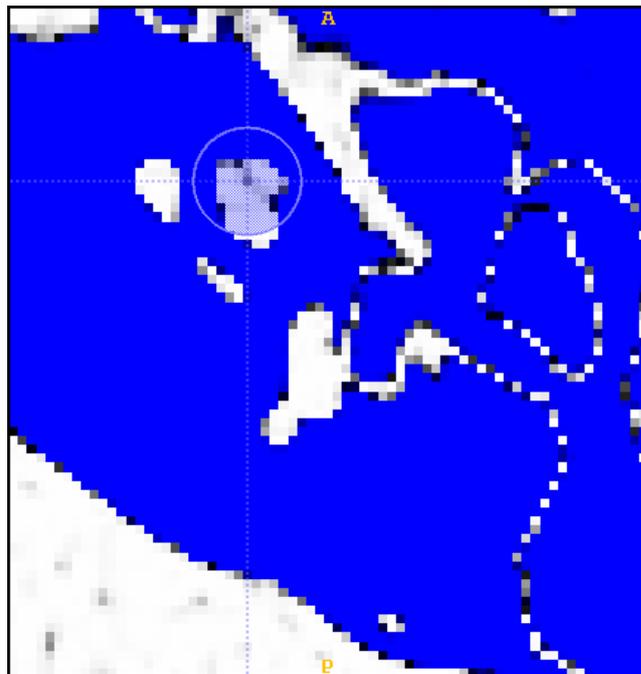
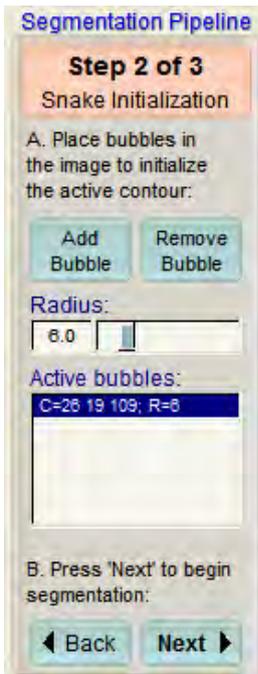


Abbildung 4.23: Segmentierungsfenster und Setzen eines „Bubbles“ auf einer der Schnittebenen

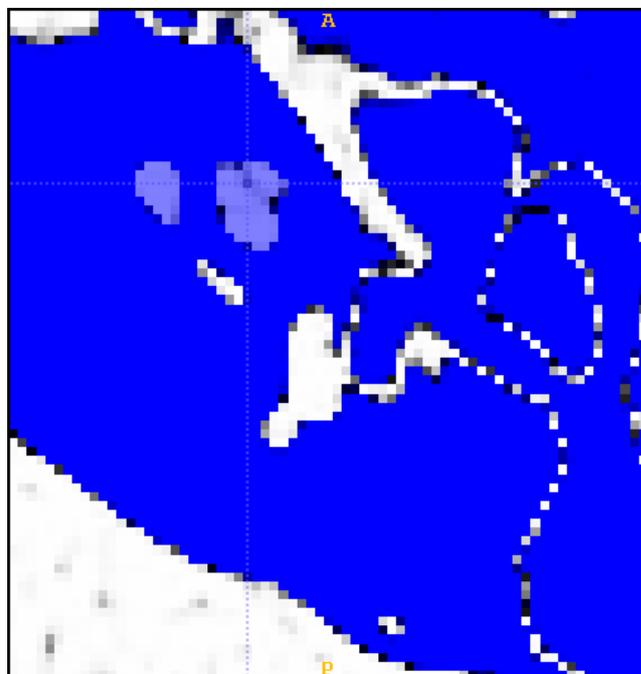
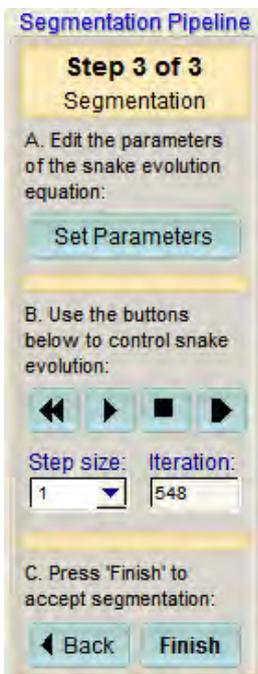


Abbildung 4.24: Segmentierungsfenster und Ergebnis des Region Growings

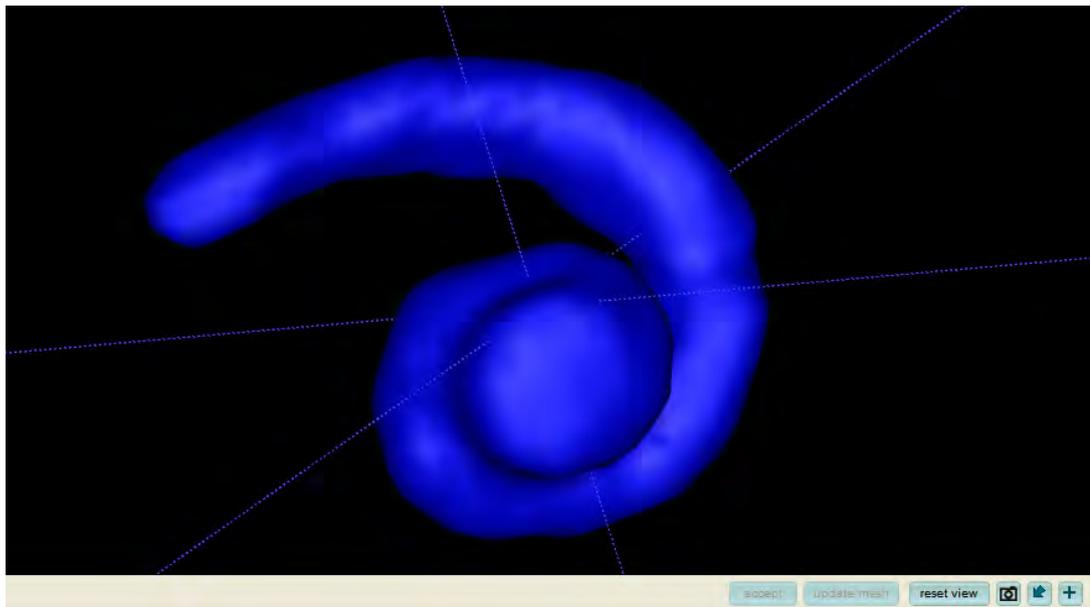


Abbildung 4.25: 3D Visualisierung der segmentierten Cochlea

4.3.2 Ergebnisse

Wie Kapitel 4.3.1 zeigt, konnte die Cochlea auch mit ITK Snap segmentiert werden. Durch die Möglichkeit, die Iterationen manuell zu stoppen, zu erhöhen und zu reduzieren, war eine Trennung in dem präsentierten Beispiel nicht erforderlich.

Durch die Unterteilung des Segmentierungsprozesses in drei Schritte wird eine intuitive Bedienung gewährleistet, wodurch eine zeitaufwändige Einarbeitung in die Software nicht notwendig ist.

Ein Nachteil dieser vorgegebenen Schritte ist die damit einhergehende Inflexibilität. Stellt der Benutzer beispielsweise nach Ausführung des Region Growing fest, dass der Schwellwertbereich nicht ideal gewählt wurde, muss er wieder mit Schritt 1 beginnen. Eine Echtzeitvisualisierung bei Schwellwertänderung ist also nicht nur ausgeschlossen, sondern der Benutzer muss die Segmentierung von vorn beginnen.

Zudem sind keine Operatoren zur Trennung von Objekten vorhanden. Zwar konnte die Cochlea durch Verwendung eines sich als geeignet erwiesenen Schwellwertbereichs auch ohne Trennoperatoren segmentiert werden, jedoch sind diese im Falle eines überlappenden Schwellwertbereichs mit einem angrenzenden Objekt zwingend notwendig.

Ein weiterer Nachteil der Software zeigte sich in der Startpunktsetzung für das Region Growing. So war es bei der Segmentierung der Cochlea nach Setzen der „Bubbles“ teilweise nicht möglich, die Region wachsen zu lassen. Die Iterationen wurden zwar hochgezählt, es fand jedoch kein Region Growing statt. Besonders fiel dies bei kleinen „Bubbles“ auf, weshalb im oben präsentierten Beispiel ein im Vergleich zur Cochlea großer Startpunkt gewählt wurde.

Eine weitere Schwierigkeit ergab sich durch die 3-Dimensionalität der „Bubbles“. Da während des Setzen der „Bubbles“ nur eine Schnittebene angezeigt wird, ist nicht ersichtlich, ob auf anderen Schichten Voxel angrenzender Objekte gesetzt werden, deren Intensitäten sich ebenfalls im Schwellwertbereich befinden. Auch haben nicht alle durch die Schwellwertsetzung markierten Voxel durch das Region Growing ein Label erhalten. Abbildung 4.26 (rechts) zeigt dieses Problem am Beispiel der Cochlea. Die weiß dargestellten Voxel in der markierten Region befinden sich im Schwellwertbereich, haben jedoch kein Label erhalten. Der Grund für dieses Problem war nicht ersichtlich.

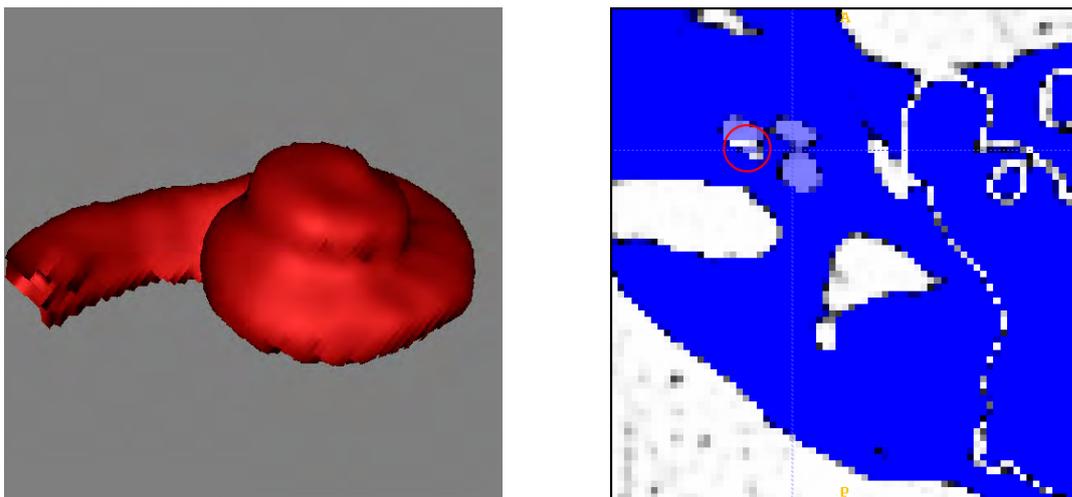


Abbildung 4.26: Die linke Abbildung zeigt die Voxel-Man Visualisierung der mit ITK Snap segmentierten Cochlea. Die unebene Oberfläche könnte mit dem in der rechten Abbildung gezeigten Problem zu erklären sein. Die durch die Schwellwerteinstellung markierten Voxel (weiß) wurden von dem Region Growing (hellblau) nicht erreicht.

Zudem kam es zu Problemen bei der 3D Visualisierung. Wurde die Snake-ROI zu groß gewählt, kam es zum Programmabsturz. Grund dafür könnte die polygonale Darstellung sein, die bei größeren Strukturen mit einem enormen Rechenaufwand verbunden ist.

Auch eine Speicherung der eingestellten Schwellwerte konnte nicht durchgeführt werden. Daher wurde der Schwellwertbereich für die Visualisierung im Voxel-Man System nach dem in Kapitel 4.1.2 erläuterten Verfahren nachträglich ermittelt. Abbildung 4.26 (links) zeigt die Voxel-Man Visualisierung mit dem ermittelten Schwellwertbereich 733 - 1246. Vergleicht man diesen mit dem Schwellwertbereich aus den MeVisLab-Daten ermittelten Schwellwertbereich 689 - 1246, bei dem auch über ein Region Growing auf Schwellwertbasis segmentiert wurde, fällt ein deutlicher Unterschied des unteren Schwellwerts auf. Dies könnte auf das eben erläuterte Problem zurückzuführen sein, dass Voxel, die zum Schwellwertbereich gehören, vom Region Growing nicht erreicht werden.

4.4 MIPAV

Medical Image Processing, Analysis and Visualization (MIPAV) wurde vom Center of Information Technology und den National Institutes of Health in Bethesda mit dem Ziel entwickelt, Methoden und Algorithmen für die medizinische Bild-datenanalyse zu Verfügung zu stellen. Es wurde in Java geschrieben und ist für Windows, UNIX und Macintosh OS X verfügbar [MIPAV 2014]. In dieser Arbeit wurde MIPAV in der Version 7.0.1 für Windows verwendet.

4.4.1 Material und Methoden

Abbildung 4.27 zeigt das User Interface von MIPAV. Nachdem die Daten im NIFTI-Format eingelesen wurden, werden die Schnittbilder in transversaler Richtung dargestellt. Anschließend wurde ein Volume-of-Interest (VOI) um die Cochlea erstellt (Abbildung 4.28), um den Bereich für die anschließende Verwendung eines globalen Schwellwerts einzuschränken. Für die Anwendung des Schwellwertbereichs (693 - 1251) muss dafür „VOI regions“ ausgewählt werden. Zudem wurde eine binäre Darstellung gewählt, um der Region ein Label zuzuweisen (Abbildung 4.29 (links)). Rechts in Abbildung 4.29 ist das Ergebnis der Schwellwertsetzung zu sehen.

Als nächstes wurde eine Erosion durchgeführt. Aufgrund des geringen Volumens der Cochlea wurde die kleinste zur Auswahl stehende Nachbarschaft gewählt (6er-Nachbarschaft). Abbildung 4.30 zeigt die Einstellungen (links) sowie das Ergebnis der Erosion (rechts).

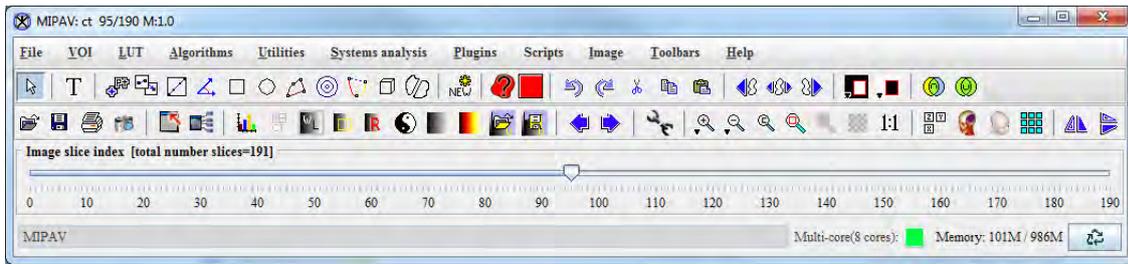


Abbildung 4.27: *MIPAV User Interface*

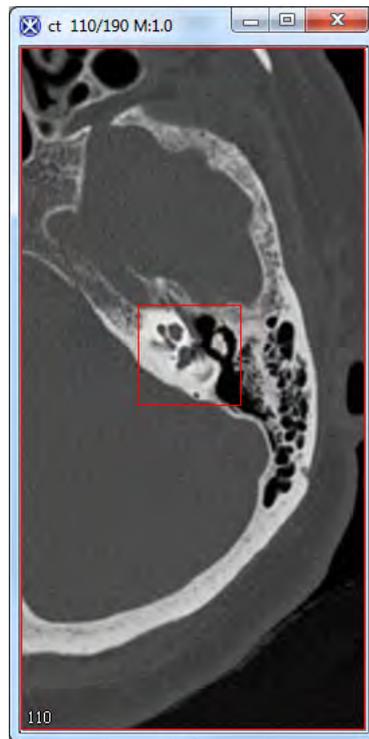


Abbildung 4.28: *VOI um die zu segmentierende Region*

Um eine Connected-Components Analyse durchzuführen, bietet MIPAV das Modul „Identify Objects“ an. Als minimale Anzahl von Voxeln, aus denen ein Objekt besteht, wurde der Wert 20 eingetragen, um auch kleine Strukturen, wie die Cochlea segmentieren zu können (Abbildung 4.31 (links)). Zudem wurde eine LUT angewendet, um die verschiedenen Objekte in unterschiedlichen Farben darzustellen (Abbildung 4.31 (rechts)).

Um alle Objekte bis auf die Cochlea zu eliminieren, wurde das „Threshold“-Modul verwendet. Wie schon bei MeVisLab wurde dabei für den minimalen als

auch den maximalen Schwellwert der Labelwert der Cochlea eingetragen. Das Ergebnis ist in Abbildung 4.32 zu sehen.

Danach wurden 2 Dilatationen einer 6er-Nachbarschaft durchgeführt, um die erodierten Voxel wiederherzustellen (Abbildung 4.33). Die Begrenzung der Dilatation auf den Schwellwertbereich musste auch hier durch eine anschließende Maskierung mit dem ursprünglichen Schwellwertvolumen realisiert werden. Dafür wurde ein „Image Calculator“ verwendet, bei dem beide Volumina sowie der Operator „AND“ ausgewählt wurden (Abbildung 4.34).

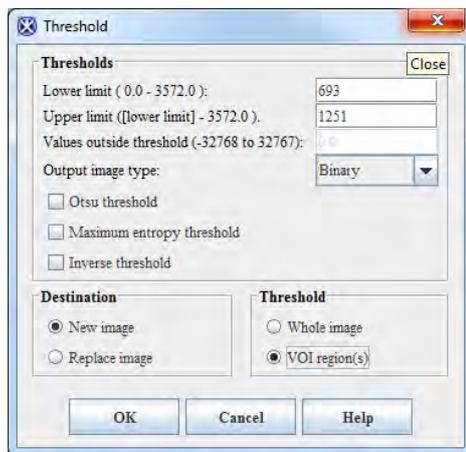


Abbildung 4.29: Schaltfläche und Ergebnis der Schwellwertsetzung

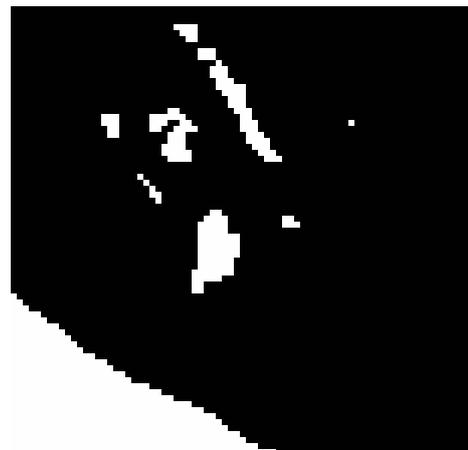
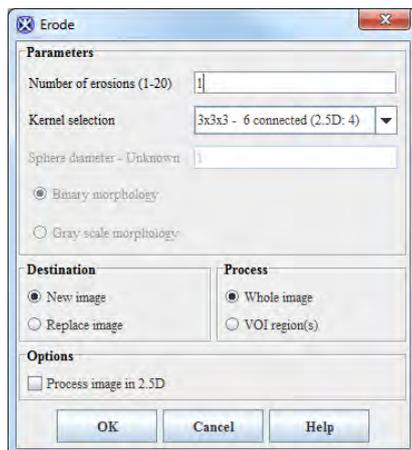


Abbildung 4.30: Schaltfläche und Ergebnis der Erosion

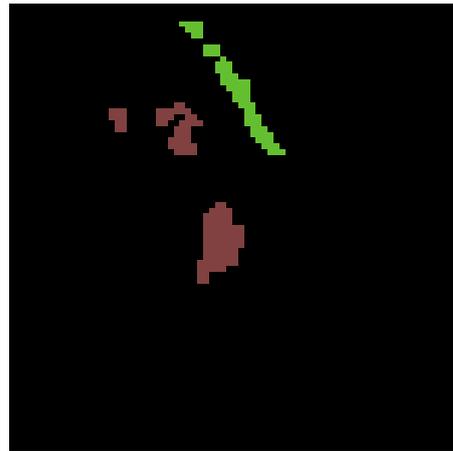
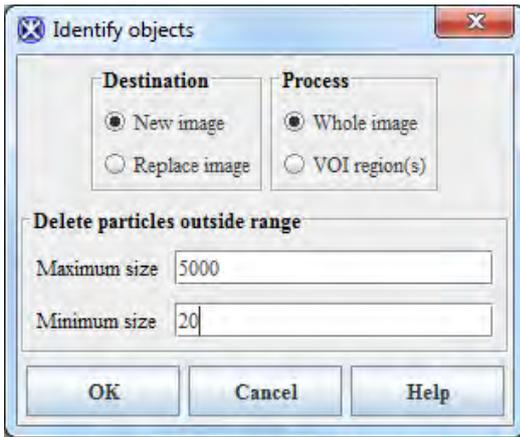


Abbildung 4.31: Schaltfläche der Connected-Components Analyse („Identify Objects“) und das Ergebnis nach Anwendung einer LUT

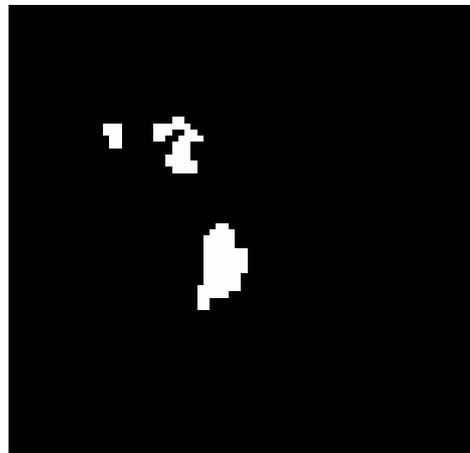


Abbildung 4.32: Ergebnis der Eliminierung anderer Strukturen über eine Schwellwertanalyse

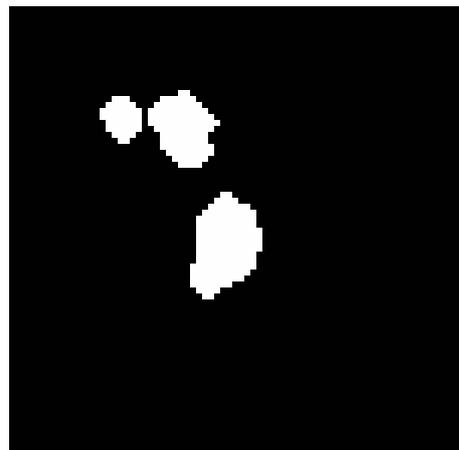
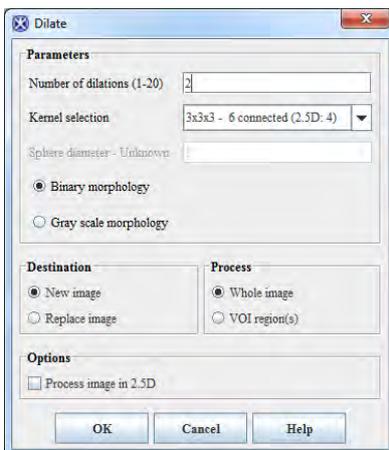


Abbildung 4.33: Schaltfläche und Ergebnis der Dilatation

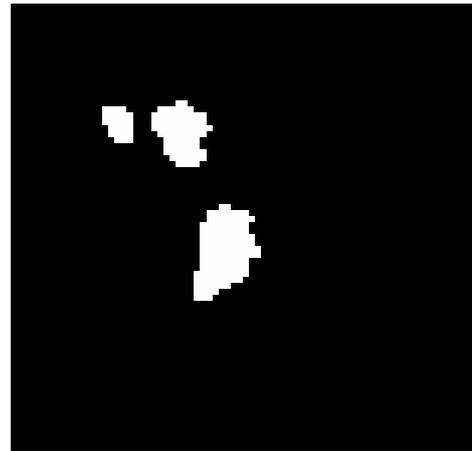
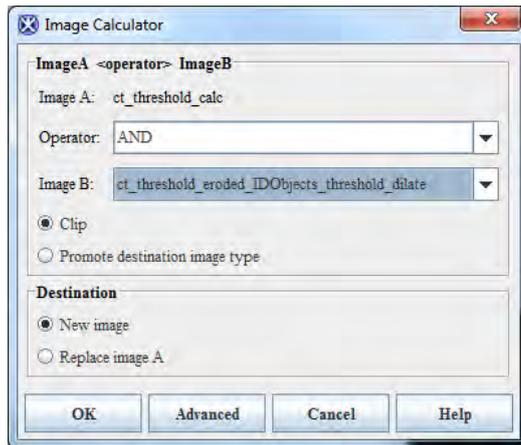


Abbildung 4.34: Schaltfläche und Ergebnis der Maskierung der dilatierten Daten mit dem Ergebnis des globalen Schwellwerts

4.4.2 Ergebnisse

Auch bei der Untersuchung von MIPAV zeigte sich, dass eine schwellwertbasierte Segmentierung der Cochlea möglich ist. Über morphologische Operatoren konnte eine Trennung der Cochlea von anderen Strukturen erzielt werden, wobei die Einschränkung der Dilatation wie bei den anderen Software-Applikationen durch eine Maskierung realisiert werden musste. Durch das Angebot diverser bildverarbeitender Algorithmen ist eine Segmentierung auf unterschiedlichste Weise möglich.

Dies erfordert allerdings gewisse Kenntnisse der Bildverarbeitung, sowie eine Einarbeitung in MIPAV. Zudem konnte das Ergebnis nicht 3-dimensional dargestellt werden, da es bei dem Versuch immer zu einem Programmabsturz kam.

MIPAV bietet kein für den Simulator geeigneten Export der Daten an, weshalb auf einen Import der Daten in das Voxel-Man System verzichtet werden musste.

4.5 Turtle Seg

Turtle Seg ist ein interaktives Segmentierung-Tool, welches am Medical Image Analysis Lab der Fraser University und im Biomedical Signal and Image Computing Laboratory der University of British Columbia entwickelt wurde. Turtle Seg bietet die Möglichkeit über aktive Konturen zu segmentieren. Das bedeutet,

dass der Benutzer eine gewünschte Kurve um das entsprechende Objekt setzt. Die Kurve sucht sich dabei die größten Gradienten in der Umgebung, um die Kante des Objekts möglichst genau zu erreichen. Eine interaktive und intuitive Bedienung stand bei der Entwicklung von Turtle Seg im Vordergrund. Zu diesem Zweck wurde ein automatisches System integriert („Spotlight“), welches den Benutzer durch die Segmentierung führt [Interactive 3D Image Segmentation Software 2014].

4.5.1 Material und Methoden

Abbildung 4.35 zeigt das User Interface nach Einlesen des Datensatzes. Die Daten werden auf allen drei Schnittebenen, sowie in einer 3D-Ansicht (unten rechts) dargestellt.

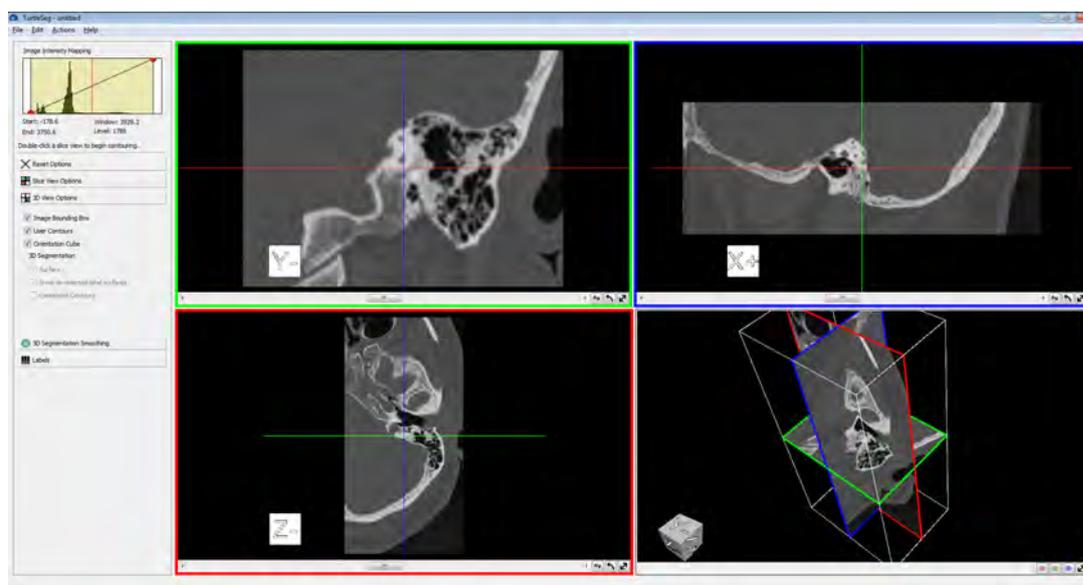


Abbildung 4.35: Turtle Seg User Interface

Durch Doppelklick auf eine der Schnittebenen kann mit der Erstellung einer Kontur begonnen werden. Durch das Setzen mehrerer Teilstrecken, die sich die maximalen Gradienten an den Kanten des Objekts suchen, wurde die Kontur um die Cochlea geschlossen (Abbildung 4.36 (links)).

Über „Actions/Build 3D Segmentation“ wird das Ergebnis der ersten Kontur in der 3D Ansicht dargestellt (Abbildung 4.36 (rechts)). Damit die Cochlea nicht

von den Schnittebenen verdeckt wird, wurden diese aus der 3D-Ansicht entfernt.

Durch Ausführung der Aktion „Suggest Plane“ empfiehlt das System eine Schicht, auf der die nächste Kontur gesetzt werden sollte. In Abbildung 4.37 (links) ist die vorgeschlagene Schicht zu sehen. Die blaue Linie kennzeichnet die Schicht, auf der die erste Kontur gesetzt wurde, die grünen Linien eine von Turtle Seg generierte Ausdehnung in orthogonaler Richtung. Auf dieser Schicht wurde eine zweite Kontur gezogen, die beide Endpunkte der blauen Linie einschließt.

Das Ergebnis der anschließenden 3D-Visualisierung ist in Abbildung 4.37 (rechts) zu sehen. Dieses Verfahren wurde fortgesetzt, bis die Cochlea vollständig segmentiert wurde (Abbildung 4.38). Für die Segmentierung waren insgesamt 8 Konturen nötig.

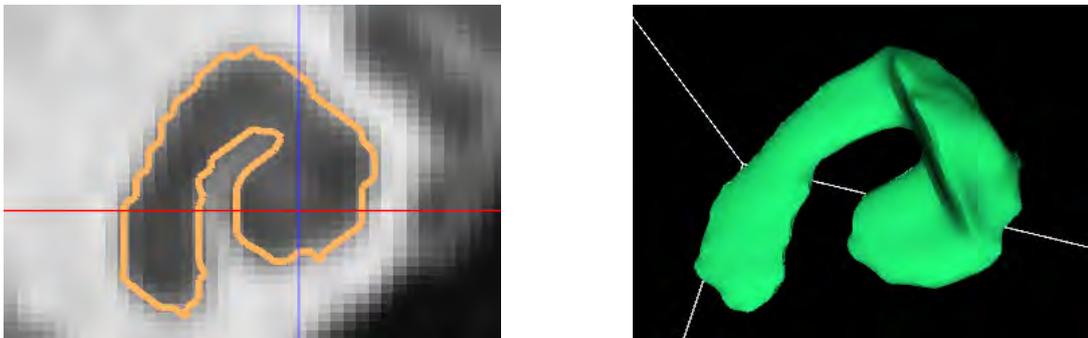


Abbildung 4.36: Erste aktive Kontur und das Ergebnis der 3D Visualisierung

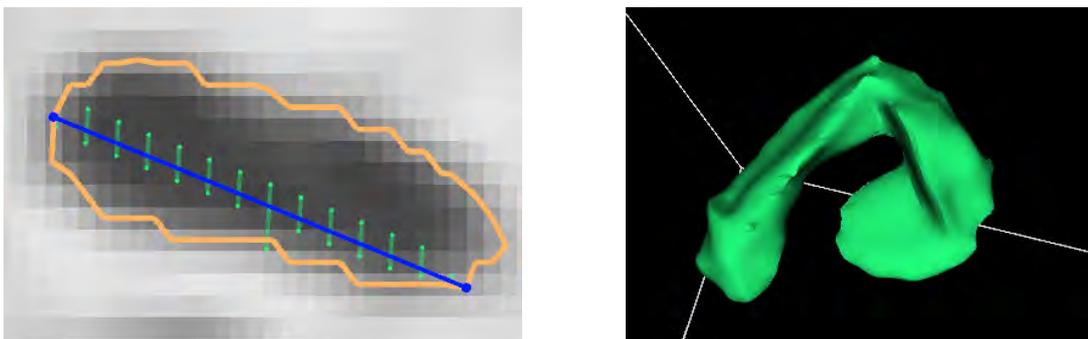


Abbildung 4.37: Zweite aktive Kontur auf einer empfohlenen Schicht und das Ergebnis der 3D Visualisierung. Die blaue Linie kennzeichnet die Schicht, auf der die erste Kontur gesetzt wurde, die grünen Linien eine von Turtle Seg generierte Ausdehnung in orthogonaler Richtung.

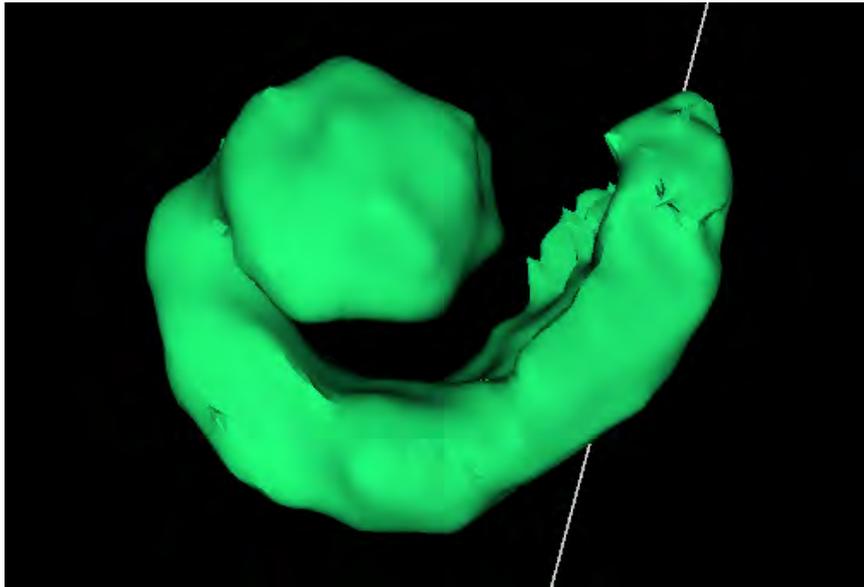


Abbildung 4.38: *3D Visualisierung nach der Erstellung von acht Konturen*

4.5.2 Ergebnisse

Wie schon in Kapitel 4.5.1 erwähnt, wurde bei der Entwicklung von Turtle Seg besonderen Wert auf eine intuitive und interaktive Bedienung gelegt. Ohne die aktive Verwendung von bildverarbeitenden Algorithmen ist der Benutzer schnell in der Lage mit der Software zu arbeiten.

Bei Turtle Seg ist es im Gegensatz zu den zuvor beschriebenen Softwareapplikationen nicht möglich, eine schwellwertbasierte Segmentierung durchzuführen. Ob auch eine Segmentierung der Cochlea über maximale Gradienten zu einer glatten Oberflächendarstellung führt, wurde durch einen Import der Daten in das Voxel-Man System überprüft. Schon die Turtle Seg Visualisierung zeigte kein gutes Ergebnis (Abbildung 4.38). Die Oberfläche ist uneben und besonders der Strang der Cochlea weist scharfe Kanten auf. Auch die Visualisierung im Simulator, welche in Abbildung 4.39 zu sehen ist, führte zu einer unebenen Oberfläche. An den Rändern der Cochlea sind die Kanten einzelner Voxel zu erkennen und der Strang der Cochlea weist Lücken auf. Die nachträgliche Ermittlung der Schwellwerte für die segmentierte Region führte zu den Werten 683 - 2525. Im Vergleich zu dem sonst genutzten Schwellwertbereich von 693 - 1251 ist eine deutliche Vergrößerung zu erkennen. Durch die Diskrepanz

zwischen dem Schwellwertbereich und den tatsächlichen Schwellwertgrenzen an den Kanten der segmentierten Region konnte keine glatte Oberfläche erreicht werden.

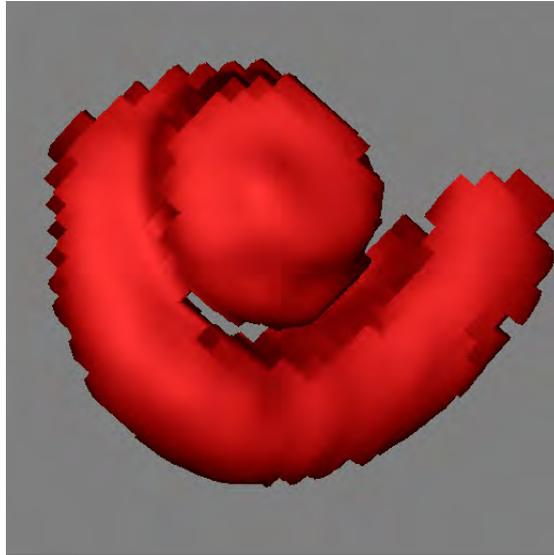


Abbildung 4.39: *Voxel-Man Visualisierung der mit Turtle Seg segmentierten Cochlea*

4.5.3 Weitere Segmentierungsapplikationen

Im Rahmen dieser Arbeit wurden zwei weitere Segmentierungsapplikationen untersucht, die sich jedoch schnell als ungeeignet erwiesen und daher keiner tiefer gehenden Analyse unterzogen wurden. Sowohl bei DeVIDE v12.2.7 als auch bei Eikona3D kam es bereits beim Einlesen der Daten im RAW-Format zu Problemen.

Mit DeVIDE konnten die Daten zwar importiert werden, jedoch wurden sie trotz korrekter Angabe der zum Import erforderlichen Parameter fehlerhaft eingelesen.

Eikona3D bietet beim Import der RAW-Daten lediglich drei verschiedene Datentypen an (unsigned char, float und integer), von denen keines mit dem Datentyp des Beispieldatensatzes (unsigned short) übereinstimmt. Da jedoch auch nach Komprimierung des Datensatzes kein korrektes Einlesen möglich war, liegt die Vermutung nahe, dass die Bytereihenfolge des Datensatzes (Little Endian), welche beim Import nicht eingestellt werden kann, ein Problem für DeVIDE darstellt.

4.6 Schlussfolgerung

Die Untersuchungen haben ergeben, dass keine der beschriebenen Segmentierungsapplikationen alle Anforderungen erfüllt, um als Standardsegmentierer für den Voxel-Man Simulator in Frage zu kommen.

Für MeVisLab, ITK Snap und Turtle Seg wurde eine Möglichkeit geschaffen, die exportierten Daten in den Simulator zu importieren und durch anschließende Schwellwertermittlung zu visualisieren. MeVisLab und ITK Snap zeigten unter Verwendung der Voxel-Man Visualisierung eine glatte Oberfläche. Die im Nachhinein ermittelten Schwellwerte unterschieden sich kaum von den bei der Segmentierung verwendeten. Selbst die fehlende Begrenzung der Dilatation bei ITK Snap veränderte dieses Resultat bei Verwendung desselben Kernel-Wertes für die Erosion und die Dilatation nicht maßgeblich. Das lässt vermuten, dass jeder auf Schwellwerten basierende segmentierte Datensatz auch durch anschließende Schwellwertermittlung im Simulator zu einer glatten Oberfläche führt. Der Import der mit Turtle Seg segmentierten Daten, bei dem mit maximalen Gradienten gearbeitet wird, zeigte eine deutliche Abweichung des Schwellwertbereichs. Dies spiegelte sich auch in der unebenen Oberfläche wider, welche durch eine Diskrepanz zwischen dem Schwellwertbereich und den tatsächlichen Schwellwertgrenzen an der Objektkante erklärt werden kann.

Da sich keines der untersuchten Systeme harmonisch in das Voxel-Man System integrieren ließ, wird im Folgenden die Entwicklung und Implementation eines eigenen Verfahrens für die Segmentierung kleiner Risikostrukturen für den Simulator beschrieben.

5 Eigener Ansatz für die Segmentierung kleiner Risikostrukturen

Kapitel 4 hat gezeigt, dass die untersuchten Software-Applikationen einige Segmentierungsmethoden anbieten, jedoch keine Applikation alle Anforderungen des Voxel-Man Simulators erfüllt. Daher wurde ein eigenes Segmentierungsverfahren entwickelt und implementiert, welches in diesem Kapitel erläutert wird.

5.1 Lösungsansatz

Da die Visualisierung des Simulators eine schwellwertbasierte Segmentierung fordert, ein globaler Schwellwert allerdings ungünstig für kleine Strukturen ist, musste eine lokale schwellwertbasierte Methode gefunden werden. Das Region Growing auf der Basis von Schwellwerten, welches auch in ITK Snap verwendet wird, bietet einen guten Ansatz zur Segmentierung kleiner Strukturen. Zunächst einen globalen Schwellwert einstellen zu müssen, der nach dem Region Growing nicht mehr verändert werden kann, machte das Verfahren allerdings sehr unflexibel. Daher entstand die Idee, den Schwellwertbereich bereits bei der Startpunktsetzung zu ermitteln. Als „Startpunkt“ sollte nicht ein einzelnes Voxel, sondern eine Voxelnachbarschaft dienen, aus dessen Intensitätswerten der Schwellwertbereich bestimmt wird. Im weiteren Verlauf der Arbeit wird einfachheitshalber trotzdem der Begriff „Startpunkt“ für die Anfangsnachbarschaft verwendet. Es kommt zudem dem intuitiven Verhalten des Benutzers entgegen, zunächst den Bereich auszuwählen, den dieser segmentieren möchte. Da es jedoch sehr unwahrscheinlich ist, allein durch das Setzen des Startpunktes einen geeigneten Schwellwert für die Region zu erhalten, sollte außerdem eine Methode zur Schwellwerterweiterung implementiert werden. Auf die Bestimmung des Schwellwerts, wird in Kapitel 5.2.1 noch einmal näher eingegangen, während Kapitel

5.2.2 und 5.2.3 die Implementation der Grow- und Shrink-Methode erläutern. Auch für die manuelle Trennung von Objekten sollte eine Methode implementiert werden (Kapitel 5.2.4).

Für die Trennung von Objekten wurde eine neue Methode entwickelt (Split). Diese sollte auf dem Verfahren der Linux-basierten Voxel-Man Segmentierung basieren, die Algorithmen jedoch vor dem Benutzer verbergen, damit keine Bildverarbeitungskenntnisse vorausgesetzt werden müssen. Ausgelöst durch den Benutzer sollte automatisch eine Trennung herbeigeführt werden. Diese Methode wird in Kapitel 5.3.1 ausführlich erläutert.

5.2 Beschreibung der implementierten Methoden

5.2.1 Setzen eines Startpunktes für das Region Growing

Wie schon in Kapitel 5.1 beschrieben, wurde als „Startpunkt“ eine Voxelnachbarschaft gewählt. Für die Form wurde ein Kreis gewählt, dessen Größe über das Mauselement einstellbar ist. Durch Setzen dieses Kreises auf einer der drei orthogonalen Schnittebenen wird aus den Intensitätswerten aller in dem Kreis befindlichen Voxel ein unterer und oberer Schwellwert ermittelt. Die Wahl der Kreisform ist dadurch begründet, dass die tatsächlichen Startvoxel für das Region Growing gleichmäßig um ein zentrales Voxel angeordnet sind. Die Wahl der 2-dimensionalen Form resultiert aus dem Umstand, dass der Startpunkt auf einer Schnittebene gesetzt wird. Schon ITK Snap (Kapitel 4.3) zeigte die Problematik einer 3-dimensionalen Form. Durch eine Kugel, wäre nicht auszuschließen, dass Voxel in anderen Schichten, die zu diesem Zeitpunkt nicht sichtbar sind, gesetzt werden, die nicht zum zu segmentierenden Objekt gehören. Zudem werden alle Voxel in einer Randvoxelliste gespeichert. Diese Liste besteht zum einen aus den Koordinaten des Randvoxels und zum anderen aus der Iteration, in der das Voxel erzeugt wurde - in diesem Fall 0. Zusätzlich wird ein Zähler mit 0 initialisiert, um die aktuelle Wachstumsiteration zu speichern. Die Speicherung der Iterationen wurde in Folge einer Shrink-Methode ergänzt, die in Kapitel 5.2.3 detailliert erläutert wird.

5.2.2 Grow

Die Methode Grow lässt die Region ausgehend von den Randvoxeln der durch den Kreis gelabelten Voxel um eine Iteration wachsen. Das bedeutet, dass alle Nachbarn der Randvoxel einer 6er-Nachbarschaft gelabelt werden, wenn sich deren Intensitätswerte im Schwellwertbereich befinden. Um die Randvoxelliste zu aktualisieren, wird bei jeder Iteration eine neue Liste erstellt. Wurde ein neues Voxel gesetzt, wird die entsprechende Koordinate mit der um 1 erhöhten Wachstumsiteration in der Randvoxelliste gespeichert. Voxel, die kein neu gelabeltes Voxel in ihrer Nachbarschaft besitzen, bleiben als Randvoxel erhalten. Sie werden mit entsprechender Iteration aus der alten Liste übernommen und in der neuen abgespeichert. Schließlich wird die Gesamtiteration um 1 erhöht.

5.2.3 Shrink

Für die inverse Operation zum Region Growing (Shrink) wurden zunächst alle Randvoxel zurückgesetzt, deren Erzeugung gleich der aktuellen Gesamtiteration ist. Die Wichtigkeit der Speicherung der Erzeugungsiteration wurde in Kapitel 5.2.1 schon erwähnt und ist erforderlich, um eine inverse Operation zum Region Growing zu erzielen. Abbildung 5.1 zeigt eine Region nach acht Wachstumsiterationen. In den Voxeln ist jeweils die Erzeugungsiteration vermerkt. Abbildung 5.2 (links) zeigt das Ergebnis nach Zurücksetzen der Randvoxel ohne Berücksichtigung der Erzeugungsiteration. Es werden Voxel zurückgesetzt, die beispielsweise schon durch die erste Wachstumsiteration entstanden sind. Nur eine Berücksichtigung der Erzeugungsiteration wie in Abbildung 5.2 (rechts) führt zu einer inversen Operation zur Grow-Methode. Lediglich die Voxel, die in der letzten Iteration - in diesem Beispiel in der achten Iteration - erzeugt wurden, werden zurückgesetzt.

Anschließend werden die Nachbarn der zurückgesetzten Voxel auf ihr Label hin untersucht. Besitzt ein Nachbar ein Label, wird er als neues Randvoxel gespeichert. Nun kann ein Voxel aber ein Nachbar mehrerer Randvoxel der letzten Iteration sein. Sieht man sich das Beispiel in Abbildung 5.3 an, die eine weitere Shrink-Iteration darstellt, so würden alle Voxel mit der Erzeugungsiteration sieben zurückgesetzt werden. Bei der Suche nach neuen Randvoxeln würde nun beispielsweise das linke Voxel der Erzeugungsiteration sechs zweimal gespeichert werden, da es sowohl ein Nachbar des linken als auch des oberen Voxels ist. Um zu verhindern, dass in der Randvoxelliste ein Duplikat erstellt wird,

wäre es nötig gewesen, vor jedem Speichern eines neuen Randvoxels die Liste zu durchlaufen und zu testen, ob dieses schon existiert. Da dies mit einem großen Rechenaufwand und somit mit einem zu großen Zeitaufwand verbunden ist, war diese Implementation keine effiziente Lösung. Um das Problem zu umgehen, wurde für alle aktuellen Randvoxel ein Label gesetzt. Werden also nun die Nachbarn der zurückgesetzten Voxel untersucht, werden sie nur als neues Randvoxel gespeichert und werden mit dem Randlabel versehen, wenn sie noch kein Randvoxellabel besitzen.

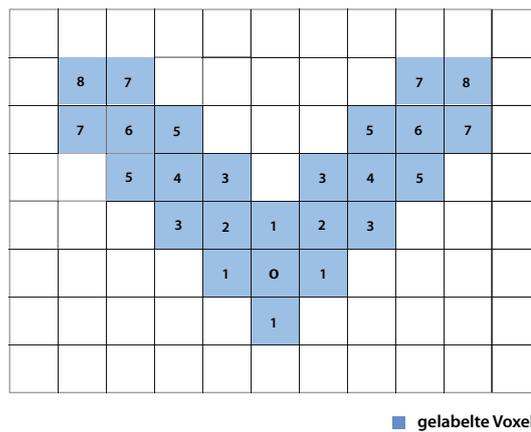


Abbildung 5.1: Region Growing nach acht Iterationen

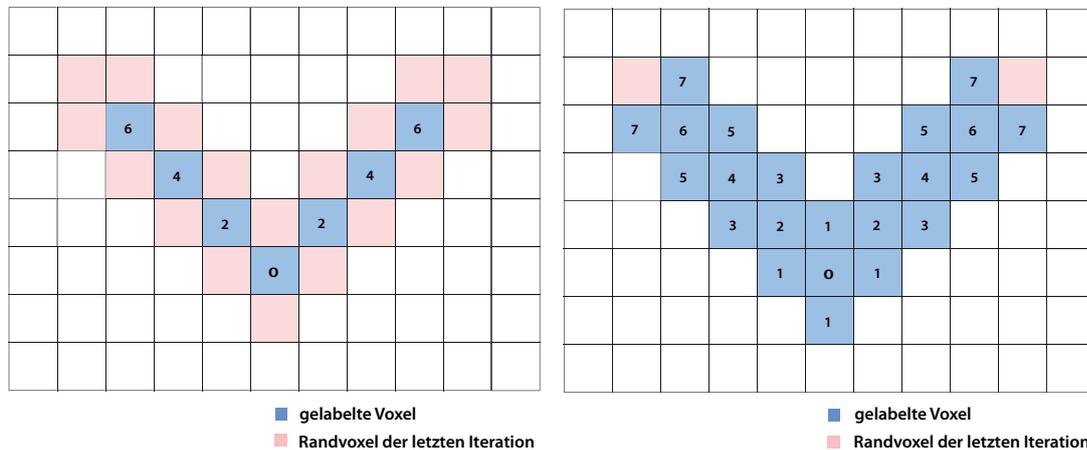


Abbildung 5.2: Zurücksetzen der Randvoxel ohne Berücksichtigung (links) und unter Berücksichtigung (rechts) der Erzeugungsiteration. Nur durch Berücksichtigung der Iteration wird eine inverse Operation zur Grow-Methode erzielt.

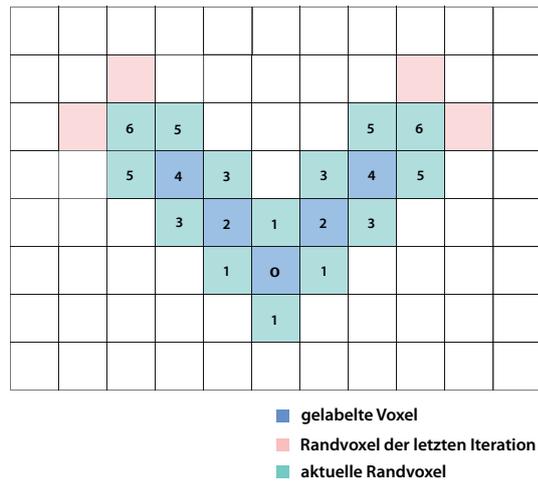


Abbildung 5.3: Zurücksetzen der Randelemente unter Berücksichtigung der Erzeugungssiteration. Durch Markierung der Randelemente wird ein Duplikat in der Randelementliste ausgeschlossen. Das linke Element der Iteration sechs beispielsweise besitzt zwar zwei Nachbarn, welche in der alten Randelementliste vorhanden sind, würde aber beim zweiten Nachbarn nicht noch einmal in der Randelementliste gespeichert werden, da es schon markiert ist.

5.2.4 Manuelle Trennung

Der Benutzer hat die Möglichkeit manuell Voxel zu löschen bzw. Verbindungen manuell aufzutrennen. Alle bei aktivierter Maustaste ausgewählten Voxel werden mit einem Maskenlabel versehen. Durch eine Aktualisierung der Grow-Funktion bis zur Gesamtiteration werden somit Strukturen, die durch maskierte Voxel abgetrennt wurden, nicht mehr erreicht. Durch erneutes Ausführen des Region Growings bis zur Gesamtiteration wird die Randvoxelliste aktualisiert. Für den Fall, dass versehentlich Voxel des zu segmentierenden Objektes gelöscht wurden, wurde eine Undo-Funktion implementiert. Wird diese über eine Schaltfläche ausgewählt, werden die maskierten Voxel angezeigt und können mit aktivierter Maustaste ausgewählt werden. Die ausgewählten Voxel erhalten dann wieder das Objektlabel und das Region Growing wird neu gestartet, um die Randvoxelliste zu aktualisieren.

5.3 Beschreibung der neu entwickelten Methoden

Nachdem in Kapitel 5.2 die Implementation geeigneter bereits bekannter Methoden für die Schwellwertsegmentierung beschrieben wurde, werden in diesem Kapitel Methoden vorgestellt, die speziell für den Voxel-Man Chirurgie-Simulator

entwickelt wurden. Die Entwicklung bezog sich zum einen auf die intuitive Trennung von Objekten. Zum anderen wurde durch ein besonderes Objektmanagement eine Segmentierung über Teilobjekte ermöglicht. Auch die Problematik der durch das Objektmanagement entstandenen manuellen Schwellwertanpassung soll in diesem Kapitel behandelt werden.

5.3.1 Split-Methode

Wie sowohl der Linux-basierte Voxel-Man Segmentierer als auch einige der untersuchten Software-Applikationen zeigten, konnte die Trennung von Objekten nur manuell, beispielsweise durch Schneiden, oder aber unter Verwendung morphologischer Operatoren erfolgen, die eine Kenntnis dieser Algorithmen voraussetzt. Daher wurde im Rahmen dieser Arbeit eine Methode zur semi-automatischen Trennung entwickelt, um eine intuitive Bedienung zu gewährleisten. Die Methode lehnt sich an das in Kapitel 3.1 beschriebene Verfahren an, bei dem über eine Erosion, eine Art Connected-Components und eine Dilatation eine Trennung zweier Strukturen erzielt wird. Zur Veranschaulichung des Algorithmus dient das vereinfachte Flussdiagramm in Abbildung 5.4. Das Verfahren arbeitet wie folgt:

1. Der Benutzer spezifiziert durch Mausklick den Bereich, der nicht zum Objekt gehört. Diese Information wird im Gegensatz zu der Linux-basierten Voxel-Man Segmentierung benötigt, damit bei anschließender Verwendung des Region Growings ein Wachsen in diese Region unterbunden werden kann. Abbildung 5.5 (links) zeigt eine Situation, in der die Methode zum Einsatz kommt. Zwei Objekte ähnlicher Intensität liegen so dicht beieinander, dass auch ein Teil des rechten Objektes segmentiert wurde. Dieser wurde bereits vom Benutzer durch Mausklick spezifiziert.
2. Anschließend wird ein Grow mit zehn Iterationen durchgeführt. (Abbildung 5.5 (rechts)). Dies soll zum einen verhindern, dass ein Voxel, welches am Rand einer Struktur ausgewählt wurde, nach wenigen Erosionen gelöscht wird. Zum anderen soll vermieden werden, dass die gesamte Struktur so klein ist, dass sie nach nur wenigen Erosionen verschwunden ist. Die Anzahl der Iterationen (zehn) richtet sich nach der maximalen Anzahl der Erosionen, die auf sieben begrenzt wurde. Um einer Erosion eines ausgewählten Voxels am Rand der segmentierten Region entgegenzuwirken, muss die Anzahl der Grow-Iterationen in jedem Fall größer als die Anzahl

der Erosionen gewählt werden. Auf die Wahl der zehn Erosionen wird in Punkt 5 noch einmal näher eingegangen.

3. Anschließend wird eine Erosion durchgeführt. Dafür wurden die aktuellen Randvoxel auf 0 zurückgesetzt und die neuen Randvoxel in der Randvozelliste gespeichert. Abbildung 5.6 (links) zeigt, dass die erodierten Voxel durch ein Label gekennzeichnet werden. Diese Kennzeichnung ist für die Begrenzung der Dilatation nötig und wird an späterer Stelle noch näher erläutert.
4. Es wird getestet, ob das ausgewählte Voxel trotz der oben genannten Maßnahme des zehnfachen Growings erodiert wurde. Ist dies der Fall, wird das Ausgangsvolumen wiederhergestellt und eine Meldung ausgegeben, dass die Regionen durch die Split-Methode nicht trennbar sind.
5. Ist das ausgewählte Voxel noch Teil des segmentierten Bereichs, wird getestet, ob zwischen diesem Voxel und dem Startpunkt noch eine Verbindung existiert. Dies erfolgt, indem die Region um das ausgewählte Voxel mit einem anderen Labelvolumen wächst und getestet wird, ob der Startpunkt der Region Growings erreicht wurde. Wurde nach sieben Iterationen keine Trennung der Objekte erzielt, wird das Ausgangsvolumen wiederhergestellt und eine Meldung ausgegeben, dass die Regionen durch diese Methode nicht trennbar sind. Die Anzahl sieben wurde empirisch gewählt und ermöglicht eine Trennung von Verbindungen einer Breite von maximal 14 Voxeln, was für die Risikostrukturen des Innenohrs ausreichend ist.
6. Konnten die Regionen voneinander getrennt werden, wird eine alternierende Dilatation der beiden Regionen durchgeführt, welche durch Verwendung der Grow-Methode realisiert wurde. Auch diese wurde wie die Erosion über die Randvozellisten gelöst. Nachdem für beide Regionen eine Randvozelliste erstellt wurde, wird für ungelabelte Nachbarn dieser Voxel ein Label gesetzt. Dies geschieht jedoch nur, wenn sie zuvor erodiert wurden, was durch das bereits erwähnte Label für erodierte Voxel bekannt ist. Diese Einschränkung gewährleistet, dass die Dilatation nicht über die ursprünglichen Grenzen hinausgeht und somit keine Voxel gesetzt werden, deren Intensitäten sich nicht im Schwellwertbereich befinden. Abbildung 5.6 (rechts) zeigt die Daten nach Beendigung dieses Prozesses. Die nun gelabelte Region um das ausgewählte Voxel wird als negative Maske im Objektvolumen gespeichert. Es ist im weiteren Verlauf der Segmentierung des Objektes nicht mehr möglich, in diese Region zu wachsen.

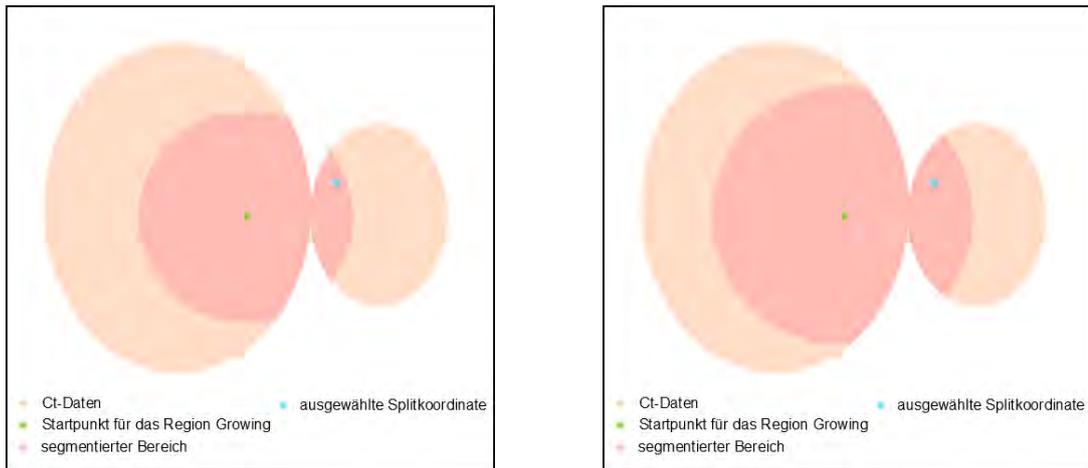


Abbildung 5.5: Links ist die Ausgangssituation der semi-automatischen Trennung zu sehen. Die Region ist in ein anderes Objekt gewachsen und der Benutzer hat diesen Bereich bereits ausgewählt. Die rechte Abbildung zeigt das Ergebnis der Grow-Methode nach zehn weiteren Iterationen.

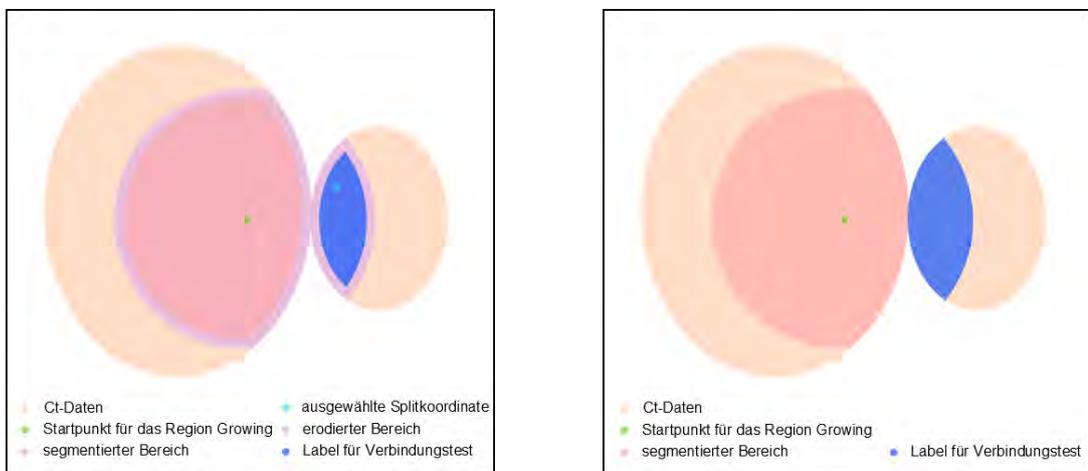


Abbildung 5.6: Eine Erosion wurde soweit durchgeführt, dass die von der ausgewählten Koordinate wachsenden Region (blau) den Startpunkt nicht erreicht (links). Erodierete Elemente wurden mit einem Label (lila) versehen. Die rechte Abbildung zeigt das Ergebnis der Dilatation. Auch die ausgewählte Koordinate wurde mit dem Maskenlabel versehen (blau).

5.3.2 Objektmanagement

Sobald der Benutzer durch eine Schaltfläche die Segmentierung eines Objektes startet, wird ein Objekt angelegt, dem ein Name und eine Farbe zugewiesen werden kann. Zunächst wurde versucht, ein Objekt durch Setzen eines einzigen Startpunktes und Zuweisung eines einzigen Schwellwertbereichs zu segmentieren. Dies führte in einigen Fällen allerdings nicht zu zufriedenstellenden Ergebnissen, was durch ein Beispiel in Abbildung 5.7 veranschaulicht werden soll. Die beige markierten Felder stellen Voxel des zu segmentierenden Objekts dar, die rosa markierten die eines Objektes ähnlicher Intensität, welches an dieses angrenzt. Alle mit einer 1 gekennzeichneten Voxel wurden bereits durch die Segmentierung dem Objekt zugeordnet. Es wird angenommen, dass die beiden blau markierten Voxel denselben Intensitätswert besitzen, der aktuell nicht im Schwellwertbereich liegt. Soll das Objekt nun weiter segmentiert werden, würde auch das markierte rosa Voxel ein Label erhalten und es würde somit ein Objekt in die Segmentierung eingeschlossen, welches nicht zu dem zu segmentierenden Objekt gehört. Durch das Auftreten solcher oder ähnlicher Fälle wurde ein Verfahren entwickelt, welches die Segmentierung eines Objektes aus mehreren Teilobjekten ermöglicht. Es wird also ein weiterer Startpunkt gesetzt und somit ein weiteres Teilobjekt eines anderen Labels (in diesem Beispiel 2) definiert. Unter der Bedingung, dass ein bereits vorhandenes Label nicht überschrieben werden kann, ist es somit möglich, das Objekt zu segmentieren, ohne Voxel des rosa markierten Objektes zu setzen (Abbildung 5.7 (rechts)).

Für die Einstellung des Schwellwerts wurde zunächst versucht, einen Schwellwert für die gesamte Region zu definieren, der durch das Segmentieren eines neuen Teilobjektes angepasst wird. Zu stark schwankende Schwellwertbereiche der einzelnen Teilobjekte hatten allerdings zur Folge, dass der Schwellwertbereich in einigen Fällen nicht mehr mit den Schwellwertgrenzen der Objektkante übereinstimmte, was aufgrund der in Kapitel 3.2 beschriebenen Visualisierung zu einer unsauberen Oberflächendarstellung führt. Selbst wenn ein Schwellwert für das gesamte Objekt vorhanden ist, ist es nicht möglich, den optimalen Schwellwert durch Setzen des Startpunktes und Erweiterung des Schwellwertbereichs zu definieren. Zu diesem Zweck wurde eine manuelle Schwellwertanpassung implementiert, die aktuell jedoch nicht in Echtzeit visualisiert werden kann. Auf die Problematik der Echtzeit Visualisierung bei der manuellen Schwellwertanpassung wird in Kapitel 5.3.3 noch einmal näher eingegangen. Da die Einstellung eines optimalen Schwellwertbereichs bei verzögerter Visualisierung zu einem mühsa-

men und schwierigen Prozess wird, wurde für jedes Teilobjekt ein individueller Schwellwertbereich bestimmt und gespeichert.

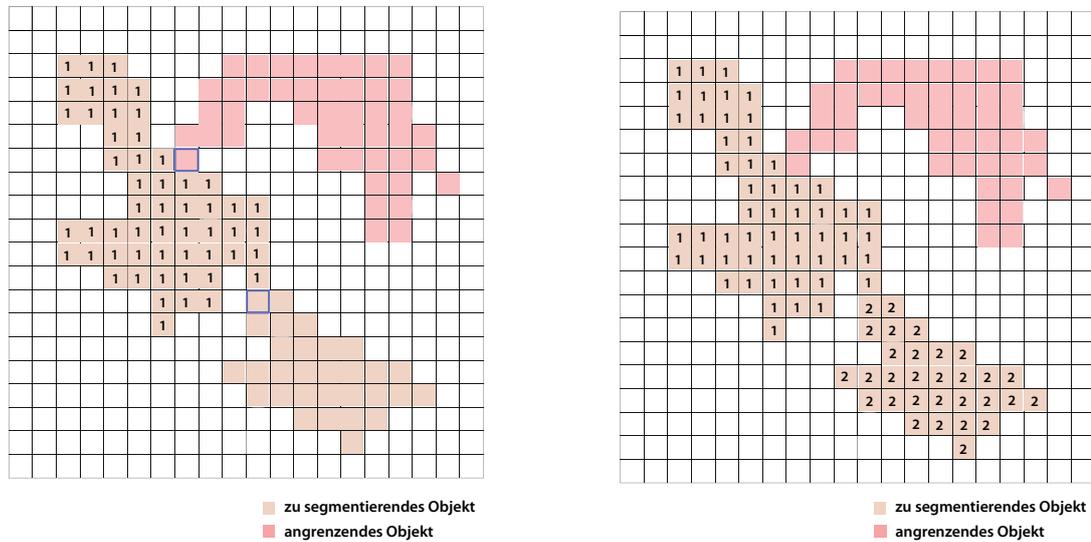
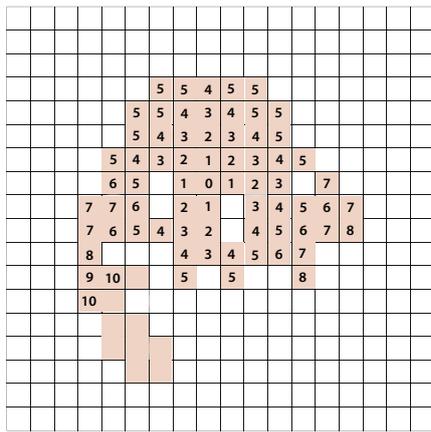


Abbildung 5.7: Links ist das Ergebnis der ersten Teilsegmentierung zu sehen. Die Segmentierung wurde an dieser Stelle beendet, da durch die nächste Wachstumsiteration ein Element des anderen Objekts (rosa) gesetzt würde. Das rechte Bild zeigt das Ergebnis nach Segmentierung eines weiteren Teilobjektes. Durch die Begrenzung des ersten Teilobjektes konnte das gesamte Objekt segmentiert werden, ohne dass ein Element des angrenzenden Objektes gesetzt wurde.

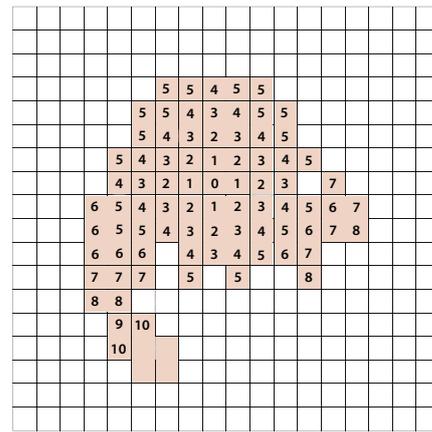
Wurde das gesamte Objekt segmentiert, hat der Benutzer die Möglichkeit, es zu speichern. Dadurch werden alle Teilobjekte mit zugehörigen Schwellwertbereichen gesichert und ihnen der eingestellte Name und die eingestellte Farbe zugewiesen. Somit sind alle Teilobjekte eindeutig einem Objekt zugeordnet und werden im Volumen trotz unterschiedlicher Label durch dieselbe Farbe dargestellt.

5.3.3 Manuelle Schwellwertanpassung

Eine manuelle Schwellwertanpassung wurde im Zuge des Objektmanagements implementiert. Die Schwierigkeit, die sich dabei ergab, war, dass die Schwellwertänderung nicht, wie bei der Segmentierung über einen globalen Schwellwert, in Echtzeit visualisiert werden kann. Bei Verwendung des globalen Schwellwerts wird jedes Voxel des gesamten Volumens daraufhin untersucht, ob seine Intensität im Schwellwertbereich liegt. Bei Verwendung eines Region Growings hingegen bestimmt zusätzlich die Iteration, ob ein Voxel ein Label erhält. Dies soll



■ Intensität im Schwellwertbereich



■ Intensität im Schwellwertbereich

Abbildung 5.8: Region Growing eines Objektes nach zehn Iterationen mit unterschiedlichen Schwellwertbereichen. Es ist deutlich zu sehen, dass eine Schwellwertänderung desselben Objektes und derselben Iteration zu einer unterschiedlichen Größe des segmentierten Bereichs führt.

anhand eines Beispiels veranschaulicht werden. Abbildung 5.8 (links) zeigt ein Region Growing nach der zehnten Iteration. In jedem Voxel ist die Erzeugungsiteration eingetragen. Die farblich markierten Voxel sind Elemente der zu segmentierenden Region, wobei die beige markierten Voxel im Schwellwertbereich und die rosa markierten außerhalb des Schwellwertbereichs liegen. Abbildung 5.8 zeigt ein Region Growing desselben Objektes und derselben Iteration unter Verwendung eines anderen Schwellwerts.

Es ist deutlich zu sehen, dass sich vollkommen andere Pfade ergeben, wodurch ein größerer Teil des Objektes segmentiert wurde. Für eine Schwellwertanpassung bedeutet dies, dass für jeden neu eingestellten Schwellwert das komplette Region Growing neu gestartet werden muss, wenn es anschließend noch korrekt ausgeführt werden soll. Je nach Größe der Struktur bringt das einen erheblichen Zeitaufwand mit sich.

6 Ergebnisse

Nachdem in Kapitel 5 die entwickelten und implementierten Methoden erläutert wurden, werden in diesem Kapitel die Ergebnisse präsentiert. Dafür wurden die Risikostrukturen des Innenohrs (Cochlea, Bogengänge, Gehörknöchelchen und Gesichtsnerv) mit den implementierten Methoden auf dem bekannten CT-Datensatz segmentiert.

6.1 Segmentierung der Risikostrukturen des Innenohrs

Abbildung 6.1 zeigt das Userinterface des implementierten Segmentierers. Die drei orthogonalen Schnittebenen stellen die Grauwertdaten des Beispieldatensatzes und das Fenster zwischen ihnen die 3D-Visualisierung der segmentierten Regionen dar. Das Fenster in der linken unteren Ecke bietet die Möglichkeit einen Namen und eine Farbe für das aktuelle Objekt festzulegen, sowie verschiedene Methoden zur Segmentierung zu verwenden. Der Name „Cochlea“ wurde bereits eingegeben und die Farbe blau ausgewählt. Zu Beginn der Segmentierung ist es lediglich möglich, einen Startpunkt für das Region Growing zu setzen („Set Seed“), sowie den Segmentierungsmodus zu verlassen („Cancel“). Die Schaltfläche „Add Seed“ wurde bereits aktiviert und die Größe des Startpunktes an die Größe der Cochlea, mit der begonnen wurde, angepasst.

In Abbildung 6.2 ist das Ergebnis des Startpunktsetzens zu sehen. Alle im Kreis befindlichen Voxel haben ein Label erhalten und werden sowohl auf allen drei Schnittebenen, als auch in der 3D-Ansicht dargestellt. Aus den Intensitäten der gelabelten Voxel wurde ein Schwellwertbereich ermittelt, der anhand der Slider abzulesen ist. Nach Setzen des Startwertes sind auch die Schaltflächen „Delete Seed“, „Split“, „DeleteVoxel“ und „Define“ aktivierbar. Der Benutzer hat also die Möglichkeit den gesetzten Startpunkt wieder zu löschen, die semi-automatische Trennung über morphologische Operatoren (Split) durchzuführen, sowie Voxel

manuell zu löschen („DeleteVoxel“), sobald eine segmentierte Struktur vorhanden ist.

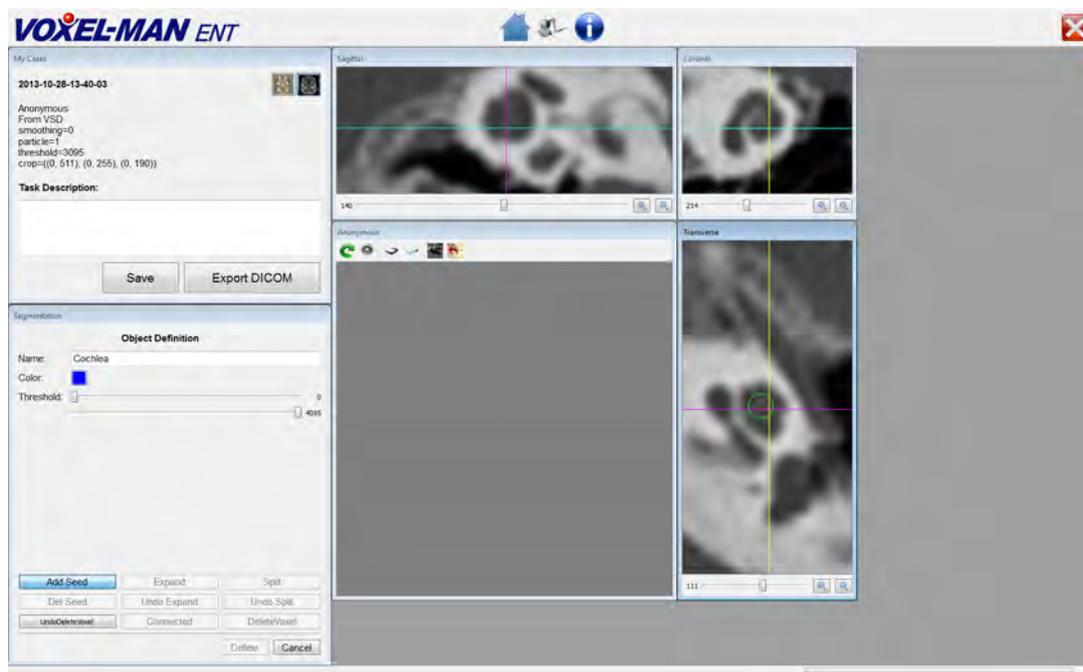


Abbildung 6.1: User Interface des integrierten Voxel-Man Segmentierers. Die Schaltfläche „Set Seed“ wurde bereits aktiviert und die Größe des Startpunktes an die Cochlea angepasst.

Anschließend wurde das Region Growing über das Mausrad ausgeführt. Die transversale Schnittebene in Abbildung 6.3 zeigt, dass nicht alle Voxel der schon segmentierten Teile der Cochlea ein Label erhalten haben, da ihre Intensitätswerte außerhalb des aktuellen Schwellwertbereichs liegen. Um den Schwellwertbereich anzupassen, wurde die Schaltfläche „Expand“ aktiviert. Über das Mausrad wurde die Größe eines Kreises eingestellt, um eine Region auszuwählen, dessen Intensitätswerte zur Anpassung des Schwellwertbereichs verwendet werden sollen.

Wie Abbildung 6.4 zeigt, wurde der Schwellwertbereich so angepasst, dass die ausgewählten Voxel nun auch ein Label erhalten haben. Für den Fall, dass die Anpassung zu unerwünschten Ergebnissen führt, beispielsweise nicht zum Objekt gehörende Voxel ein Label erhalten, ist es möglich, den Vorgang wieder rückgängig zu machen.

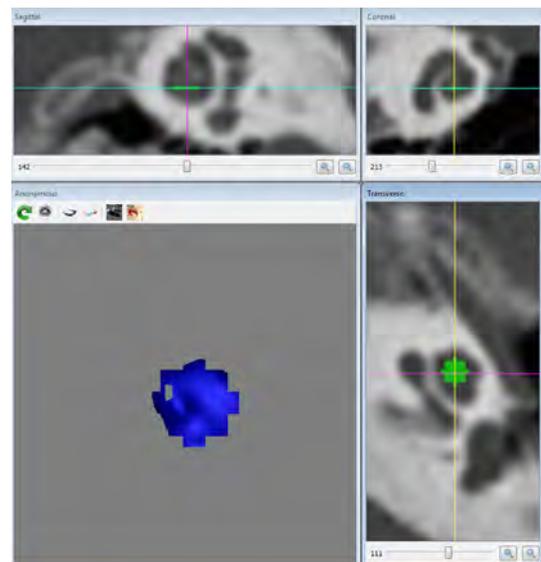
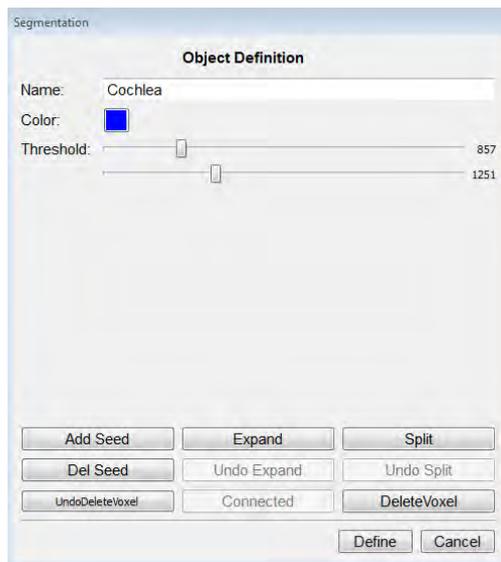


Abbildung 6.2: Nach Setzen des Startpunkts haben alle im Kreis befindlichen Voxel ein Label erhalten.

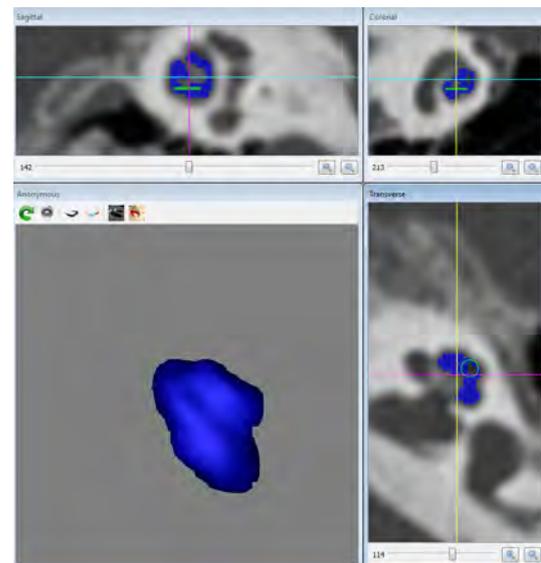
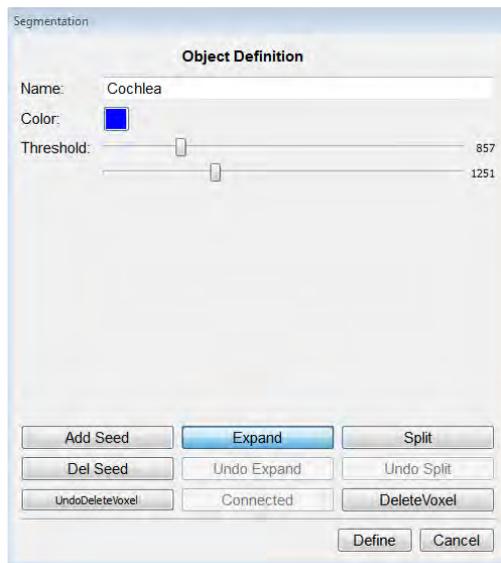


Abbildung 6.3: Nach einigen Wachstumsiterationen des Region Growing ist zu erkennen, dass einige Voxel der Cochlea kein Label erhalten haben, da sie nicht im Schwellwertbereich liegen. Um den Schwellwertbereich um die Intensitäten dieser Voxel zu erweitern, wurde die Schaltfläche „Expand“ aktiviert und die Größe des Kreises an die Region angepasst.

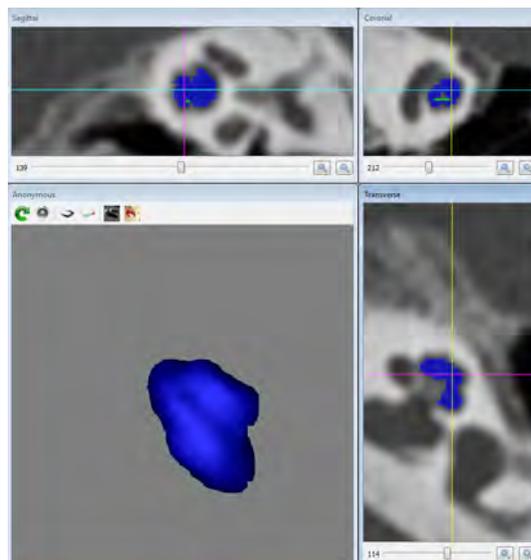
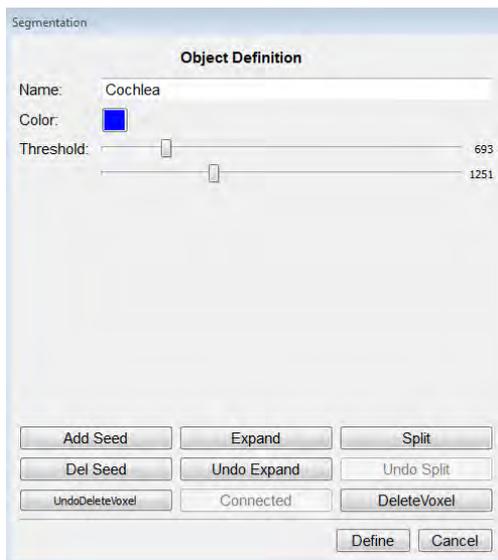


Abbildung 6.4: Ergebnis der Schwellwerterweiterung. Die Voxel der ausgewählten Region haben ein Label erhalten und der Schwellwertbereich wurde entsprechend angepasst.

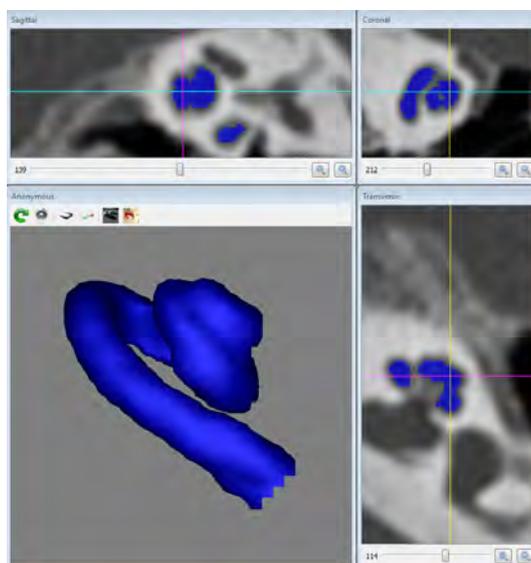
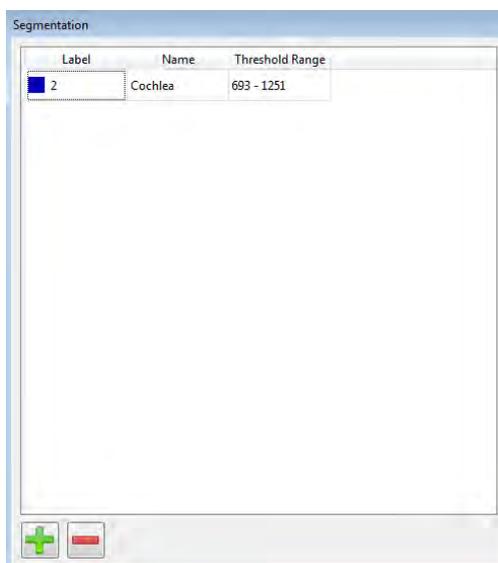
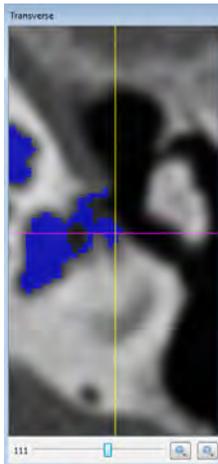


Abbildung 6.5: Nach weiteren Wachstumsiterationen konnte die Cochlea komplett segmentiert werden. Über die Schaltfläche „Define“ wurde sie mit zugehöriger Farbe und Schwellwertbereich in einer Objektliste gespeichert.

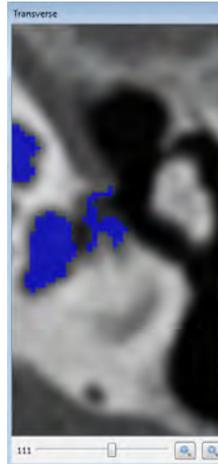
Nach der vollständigen Segmentierung der Cochlea wurde sie durch Betätigung der Schaltfläche „Define“ gespeichert. Abbildung 6.5 zeigt das Fenster, welches sich nach Definierung eines Objektes öffnet. Es beinhaltet eine Liste der aktuell segmentierten Objekte. Für die Cochlea wurde als Label der Wert 2 gesetzt und ein Schwellwertbereich von 693 - 1251 ermittelt. Über die Schaltfläche (-) hat der Benutzer nun die Möglichkeit, das Objekt zu löschen oder ein weiteres hinzuzufügen (+).

Anschließend wurden die Bogengänge segmentiert. Abbildung 6.6a zeigt die transversale Schnittebene nach einigen Wachstumsiterationen. Man sieht deutlich, dass schon Teile der angrenzenden Regionen ein Label erhalten haben, die Bogengänge jedoch noch nicht vollständig segmentiert wurden. An dieser Stelle bot sich die in Kapitel 5.3.2 erläuterte Segmentierung über Teilobjekte an. Bevor mit der Segmentierung eines weiteren Teilobjektes begonnen werden konnte, mussten jedoch zunächst die gesetzten Voxel der angrenzenden Region zurückgesetzt werden. Sowohl die Schnittebene als auch die 3D-Ansicht zeigen, dass die fälschlicherweise segmentierte Struktur nur wenige Voxel breit ist. An dieser Stelle war eine semi-automatische Trennung über morphologische Operatoren nicht möglich, da das ausgewählte Voxel nach wenigen Iterationen erodiert worden wäre. Daher wurde die Trennung manuell durchgeführt. Die Schaltfläche „Delete Voxel“ wurde aktiviert und durch aktivierte Maustaste die verbindenden Voxel gelöscht. Abbildung 6.6b und Abbildung 6.6c zeigen das Volumen während und nach der Trennung. Es ist deutlich zu erkennen, dass die Region, die durch die manuelle Trennung nicht mehr mit der Region des Startpunktes verbunden ist, von der Segmentierung ausgeschlossen wurde.

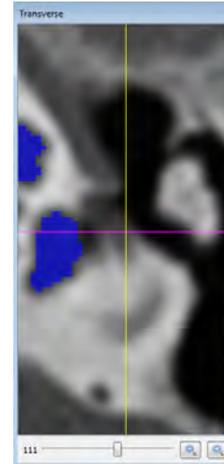
Nach Fertigstellung des ersten Teilobjektes wurde ein zweiter Startpunkt in eine noch nicht gelabelte Region der Bogengänge gesetzt (Abbildung 6.7). Schon zu diesem Zeitpunkt ist zu sehen, dass der Schwellwertbereich von 1346 - 1520 nicht mit dem des ersten Teilobjektes übereinstimmt. Dieses Verfahren wurde mit einem weiteren Teilobjekt fortgesetzt, sodass die gesamte Struktur der Bogengänge segmentiert werden konnte. Der Name wurde eingetragen und eine Farbe festgelegt. Abbildung 6.8 zeigt, dass in der Objektliste für jedes Teilobjekt ein neues Label verwendet und der entsprechende Schwellwertbereich zugeordnet wurde. Die Unterschiede der Schwellwertbereiche sind deutlich zu erkennen, weshalb an dieser Stelle eine Anpassung des Schwellwertbereichs für die gesamte Region, wie schon in Kapitel 5.3.2 erklärt, zu einer fehlerhaften Visualisierung geführt hätte. Durch denselben Namen sind die Teilobjekte eindeutig einem Gesamtobjekt zu-



(a) *Transversale Schnittebene vor der manuellen Trennung*



(b) *Die verbindenden Voxel wurden bei aktivierter Maustaste ausgewählt.*



(c) *Nach dem Loslassen der Maustaste wurden alle Voxel, die nicht mehr mit der Cochlea zusammenhängen, zurückgesetzt.*

Abbildung 6.6: *manuelle Trennung*

geordnet. Abbildung 6.9 zeigt das Ergebnis des Region Growings bei den Gehörknöchelchen. Die Region ist großflächig in einen Teil einer angrenzenden Region gewachsen, weshalb eine Erosion des ausgewählten Pixels unwahrscheinlich ist. An dieser Stelle kam es deshalb zum Einsatz der Split-Methode. Abbildung 6.10 zeigt das Ergebnis der Trennung. Der angrenzende Bereich wurde vollständig mit dem Maskenlabel versehen, sodass ein Wachsen in diese Region nicht mehr möglich ist. In Abbildung 6.11 ist das Endergebnis der Segmentierung zu sehen, nachdem auch der Gesichtsnerv segmentiert wurde.

Wie die Objektliste zeigt, konnten alle Objekte bis auf die Bogengänge über einen Schwellwertbereich segmentiert werden.

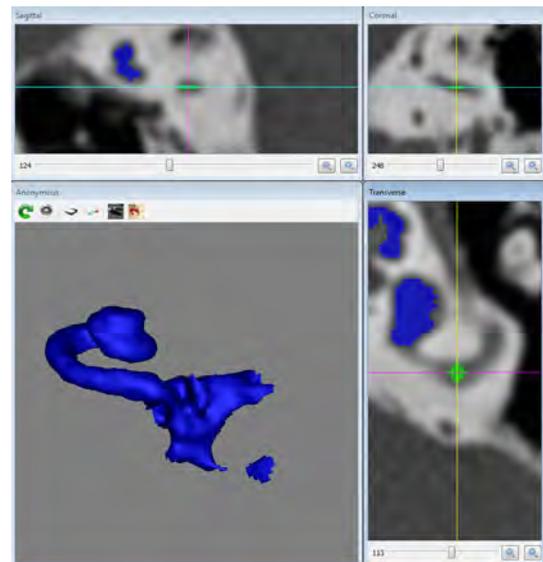
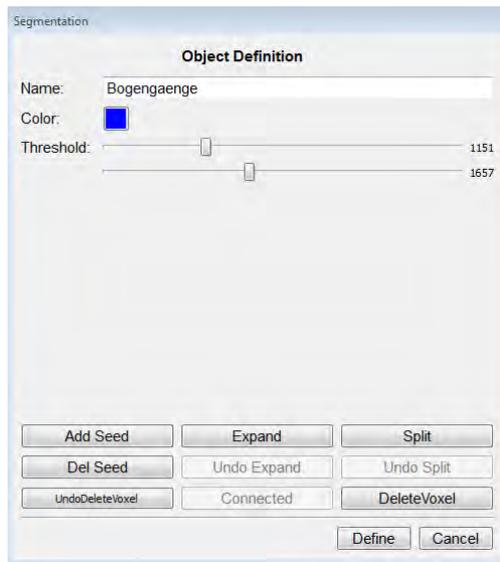


Abbildung 6.7: Für die Bogengänge wurde ein zweiter Startpunkt für ein weiteres Teilobjekt gesetzt.

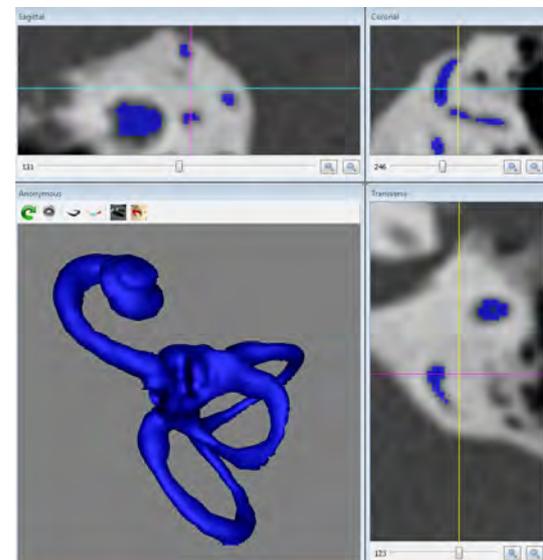
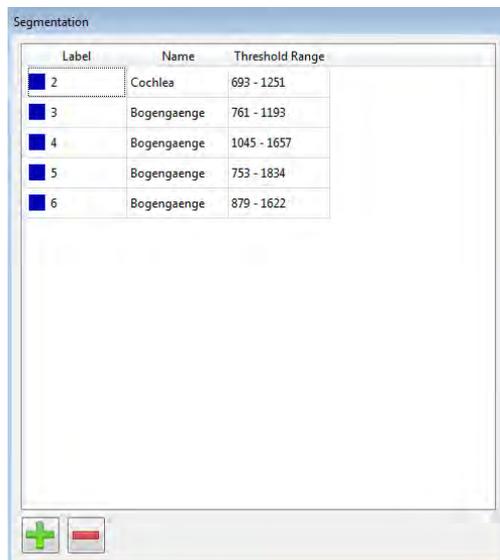


Abbildung 6.8: Die Bogengänge wurden durch vier Teilobjekte segmentiert. Diese Teilobjekte wurden in der Objektliste gespeichert und sind durch den Namen eindeutig den Bogengängen zugeordnet.

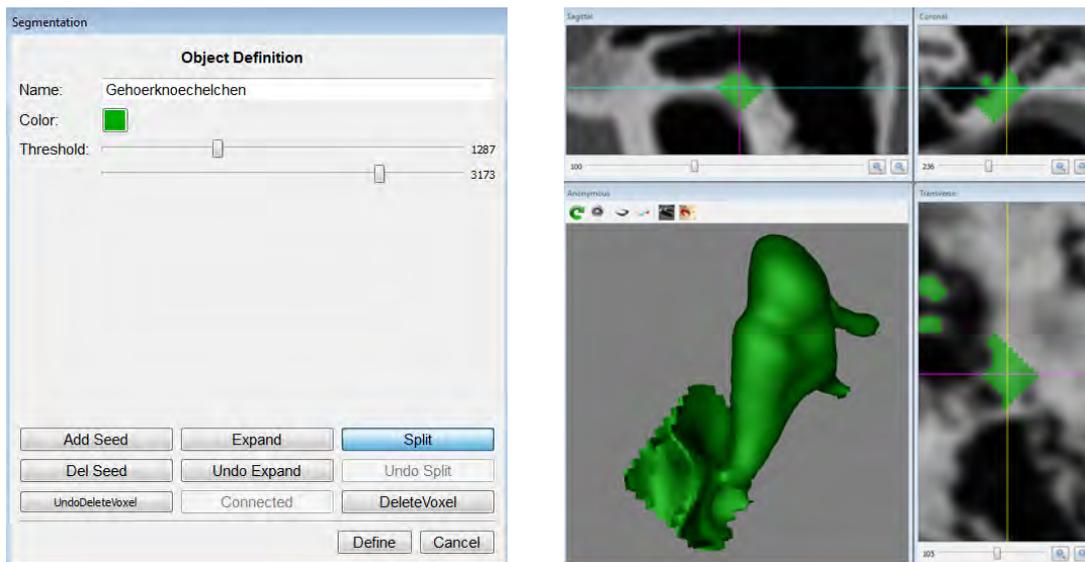


Abbildung 6.9: Bei der Segmentierung der Gehörknöchelchen kam es zu einem großflächigen Wachsen in eine angrenzende Struktur. An dieser Stelle wurde die „Split“-Methode angewandt. Die zu entfernende Struktur wurde bereits ausgewählt (gelb-pinkes Kreuz).

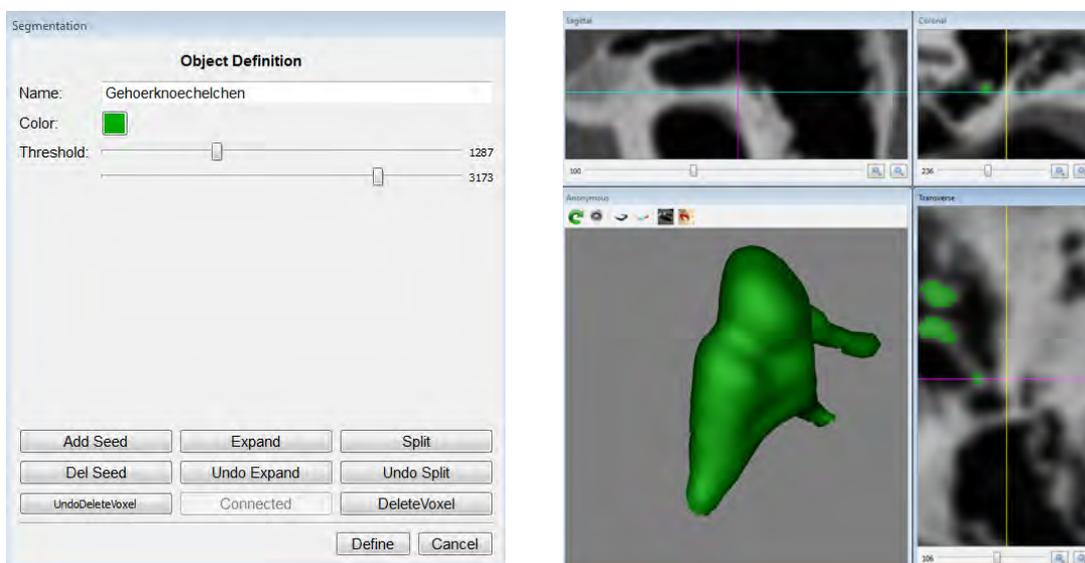


Abbildung 6.10: Durch die „Split“-Methode konnte das an die Cochlea angrenzende Objekt komplett entfernt werden.

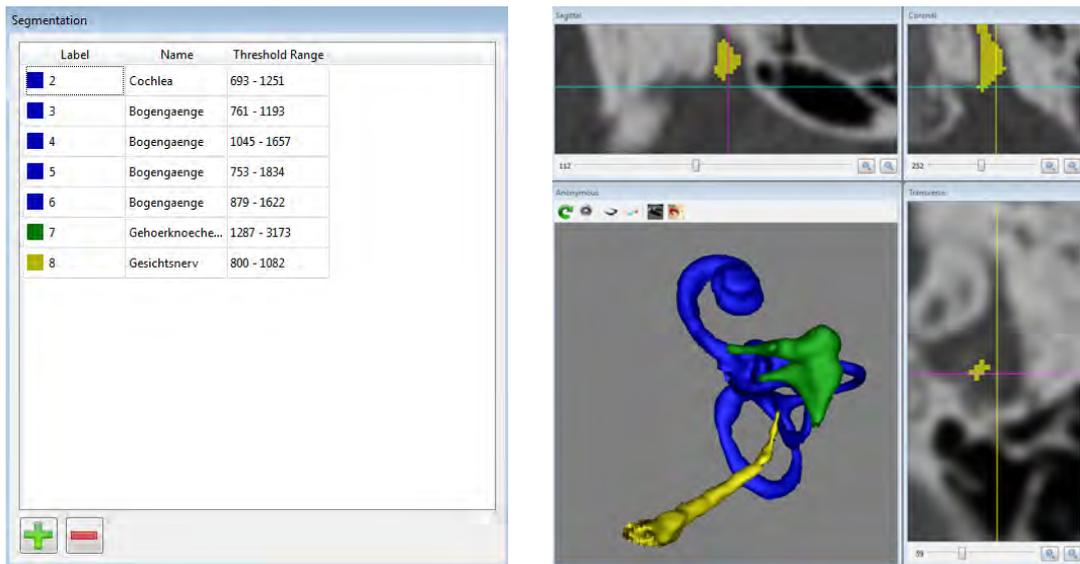


Abbildung 6.11: *Ergebnis der Segmentierung aller vier Risikostrukturen*

7 Schlussfolgerungen

7.1 Fazit

Im Rahmen dieser Arbeit wurden zunächst bestehende Segmentierungsapplikationen hinsichtlich der Eignung für den Voxel-Man Chirurgie-Simulator untersucht. Für MeVisLab DICOM-Daten wurde ein Importer zur Verfügung gestellt, sowie eine Methode integriert, um den Schwellwertbereich einer segmentierten Region nachträglich ermitteln zu können. Dadurch wurde die Möglichkeit geschaffen, mit MeVisLab segmentierte Daten in den Simulator zu importieren und zu visualisieren. Zudem konnte gezeigt werden, dass eine anschließende Ermittlung des Schwellwertbereichs zu glatten Oberflächen im Voxel-Man System führt, solange die Segmentierung über Schwellwerte erfolgt. Eine gründliche Untersuchung verschiedener Applikationen führte zu dem Schluss, dass keines dieser Segmentierungsapplikationen die Anforderungen erfüllt, um standardmäßig als Segmentierer für den Simulator verwendet zu werden.

Daher wurde im Anschluss daran ein Segmentierungsverfahren entwickelt und in das bestehende Voxel-Man System integriert. Das Verfahren arbeitet über ein Region Growing unter Berücksichtigung eines Schwellwertbereichs, der durch Setzen eines Startpunktes für das Region Growing ermittelt wird. Auch eine Shrink-Methode wurde implementiert, um einzelne Iterationen des Region Growings rückgängig zu machen. Eine weitere Methode ermöglicht die Erweiterung dieses Schwellwertbereichs durch Auswahl weiterer Voxel. Für die Trennung von Objekten wurde zunächst eine Methode zur manuellen Trennung implementiert. Durch Verwendung einer Maske für die gelöschten Voxel werden die abgetrennten Bereiche nach erneutem Ausführen des Region Growings nicht mehr erreicht.

Zur Trennung über morphologische Operatoren wurde eine Methode entwickelt (Split-Methode), die sich an das Verfahren des Linux-basiereten Voxel-Man Segmentierer anlehnt. Nach Spezifizierung des abzutrennenden Bereichs wird automatisch eine Trennung über Erosion, Connected-Components und Dilatation

durchgeführt. Der abgetrennte Bereich wird mit einem Maskenlabel versehen, wodurch ein Wachsen in diese Region unterbunden wird.

Die 3D-Visualisierung in Echtzeit konnte bei Änderung des Schwellwertbereichs nicht umgesetzt werden, was eine Feinjustierung des Schwellwertbereichs erschwert. Da es somit schwierig ist, einen optimalen Schwellwertbereich zu ermitteln, wurde über ein spezielles Objektmanagement die Möglichkeit gegeben, ein Objekt aus mehreren segmentierten Teilobjekten zusammenzufügen.

Wie Kapitel 6 zeigte, konnten alle vier Risikostrukturen des Innenohrs mithilfe der implementierten Methoden segmentiert werden. Der Benutzer muss keine bildverarbeitenden Algorithmen manuell bedienen, sondern lediglich per Mausklick Regionen auswählen, die zur gewünschten Region oder einer anderen Region gehören. Dadurch wird eine einfache und intuitive Bedienung gewährleistet. Die Segmentierung führte bis auf die Trennstellen zu glatten Oberflächen. Weitere Unebenheiten ergaben sich an den Schnittstellen zweier Teilobjekte. Diese sind auf die Größenunterschiede der Teilobjekte zurückzuführen, die durch die momentan fehlende Feinjustierung des Schwellwertbereichs nicht behoben werden können.

7.2 Ausblick

In Abschnitt 5.3.3 wurde bereits auf die Schwierigkeit der Echtzeitvisualisierung bei der Schwellwertanpassung eingegangen. Eine Möglichkeit eine Schwellwertanpassung mit 3D-Visualisierung in Echtzeit anzunähern bzw. eine Vorschau zu liefern, soll im Folgenden anhand eines Beispiel erläutert werden.

Abbildung 7.1 zeigt das Ergebnis eines Region Growings nach der 16. Iteration bei einem Schwellwert von $S = [25, 60]$. Die linke Abbildung zeigt die Intensitätswerte der einzelnen Elemente, während in den Elementen der rechten Abbildung die Erzeugungssiterationen eingetragen sind. Eine Möglichkeit bestünde nun darin, bei Änderung des Schwellwerts die bisherigen Pfade des Region Growings beizubehalten und ausgehend von den aktuellen Randvoxeln die Umgebung anzupassen. Damit gewährleistet ist, dass keines der bereits gesetzten Voxel außerhalb des Schwellwertbereichs liegt, darf nur eine Schwellwerverweiterung stattfinden.

Man nehme an, das tatsächliche Objekt sei an den Kanten noch nicht vollständig ausgefüllt, sodass der Schwellwert manuell angepasst werden soll. Kapitel 5.3.3

zu suchen, die für die Split-Methode geeignet wären und diese dem Benutzer anzubieten.

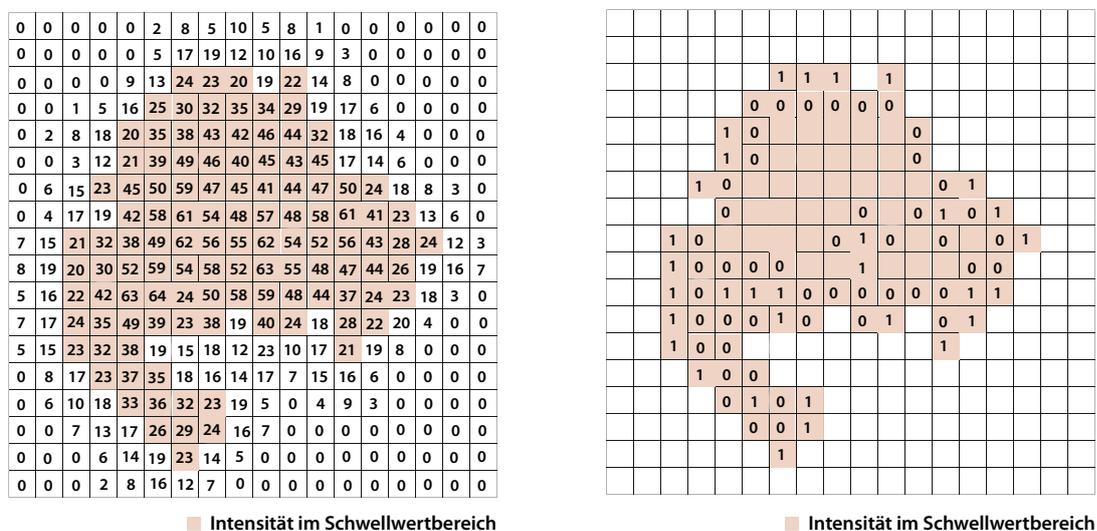


Abbildung 7.2: Die Abbildung zeigt das mögliche Verfahren, um eine Art „Vorschau“ einer Schwellwertanpassung in Echtzeit zu realisieren. Die Randvoxel haben die Erzeugungiteration 0 erhalten und es wurde eine Iteration des Region Growings auf dem neuen Schwellwertbereich von $S = [20, 65]$ durchgeführt.

Die Segmentierungsmethode wurde für kleine Strukturen entwickelt, da es bei größeren Objekten sinnvoller wäre, über einen globalen Schwellwert zu segmentieren. Aktuell findet der Verbindungstest zweier Objekte in der Split-Methode über ein Region Growing statt. Dies ist ausreichend, wenn es um die Segmentierung kleiner Strukturen geht. Möchte man die Methode jedoch auch für größere Strukturen verwenden, sollten auf rechenzeitsparendere Algorithmen umgestiegen werden. Ein Idee wäre es einen A*-Algorithmus zu verwenden, der speziell für 3-dimensionale Daten optimiert wurde. Ein Beispiel eines solchen Algorithmus ist in Niu und Zhuo [2008] beschrieben. Dabei handelt es sich um einen heuristischen Algorithmus, welcher den kürzesten Pfad zweier Punkte ermittelt.

Literatur

- 3D Slicer* (2014). Surgical Planning Laboratory. URL: <http://www.slicer.org/pages/Introduction>.
- Foley, James D. u. a. (1990). *Computer Graphics: Principles and Practice (2Nd Ed.)* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. ISBN: 0-201-12110-7.
- Handels, H. (2009). *Medizinische Bildverarbeitung: Bildanalyse, Mustererkennung und Visualisierung für die computergestützte ärztliche Diagnostik und Therapie*. Leitfäden der Informatik. Vieweg Verlag, Friedr. & Sohn Verlagsgesellschaft mbH. ISBN: 9783834895714.
- Höhne, K. H. und R. Bernstein (1986). „Shading 3D-Images from CT Using Gray-Level Gradients“. In: *IEEE Trans Med Imaging* 5.1, S. 45–47.
- Höhne, K. H. und U. Tiede (1987). „Viewing operations for 3D-Visualisierungstomographic gray level data“. In: *Computer Assisted Radiology*, S. 599–609.
- Höhne, K. und W. Hanson (1992). „Interactive 3D segmentation of MRI and CT volumes using morphological operations“. In: *Journal of Computer Assisted Tomography* 16.2, S. 285–294.
- Interactive 3D Image Segmentation Software* (2014). TurtleSeg. URL: <http://www.turtleseg.org/index.php?page=software>.
- ITK Snap* (2014). ITK Snap.
- Jähne, Bernd (2002). *Digitale Bildverarbeitung*. 5. Aufl. 2002. Heidelberg: Springer-Verlag GmbH. ISBN: 978-3-540-41260-1.
- Maintz, J. B. Antoine und Max A. Viergever (1998). „A Survey of Medical Image Registration“. In: *Medical Image Analysis* 2, S. 1–36.

- Marr, D. und E. Hildreth (1980). „Theory of Edge Detection“. In: *Proc. Royal Society of London* 207, S. 187–217.
- Meier (2014). *Optische Täuschungen*. URL: <http://www.3d-meier.de/Galerie/Tausch/Seite2.html>.
- MeVisLab (2014). MeVis Medical Solutions. URL: <http://www.mevislab.de>.
- MIPAV (2014). Center for Information Technology. URL: <http://mipav.cit.nih.gov/>.
- Niu, Lei und Guobin Zhuo (2008). „An improved real 3D A * Algorithm for difficult path finding situation“. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, S. 927–930.
- Regensburg, Universitätsklinikum (2014). *Grundlagen Lärm und Lärmschwerhörigkeit*. URL: <http://www.uniklinikum-regensburg.de/imperia/md/content/kliniken-institute/hals-nasen-ohren-heilkunde/audiologie/laerm.pdf>.
- Stiehl H. Siegfried und Jackèl, U. (1987). „On a framework for processing and visualizing spatial images“. In: *Computer Assisted Radiology*, S. 665–670.
- Tiede (1998). „High quality rendering of attributed volume data“. In: *Proc. IEEE Visualisation*, S. 255–262.
- Tiede, U. (1999). „Realistische 3D-Visualisierung multiattributierter und multiparametrischer Volumendaten“. Diss. Universität Hamburg, Fachbereich Informatik.
- Withey, D. J. und Z. J. Koles (2004). „Medical Image Segmentation: Methods and Software“. In: *Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging*, S. 140–143.

Eidesstattliche Erklärung

Hiermit versichere ich, Teresa Schiffner, Matrikel-Nr. 1894208, dass ich diese Masterarbeit mit dem Thema

Evaluierung von 3D Segmentierungsapplikationen sowie Entwicklung und Implementierung einer Segmentierungsmethode für den Voxel-Man Chirurgie-Simulator

selbständig und ohne fremde Hilfe angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht.

Hamburg, den 18.03.2014

TERESA SCHIFFNER