KOMPLEXE REGIONENBASIERTE SEGMENTIERUNGSVERFAHREN MIT DER LEVEL SET METHODE IN DER MEDIZINISCHEN BILDVERARBEITUNG

Diplomarbeit im Studiengang Mathematik mit Studienrichtung Informatik

vorgelegt von Björn Scheuermann

am

Institut für Mensch-Maschine-Kommunikation Leibniz Universität Hannover Lehrstuhl für Graphische Datenverarbeitung

Prüfer: Prof. Dr. Franz-Erich Wolter Zweitprüfer: PD Dr. Peter Milbradt

Abgabedatum: 29. September 2008

KOMPLEXE REGIONENBASIERTE SEGMENTIERUNGSVERFAHREN MIT DER LEVEL SET METHODE IN DER MEDIZINISCHEN BILDVERARBEITUNG

Diplomarbeit
im Studiengang Mathematik mit Studienrichtung
Informatik

vorgelegt von Björn Scheuermann

am

Institut für Mensch-Maschine-Kommunikation Leibniz Universität Hannover Lehrstuhl für Graphische Datenverarbeitung

> Prüfer: Prof. Dr. Franz-Erich Wolter Zweitprüfer: PD Dr. Peter Milbradt

Abgabedatum: 29. September 2008

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und ausschließlich mit Hilfe der angegebenen Quellen und Hilfsmittel angefertigt habe.

Hannover, 29. September 2008, Björn Scheuermann

Danksagung

"Ich sage nur ein Wort: Vielen Dank" Andreas Brehme, deutscher Fußballspieler.

An dieser Stelle möchte ich mich bei allen bedanken, die mich in den letzten sechs Monaten während meiner Diplomarbeit unterstützt haben.

Herzlichst bedanken möchte ich mich bei meinem Betreuer Dipl. Math. Philipp Blanke und bei Dipl. Math. Ingo Friese für die Unterstützung und die zahlreichen Diskussionen, die mir im Laufe dieser Arbeit sehr geholfen haben.

Ganz besonderer Dank gebührt meinen Eltern, meiner Familie und meiner Freundin Christine Wolf, ohne deren uneingeschränkte finanzielle und moralische Unterstützung es wahrscheinlich nicht zum Abschluss dieser Arbeit und dem damit verbundenen ?erfolgreichen? Abschluss meines Studiums gekommen wäre.

Zusammenfassung

In der medizinischen Diagnostik oder Operationsplanung werden immer häufiger Bilddaten, die von einem Röntgen-Gerät, Magnet-Resonanz-Tomographen oder Computer-Tomographen, herangezogen. Diese Daten müssen von einem Experten manuell segmentiert werden, um so die relevanten von den irrelevanten Daten zu trennen. Besonders bei dreidimensionalen Bilddaten ist der manuelle Prozess der Segmentierung jedoch sehr zeitaufwendig. Zum einen muss sich der Experte durch eine große Anzahl von Schnittbildern arbeiten und zum anderen wird es dem Experten durch die typischen Eigenschaften der Bilddaten erschwert die relevanten Daten von irrelevanten zu unterscheiden.

Daher sind effiziente, automatische oder semiautomatische Segmentierungsverfahren, die dem Benutzer helfen, mit wenig Interaktion relevante Informationen zu finden, Gegenstand aktueller Forschung. Solche Segmentierungsalgorithmen trennen relevante von irrelevanten Daten und erkennen Bildinhalte, die unter bestimmten Homogenitätskriterien inhaltlich zusammen gehören.

Ziel der vorliegenden Diplomarbeit ist die Weiterentwicklung und Optimierung zweier regionenbasierter Segmentierungsverfahren die auf der Level Set Methode beruhen. Zum einen wird das sogenannte "edge stopping" Verfahren untersucht und zum anderen der "energy minimization" Algorithmus, der auf der Minimierung eines Energiefunktionals beruht. Die Hauptziele, die in dieser Arbeit verfolgt werden sind:

- Weiterentwicklung der genannten Segmentierungsverfahren,
- Anpassung der Segmentierungsmodelle an medizinische Volumendaten und
- Verbesserung der Laufzeit der beiden Verfahren.

Hierzu werden die beiden verschiedenen Modelle der Segmentierung ausführlich hergeleitet, analysiert und die Vor- beziehungsweise Nachteile der einzelnen Verfahren aufgezeigt. Darauf aufbauend werden einige Veränderungen an den Modellen vorgeschlagen und Algorithmen vorgestellt, die die Laufzeit der Segmentierungsverfahren deutlich verbessern.

Abschließend werden die Verfahren ausgiebig an erstellten Phantomdaten und Volumendaten aus der medizinischen Praxis vorgeführt und getestet.

Inhaltsverzeichnis

1 Einführung						
		1.0.1	Gliederung	2		
	1.1	.1 Dreidimensionale bildgebende Verfahren in der Medizin				
		1.1.1	Computer-Tomographie	3		
		1.1.2	Magnet-Resonanz-Tomographie	4		
	1.2	Segme	entierung	5		
		1.2.1	Min-Max Segmentierung	7		
		1.2.2	Region-Grow Segmentierung	7		
2	Ma	themat	tische Grundlagen	9		
	2.1 Differentialgeometrie					
	2.2	2 Aktive Konturen				
	2.3	B Level Set Methode				
	2.4	Variat	ionsrechnung	20		
3	Modelle und Methoden der Segmentierung					
	3.1	1 Das "edge stopping" Verfahren				
	3.2	3.2 Der "energy minimization" Algorithmus				
		3.2.1	Herleitung des <i>Chan-Vese</i> Funktionals	35		
		3.2.2	Beziehung zum Mumford-Shah Funktional	39		
		3.2.3	Das Chan-Vese Funktional und Level Set Methoden	40		
		3.2.4	Approximationen der Heaviside Funktion	47		

INHALTSVERZEICHNIS

	3.3	Weiterentwicklung der Modelle			
		3.3.1	Weiterentwicklung des "edge stopping" Verfahrens	50	
		3.3.2	Weiterentwicklung des "energy minimization" Algorithmus $$.	56	
4	Imp	lement	ierung	63	
	4.1	Die Kla	asse Distancefunction	65	
		4.1.1	Die Attribute der Klasse Distancefunction	65	
		4.1.2	Die Methoden der Klasse Distancefunction	65	
	4.2	Die Kla	assen SegGenSnake und SegGenSnakeEnergy	66	
5	Erg	ebnisse	•	69	
	5.1	Laufzei	itanalyse der Verfahren	69	
		5.1.1	Das "edge stopping" Verfahren	69	
		5.1.2	Der "energy minimization" Algorithmus	70	
	5.2	Validie	rung der Verfahren	71	
		5.2.1	Das "edge stopping" Verfahren	71	
		5.2.2	Der "energy minimization" Algorithmus	76	
6	Fazit und Ausblick 8				
	6.1	Zusam	menfassung	83	
	6.2	Ausblie	ck	84	
	6.3	Fazit .		85	
A	Bild	ler der	Beispiele	87	
В	Not	ation		101	
\mathbf{C}	C Beiligende CD			105	

Kapitel 1

Einführung

In vielen Bereichen der modernen Medizin dienen Bilddaten aus Röntgen-Geräten, Magnet-Resonanz-Tomographen, Computer-Tomographen sowie Sonographen der Diagnostik oder Operationsplanung. Typischerweise sind diese Bilddaten häufig relativ kontrastarm und durch Bildrauschen gestört. Zudem ist die optimale Wahl des Bildausschnitts und der Richtung der Aufnahme nicht eindeutig zu bestimmen. Dem Betrachter wird es dadurch erschwert, relevante Informationen von irrelevanten Informationen zu unterscheiden. Durch diese Problematik können die relevanten Informationen meist nur von einem Experten manuell segmentiert werden.

Diese manuelle Segmentierung ist vor allem bei dreidimensionalen Volumendaten sehr zeitaufwändig, da sich der Experte durch mehrere dutzend oder sogar hunderte Schichtbilder beziehungsweise Datensätze arbeiten muss. Hier wäre ein effizientes Verfahren notwendig, dass mit geringem Bedarf an Benutzerinteraktion die relevanten Informationen findet und von den irrelevanten Daten trennt beziehungsweise alle Daten segmentiert, die unter einem bestimmten Homogenitätskriterium inhaltlich zusammen gehören. Diese semiautomatischen Verfahren sind daher Gegenstand aktueller Forschung.

Den oben genannten Kriterien übergeordnet ist das Kriterium der Robustheit gegenüber Bildstörungen und das Kriterium der Genauigkeit, weil ein (semi-) automatisches Verfahren das manuelle Verfahren nur dann vollständig ersetzen kann, wenn es zumindest eine ebenbürtige Genauigkeit bietet.

Zurzeit existieren bereits eine Vielzahl von Verfahren, die unterschiedliche Ansätze verfolgen und es dem Benutzer erleichtern, die relevanten Informationen von irrelevanten zu trennen. Zu den Einfachsten solcher Verfahren gehören unter anderem die Min-Max Segmentierung und die Region-Grow Segmentierung, die kurz vorgestellt werden, da sie im Laufe der Arbeit zum Vergleich und zur sogenannten

1.1. DREIDIMENSIONALE BILDGEBENDE VERFAHREN IN DER MEDIZIN

Vorsegmentierung verwendet werden. In meiner Studienarbeit [Sch07] wurden zwei regionenbasierte Verfahren implementiert und untersucht, die zur Segmentierung aktive Konturen (oder "Snakes") verwenden und auf den Level Set Methoden aufbauen. Das erste Verfahren nutzt das sogenannte "edge-stopping", wohingegen das zweite Verfahren auf einem "energy minimization" Algorithmus basiert.

Ziel dieser Diplomarbeit ist die Optimierung und Weiterentwicklung der oben genannten Segmentierungsverfahren. Mit den entwickelten Algorithmen soll es dem Benutzer erleichtert werden, Volumendaten aus der medizinischen Praxis zu segmentieren, um so die relevanten Informationen schneller als bisher von den irrelevanten zu unterscheiden. Diesbezüglich ist eine Verbesserung der Laufzeiten der beiden Verfahren und eine Anpassung der Modelle an medizinische Volumendaten erforderlich.

1.0.1 Gliederung

Nach einer kleinen Einführung in die medizinische Bildgebung werden in Kapitel 2 die mathematischen Grundlagen zum Verständnis dieser Arbeit geschaffen. Diese umfassen eine Einführung in die Differentialgeometrie, die aktiven Konturen und die Level Set Methode sowie einen kurzen Einblick in die Variationsrechnung, die zum numerischen Lösen beim "energy minimization" Algorithmus benötigt wird. Aufbauend auf den mathematischen Grundlagen werden in Kapitel 3 die beiden Modelle der Segmentierung, das sogenannte "edge stopping" Verfahren und der "energy minimization" Algorithmus, hergeleitet. In Abschnitt 3.3 werden die Weiterentwicklungen und Optimierungen der Modelle, die im Laufe dieser Arbeit entstanden, ausführlich erläutert. In Kapitel 4 wird näher auf die Implementierung der Segmentierungsverfahren und deren Optimierungen eingegangen. Abschließend werden die implementierten Algorithmen in Kapitel 5 an Phantomdaten und medizinischen MRT- und CT-Datensätzen sowohl im Zweidimensionalen als auch im Dreidimensionalen Anwendungsfall ausführlich getestet und in Kapitel 6 folgt schließlich eine Zusammenfassung der Ergebnisse und ein Ausblick auf zukünftige Arbeiten.

1.1 Dreidimensionale bildgebende Verfahren in der Medizin

Im Folgenden werden die beiden zurzeit am häufigsten verwendeten dreidimensionalen bildgebenden Verfahren in der Medizin kurz vorgestellt. Dazu zählen

zum einen die Computer-Tomographie und zum anderen die Magnet-Resonanz-Tomographie. Eine ausführliche Beschreibung dieser Verfahren findet sich unter anderem in dem Buch [Dös06] von O. Dössel oder in [Mor95] von H. Morneburg.

1.1.1 Computer-Tomographie

Die Computer-Tomographie (kurz "CT") ist ein auf Röntgenstrahlen basierendes, schnittbildgebendes Verfahren.

Es wurde 1972 durch den amerikanischen Physiker A.M. Cormack und den britischen Ingenieur G.N. Hounsfield entwickelt und gehört seitdem zu einem der wichtigsten diagnostischen Verfahren der Radiologie.

Seit Einführung der ersten Computer-Tomographie-Geräte hat sich am Grundprinzip des Verfahrens nichts verändert. Die Geräte wurden jedoch schnell technisch weiterentwickelt, wodurch sich die Bildqualität verbessert und auch die Aufnahmezeit verkürzt hat. Bei der Bildqualität wurde das Rauschen vermindert und die Anzahl der auftretenden Artefakte verringert. Die Aufnahmezeit wurde von fünf Minuten für eine Schichtaufnahme bei den ersten Geräten auf einen Bruchteil von Sekunden bei moderneren Geräten reduziert.

Bei der Computer-Tomographie wird durch das Gerät ein schmaler Röntgen-Fächerstrahl erzeugt. Dieser durchdringt das gewünschte Objekt beziehungsweise die gewünschte Körperstelle. Die verschiedenen Strukturen innerhalb des Objekts beziehungsweise des Körpers schwächen diesen Strahl unterschiedlich stark ab. Die abgeschwächten Strahlen werden von Detektoren gegenüber der Strahlenquelle empfangen, elektronisch aufbereitet und an einen Computer weitergeleitet. Dieser Vorgang wird dann immer wieder aus verschiedenen Richtungen durchgeführt, so dass verschiedene Projektionen beziehungsweise Ansichten von der gleichen Schicht erzeugt werden.

Durch ein iteratives Verfahren berechnet der Computer aus diesen Projektionen ein Graustufenbild. Dieses kann am Bildschirm oder auf einem Ausdruck betrachtet und ausgewertet werden. Die einzelnen Grauwerte stellen den Schwächungskoeffizienten auf der Hounsfield-Skala dar, beschreiben dementsprechend Dichtewerte. In der praktischen Anwendung ist die Hounsfield-Skala auf 12bit (-1024 bis +3071) beschränkt. Für die Auswertung wird der Bereich der Grauwertdarstellung zusätzlich, je nach untersuchtem Objekt, durch eine sogenannte Fenster-Weite und das Fenster-Zentrum begrenzt, weil das menschliche Auge nicht in der Lage ist, 12bit verschiedene Grauwerte zu unterscheiden.

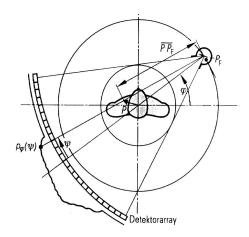


Abbildung 1.1: Prinzip der Computer-Tomographie. Quelle: [Lie06].

Der wesentliche Nachteil der Computer-Tomographie ist die hohe Strahlenbelastung für den Patienten, die bis zu 1000-mal höher als bei der normalen Röntgenaufnahme ist. Ein weiterer Nachteil besteht in der Darstellung der Weichteile wie Organe, Muskel oder Knorpel, da diese einen ähnlichen Schwächungskoeffizienten haben. Hier lässt sich die Aussagekraft der Computer-Tomographie-Aufnahme durch das Verabreichen eines Kontrastmittels erheblich steigern.

1.1.2 Magnet-Resonanz-Tomographie

Im Gegensatz zur Computer-Tomographie basiert die Magnet-Resonanz-Tomographie (kurz "MRT") physikalisch nicht auf Röntgenstrahlen, sondern auf der sogenannten Kernspinresonanz. Diese beschreibt den Eigendrehimpuls, auch Spin genannt, von Protonen und Neutronen und den dadurch entstehenden magnetischen Moment des Atomkerns.

Bei der Magnet-Resonanz-Tomographie wird ein magnetisches Feld erzeugt, woraufhin sich der rotierende Atomkern ausrichtet und mit einer Präzessionsbewegung beginnt, dass heißt die Rotationsachse des Kerns dreht sich um die Richtung des angelegten Magnetfeldes. Um die Kerne zur dauerhaften Präzession anzuregen, wird nun ein zweites hochfrequentes Wechselmagnetfeld senkrecht zum Ersten angelegt. Zusätzlich wird ein Radiowellenimpuls ausgestrahlt, wodurch die präzedierenden Kerne beginnen Radioenergie auszustrahlen.

Informationen über das Gewebe im Inneren erhält man, indem zum einen die Zeit gemessen wird, die der Kern zur Ausrichtung benötigt (die sogenannte T1-

Relaxationszeit) und zum anderen durch Messen der die Zeit, nach der 63% der Radioenergie abgestrahlt wurde (die sogenannte T2-Relaxationszeit).

T1-gewichtete Bildserien erlauben eine hohe Ortsauflösung, wohingegen T2-gewichtete Bilder einen höheren Gewebekontrast haben. Durch abgestufte Magnetfelder wird eine Ortskodierung erzeugt, damit die gemessenen Signale den einzelnen Voxeln zugeordnet werden können.

Die Signale der einzelnen Voxel werden nun auf Grauwerte abgebildet, um sie anschaulich darzustellen. Da die Signale von vielen Parametern abhängen, wie der Stärke des Magnetfeldes, gibt es im Gegensatz zur Computer-Tomographie keine Normwerte für einzelne Gewebearten. Dadurch stützt sich die Bildinterpretation auf den Gesamtkontrast und die Signalunterschiede verschiedener Gewebearten.

Der bedeutendste Vorteil der Magnet-Resonanz-Tomographie ist die gute Darstellbarkeit von Weichteilgewebe. Im Gegensatz dazu lässt sich etwa Nervengewebe schlecht durch die Computer-Tomographie darstellen. Ein weiterer Vorteil ist die hohe Detailerkennbarkeit im Gegensatz zur Computer-Tomographie. Zudem gibt es für den Patienten keine gefährliche Strahlenbelastung. Jedoch liegen der Anschaffungspreis und der Preis für den Betrieb eines Magnet-Resonanz-Tomographen deutlich höher als der eines Computer-Tomographen.

1.2 Segmentierung

In der medizinischen Bildverarbeitung wird häufig der Begriff "Segmentierung" verwendet. Nach der Bildvorverarbeitung ist die Segmentierung meist der erste Schritt der Bildanalyse im Prozess des maschinellen Sehens.

Zu den einzelnen Schritten des maschinellen Sehens gehören (vgl. [Fri06]):

- ➤ Bildaufnahme
- ➤ Bildvorverarbeitung
- > Segmentierung
- ➤ Merkmalsextraktion
- > Klassifizierung
- ➤ Aussage

Definition 1.2.0.1

Die Zusammenfassung von Pixeln beziehungsweise Voxeln, als inhaltlich zusammengehörende Region unter einem Homogenitätskriterium, wird als Segment be-

1.2. SEGMENTIERUNG

zeichnet. Unter einer Segmentierung wird sowohl die Gesamtheit der Segmente als auch der Prozess der Erzeugung von einem oder mehreren Segmenten verstanden.

Man nennt eine Segmentierung (als Menge von Segmenten):

- vollständig, wenn jedem Pixel beziehungsweise Voxel mindestens ein Segment zugeordnet ist.
- *überdeckungsfrei*, wenn jedem Pixel beziehungsweise Voxel höchstens ein Segment zugeordnet ist.
- zusammenhängend, wenn jedes Segment ein zusammenhängendes Gebiet repräsentiert. Das heißt, dass für je zwei Pixel beziehungsweise Voxel a, b eines Segments ein Pfad von benachbarten Pixeln beziehungsweise Voxeln von a nach b existiert, der vollständig im Segment liegt.

Das Verfahren der Segmentierung (als Erzeugung von Segmenten) nennt man:

- pixel-/voxelorientiert, wenn die Entscheidung, ob ein Pixel beziehungsweise Voxel zu einem Segment gehört, für jeden Pixel beziehungsweise Voxel separat entschieden wird.
- kantenorientiert, wenn die Pixel- beziehungsweise Voxelmenge vor der Segmentierung nach Kanten oder Objektübergängen abgesucht wird.
- regionenbasiert, wenn zur Entscheidung, ob ein Pixel beziehungsweise Voxel zu einem Segment gehört, auch seine Nachbarschaft berücksichtigt wird.
- modellbasiert, wenn zur Entscheidung, ob ein Pixel beziehungsweise Voxel zu einem Segment gehört, zusätzlich a priori Information über den Bildinhalt genutzt wird.

Im Folgenden wird der Begriff Segmentierung, wenn nicht anders erwähnt, als Prozess der Erzeugung von Segmenten verstanden. Für den Menschen ist die Segmentierung ein alltäglicher und relativ einfacher Prozess, weil er dabei seine Abstraktionsfähigkeit sowie das Verständnis des gesamten Kontexts ausnutzen kann. Für den Computer hingegen ist der Prozess des Erkennens eine sehr aufwändige und rechenintensive Aufgabe.

Zunächst werden zwei elementare Verfahren der Segmentierung vorgestellt, die im Verlauf dieser Arbeit zur sogenannten Vorsegmentierung genutzt werden.

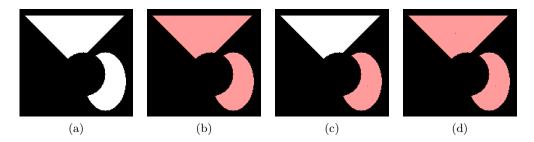


Abbildung 1.2: (a): Darstellung der Ausgangsdaten. (b): Darstellung des erzeugten Segments mit der Min-Max Segmentierung. (c): Darstellung des erzeugten Segments mit dem Region-Grow Algorithmus und einem Startvoxel. (d): Darstellung des erzeugten Segments mit dem Region-Grow Algorithmus und je einem Startvoxel pro Objekt.

1.2.1 Min-Max Segmentierung

Bei der Min-Max Segmentierung handelt es sich um ein pixel-/ voxelorientiertes Verfahren. Hierbei wird durch den Benutzer eine obere und eine untere Grauwertschranke beziehungsweise Dichteschranke festgelegt. Danach wird für jeden Pixel beziehungsweise Voxel entschieden, ob sich der Grauwert innerhalb der festgelegten Grauwertschranken befindet und er damit zum Segment gehört.

Das Segment, welches während des Prozesses der Segmentierung entsteht, besteht also aus allen Pixeln beziehungsweise Voxeln p, für die gilt:

- (i) $u_0(p) \leq \text{ obere Schranke}$,
- (ii) $u_0(p) \ge$ unter Schranke,

wobei $u_0(p)$ die Grauwertfunktion¹ beschreibt. Offensichtlich handelt es sich bei der Min-Max Segmentierung um eine überdeckungsfreie Segmentierung. Ein Beispiel der Min-Max Segmentierung ist in Abbildung 1.2 dargestellt.

1.2.2 Region-Grow Segmentierung

Die Region-Grow Segmentierung ist, wie der Name vermuten lässt, eine regionenbasierte Segmentierung und erzeugt zunächst einmal genau ein zusammenhän-

 $^{^{1}\}mathrm{Die}$ sogenannte Grauwertfunktion gibt den Grauwert eines Pixels beziehungsweise eines Voxels an.

1.2. SEGMENTIERUNG

gendes Segment. Von der Idee ist der Algorithmus ähnlich dem der Min-Max-Segmentierung, da die Entscheidung, ob ein Pixel beziehungsweise Voxel zu einem Segment gehört, wieder über eine obere und eine untere Schranke fällt. Um näher auf den Algorithmus eingehen zu können, ist der Begriff der Nachbarschaft eines Pixels beziehungsweise eines Voxels erforderlich, der im Folgenden definiert ist.

Definition 1.2.2.1

Als 6er-Nachbarschaft wird in der dreidimensionalen Bildverarbeitung, die Erweiterung der zweidimensionalen 4er-Nachbarschaft verstanden. Bei dieser werden jeweils die horizontalen und vertikalen Nachbarn betrachtet. Dieses Konzept wird nun auf den dreidimensionalen Fall erweitert, in dem zusätzlich die beiden Nachbarn bezüglich der neuen Dimension betrachtet werden. Die 26er-Nachbarschaft ist, analog zur 6er-Nachbarschaft, die Erweiterung der zweidimensionalen 8er-Nachbarschaft auf den dreidimensionalen Fall.

Bei der Region-Grow Segmentierung wird zunächst ein Startpixel beziehungsweise Startvoxel sowie eine Varianz $v \in [0...1]$ vom Benutzer festgelegt. Nun werden ausgehend vom gewählten Startpixel beziehungsweise Startvoxel alle Nachbarn betrachtet. Liegt deren Grauwert innerhalb der Varianz vom Startpixel beziehungsweise Startvoxel gehören diese zum Segment und es werden rekursiv auch deren Nachbarn betrachtet. Für jeden Pixel beziehungsweise Voxel p wird also geprüft, ob

$$u_0(p) \in [u_0(s) - u_0(s) \cdot v, u_0(s) + u_0(s) \cdot v],$$
 (1.1)

wobei $u_0(p)$ wieder den Grauwert eines Pixel beziehungsweise Voxels angibt und s der gewählte Startpixel beziehungsweise Startvoxel ist. Bei der Region-Grow Segmentierung handelt es sich offensichtlich um eine überdeckungsfreie und zusammenhängende Segmentierung. Um mehrere Segmente gleichzeitig zu erzeugen, kann das Verfahren erweitert werden, indem mehrere Startpixel oder Startvoxel zugelassen werden. Damit ist es möglich mehrere nicht zusammenhängende Segmente zu erzeugen. Zum besseren Verständnis ist in Abbildung 1.2 ein Beispiel der Region-Grow Segmentierung gegeben.

Bei den im weiteren Verlauf der Arbeit vorgestellten Verfahren wird es sich ausschließlich um regionenbasierte Verfahren handeln.

Kapitel 2

Mathematische Grundlagen

Im folgenden Kapitel werden die mathematischen Grundlagen dargestellt, die in dieser Arbeit erforderlich sind. Die wichtigsten Aspekte werden kurz angesprochen und erläutert. Zum tiefer gehenden Verständnis der mathematischen Grundlagen sei dem Leser die angegebene Literatur empfohlen.

Den Mittelpunkt dieses Kapitels stellen die aktiven Konturen und die Level Set Methode dar. Zuvor werden wichtige Begriffe der Differentialgeometrie wie parametrisierte Kurven, parametrisierte Flächen, Krümmungen von Kurven und Flächen sowie der Begriff der Flächennormale, die in dieser Arbeit benötigt werden, angegeben.

2.1 Differentialgeometrie

Um die Konzepte der Flächendarstellung zu verstehen, wird zunächst auf die Darstellung von Kurven und deren Eigenschaften eingegangen. In Folge dessen werden Flächenkurven eingeführt, um schließlich die Eigenschaften von Flächen zu erläutern. Eine ausführliche Einführung in die Konzepte der Differentialgeometrie findet man in [Car98] von M.P. do Carmo oder [Küh05] von W. Kühnel.

Definition 2.1.0.2

Unter einer parametrisierten Kurve versteht man eine stetige Abbildung

$$\alpha: I \subset \mathbb{R} \to \mathbb{R}^n$$

$$t \mapsto \alpha(t). \tag{2.1}$$

2.1. DIFFERENTIALGEOMETRIE

In diesem Zusammenhang bezeichnet I das Parametergebiet beziehungsweise Parameterintervall und $t \in I$ den Parameter. Die Kurve α heißt differenzierbar, genau dann, wenn sie unendlich oft differenzierbar ist. Genauer:

$$\alpha$$
 differenzierbar $\Leftrightarrow \alpha \in C^{\infty}$.

Bemerkung 2.1.0.1

Sei $\alpha: I \subset \mathbb{R} \to \mathbb{R}^n$ eine parametrisierte Kurve mit I = [a, b]. Dann berechnet sich die Bogenlänge oder Länge der Kurve durch:

$$\int_{a}^{b} |\alpha'(t)| dt. \tag{2.2}$$

Da eine parametrisierte Kurve nun definiert ist, werden als nächstes die Begriffe Tangentenvektor und Krümmung definiert sowie erläutert.

Definition 2.1.0.3

Sei $\alpha:I\subset\mathbb{R}\to\mathbb{R}^n$ eine parametrisierte Kurve und $t_0\in I$. Dann heißt der Vektor

$$T(t_0) := \alpha'(t_0) \tag{2.3}$$

Tangentenvektor im Kurvenpunkt $\alpha(t_0)$. Hierbei bezeichnet

$$\alpha'(t_0) = \frac{\partial \alpha}{\partial t} = (\alpha'_1(t_0), \alpha'_2(t_0), \dots, \alpha'_n(t_0))^T$$
(2.4)

die erste Ableitung für

$$\alpha(t_0) = (\alpha_1(t_0), \alpha_2(t_0), \dots, \alpha_n(t_0))^T.$$
 (2.5)

Falls $\alpha'(t) \neq 0 \ \forall t \in I$ gilt, so wird α als reguläre parametrisierte Kurve bezeichnet.

Der Tangentenvektor in einem Kurvenpunkt ist also nach Definition, die beste lineare Annäherung für die Kurve. Im späteren Verlauf der Arbeit wird auch der Begriff der Krümmung von Kurven und Flächen von Bedeutung sein. Im Folgenden wird der Begriff der Krümmung zunächst für den ebenen Fall definiert und danach auf den dreidimensionalen Fall erweitert.

Definition 2.1.0.4

Sei $\alpha: I \subset \mathbb{R} \to \mathbb{R}^2$ eine ebene parametrisierte Kurve und $t_0 \in I$. Dann bezeichnet

$$k(t_0) = \lim_{t \to t_0} \frac{\angle(\alpha'(t_0), \alpha'(t))}{|\alpha(t_0) - \alpha(t)|}$$

$$(2.6)$$

die Krümmung der Kurve im Punkt $\alpha(t_0)$.

Die Krümmung ist also ein Maß dafür, wie stark die Kurve von der linearen Annäherung abweicht. Für dreidimensionale Kurven, auch Raumkurven genannt, ist die Krümmung wie folgt definiert:

Definition 2.1.0.5

Sei $\alpha: I \to \mathbb{R}^3$ eine reguläre parametrisierte Kurve und $t_0 \in I$. Dann bezeichnet

$$k(t_0) = \frac{|\alpha'(t_0) \times \alpha''(t_0)|}{|\alpha'(t_0)|^3}$$
 (2.7)

die Krümmung der Raumkurve α an der Stelle $t_0 \in I$. Dabei bezeichnet

$$\alpha'(t_0) = \frac{\partial \alpha}{\partial t} = (\alpha'_1(t_0), \alpha'_2(t_0), \alpha'_n(t_0))^T, \qquad (2.8)$$

wie in (2.4) die erste Ableitung für

$$\alpha(t_0) = (\alpha_1(t_0), \alpha_2(t_0), \alpha_3(t_0))^T$$
(2.9)

und dementsprechend

$$\alpha''(t_0) = \frac{\partial^2 \alpha}{\partial t^2} = (\alpha_1''(t_0), \alpha_2''(t_0), \alpha_n''(t_0))^T, \qquad (2.10)$$

die zweite Ableitung der Raumkurve $\alpha(t)$ an der Stelle t_0 .

Da sich die vorliegende Arbeit mit der Segmentierung von dreidimensionalen Objekten auseinandersetzt, werden im Folgenden die Erweiterung der oben genannten Definitionen auf den dreidimensionalen Fall dargestellt.

Definition 2.1.0.6

Unter einer parametrisierten Fläche versteht man eine stetige Abbildung

$$\beta: I_1 \times I_2 \to \mathbb{R}^n (x_1, x_2) \mapsto \beta(x_1, x_2).$$
 (2.11)

Hier bezeichnet $I_1 \times I_2 \subset \mathbb{R}^2$ das Parametergebiet und $x_1 \in I_1, x_2 \in I_2$ die Parameter.

Ist $\alpha:I_3\to I_1\times I_2$ eine parametrisierte Kurve, so bezeichnet die stetige Abbildung

$$\beta \circ \alpha : I_3 \to \mathbb{R}^n$$

$$t \mapsto \beta(\alpha(t))$$
(2.12)

eine Flächenkurve.

Bemerkung 2.1.0.2

Eine parametrische Fläche $\beta(x_1, x_2)$ heißt regulär, falls

2.1. DIFFERENTIALGEOMETRIE

(i)
$$\frac{\partial \beta}{\partial x_1}(x_1, x_2) = \beta_{x_1}(x_1, x_2) \neq 0 \ \forall (x_1, x_2) \in I_1 \times I_2 \text{ und}$$

(ii)
$$\frac{\partial \beta}{\partial x_2}(x_1, x_2) = \beta_{x_1}(x_1, x_2) \neq 0 \ \forall (x_1, x_2) \in I_1 \times I_2.$$

Hierbei bezeichnen β_{x_1} und β_{x_2} die partiellen Ableitungen der Abbildung $\beta(x_1, x_2)$. Ebenso heißt eine parametrische Flächenkurve $\beta(\alpha(t))$ regulär, falls

(i)
$$\frac{\partial \beta}{\partial x_1}(\alpha(t)) = \beta_{x_1}(\alpha(t)) \neq 0 \ \forall t \in I_3 \text{ und}$$

(ii)
$$\frac{\partial \beta}{\partial x_2}(\alpha(t)) = \beta_{x_2}(\alpha(t)) \neq 0 \ \forall t \in I_3.$$

Der Tangentenvektor einer Flächenkurve $\beta(\alpha(t))$ ist wie folgt definiert:

Definition 2.1.0.7

Sei $\beta(\alpha(t))$ eine reguläre Flächenkurve mit $\alpha(t) = (\alpha_1(t), \alpha_2(t))^T$ und $\beta(x_1, x_2) = (\beta_1(x_1, x_2), \beta_2(x_1, x_2), \beta_3(x_1, x_2))^T$. Dann bezeichnet

$$\frac{\partial \beta(\alpha(t))}{\partial t} = (\beta_{x_1}(\alpha(t)), \ \beta_{x_2}(\alpha(t))) \cdot \alpha'(t) = \alpha_1'(t)\beta_{x_1}(\alpha(t)) + \alpha_2'(t)\beta_{x_2}(\alpha(t))$$
 (2.13)

den Tangentenvektor der Flächenkurve.

Die Menge der Tangentialvektoren aller möglichen Flächenkurven durch den Punkt $\beta(x_{1_0}, x_{2_0})$ der Fläche β , wird Tangentialebene der Fläche β im Punkt $\beta(x_{1_0}, x_{2_0})$ genannt.

Definition 2.1.0.8

Sei $\beta(x_1, x_2)$ eine reguläre parametrisierte Fläche. Dann gilt:

$$T(x_{10}, x_{20}) = \{\lambda_1 \beta_{x_1}(x_{10}, x_{20}) + \lambda_2 \beta_{x_2}(x_{10}, x_{20}) \mid \lambda_1, \lambda_2 \in \mathbb{R}\}$$
 (2.14)

beschreibt die Tangentialebene in einem Punkt $\beta(x_{1_0}, x_{2_0})$.

$$N(x_{1_0}, x_{2_0}) = \frac{\beta_{x_1}(x_{1_0}, x_{2_0}) \times \beta_{x_2}(x_{1_0}, x_{2_0})}{|\beta_{x_1}(x_{1_0}, x_{2_0}) \times \beta_{x_2}(x_{1_0}, x_{2_0})|}$$
(2.15)

bezeichnet die Flächeneinheitsnormale im Punkt $\beta(x_{1_0}, x_{2_0})$.

Bemerkung 2.1.0.3

Sonderfälle in denen $\beta_{x_1}(x_{1_0}, x_{2_0})||\beta_{x_2}(x_{1_0}, x_{2_0})$ in einem Punkt $\beta(x_{1_0}, x_{2_0})$ und somit $|\beta_{x_1}(x_{1_0}, x_{2_0}) \times \beta_{x_2}(x_{1_0}, x_{2_0})| = 0$ gilt, werden in dieser Arbeit nicht betrachtet und demzufolge ausgeschlossen.

Nach diesem kurzen Einblick in die Darstellung von Flächen und Flächenkurven mit Hilfe der Differentialgeometrie, werden noch die mittlere Krümmung sowie die Gauß-Krümmung von Flächen angegeben. Hierzu wird zunächst der Begriff der Normalkrümmung benötigt, der im Folgenden definiert wird.

Definition 2.1.0.9

Sei $\beta(\alpha(t))$ eine Flächenkurve mit $\alpha(t) = (\alpha_1(t), \alpha_2(t))^T$, mit $k(t_0)$ der Krümmung der Flächenkurve und T_0 der Tangentialebene im Punkt t_0 . Dann gilt:

$$k(t_0) = k_n N(\alpha(t_0)) + \gamma_g S, \qquad (2.16)$$

wobei $N(\alpha(t_0))$ die Einheitsnormale von T_0 im Punkt $\beta(\alpha(t_0))$ bezeichnet und S einen Einheitsvektor auf T_0 . Der Anteil $k_nN(\alpha(t_0))$ der Krümmung k wird in diesem Zusammenhang als Normalkrümmungsvektor und k_n Normalkrümmung bezeichnet. Der Anteil $\gamma_g S$ wird geodätischer Krümmungsvektor und γ_g geodätische Krümmung in Richtung S genannt.

Anhand der Normalkrümmung von Flächenkurven kann als nächstes die Krümmung von Flächen definiert werden. Dazu werden in einem Punkt $\beta(x_{1_0}, x_{2_0})$ der Fläche β die Normalkrümmungen aller Tangentialvektoren betrachtet. Das heißt, zunächst wird die Ebene gebildet, die durch die Flächeneinheitsnormale $N(x_{1_0}, x_{2_0})$ im Punkt $\beta(u_0, v_0)$ und den Tagentialvektor $T(x_{1_0}, x_{2_0})$ aufgespannt wird. Diese Ebene wird mit der Fläche β geschnitten und schließlich die Normalkrümmung der so konstruierten Flächenkurve berechnet.

Definition 2.1.0.10

Sei $\beta(x_1, x_2)$ eine parametrisierte Fläche. Die maximale und minimale Normal-krümmung k_1, k_2 von Tangentialvektoren der Länge 1 im Punkt $\beta(x_{1_0}, x_{2_0})$ heißen Hauptkrümmungen. Anhand der beiden Hauptkrümmungen ist die Gausskrümmung K als

$$K := k_1 k_2 \tag{2.17}$$

und die mittlere~Kr"ummung~H als

$$H := \frac{k_1 + k_2}{2} \tag{2.18}$$

definiert. Die Krümmung einer Fläche beschreibt also, analog zur Krümmung einer Kurve, ein Maß für die Abweichung der Fläche von ihrer Tangentialebene.

2.2 Aktive Konturen

Aktive Konturen, auch Snakes genannt, werden in der digitalen Bildverarbeitung häufig verwendet, um automatisch eine Objektkontur zu finden sowie zu segmentieren. Ein häufiges Anwendungsgebiet bildet hier die medizinische Bildverarbeitung, wo aktive Konturen zum Beispiel zur computergestützten Objektverfolgung in Ultraschallaufnahmen verwendet werden.

Aktive Konturen wurden erstmals 1988 von M. Kass, A. Witkin und D. Terzopoulos in [KWT88] erfolgreich benutzt, um Kanten und Objektübergänge zu finden.

Die Idee der aktiven Konturen ist es, die gesuchte Objektkontur durch eine parametrische Kurve zu beschreiben. Diese wird meist manuell initialisiert und in Abhängigkeit von den sogenannten internen und externen Energien iterativ entwickelt bis die Kurve das gewünschte Objekt segmentiert hat¹ beziehungsweise bis die internen und externen Energien minimal sind. Es ergibt sich also ein Minimierungsproblem.

$$E_{\text{snake}}(\alpha(s)) = E_{\text{ext}}(\alpha(s)) + E_{\text{int}}(\alpha(s)). \tag{2.19}$$

M. Kass, A. Witkin und D. Terzopoulos benutzten in [KWT88] folgendes Modell zur Segmentierung von Kanten, welches hier kurz hergeleitet wird.

Die externe Energie: Die externen Energien berechnen sich aus dem Bildinhalt im Bezug zur Position der Kontur. Hierzu wird häufig eine Form des Gradienten benutzt, um die Kanten des gesuchten Objekts zu finden. Ein Bild wird dabei als Funktion

$$u_0: \Omega \subseteq \mathbb{R}^2 \to \mathbb{R} \tag{2.20}$$

betrachtet, wobei $u_0(x_1, x_2)$ den Grauwert im Punkt (x_1, x_2) angibt. Unter der Vorraussetzung, dass die Grauwertfunktion $u_0(x_1, x_2)$ stetig differenzierbar ist, können die Kanten des Bildes durch den Gradienten $\nabla(u_0(x_1, x_2))$ der Grauwertfunktion beschrieben werden.

Zur Erinnerung: Kanten befinden sich in einem Bild genau an den Stellen, an denen $|\nabla(u_0(x_1,x_2))|$ groß ist.

Um die Kanten eines Bildes zu finden, kann also folgendes Energiefunktional verwendet werden:

$$E_{\text{ext}}(\alpha(s)) = \int -\omega_1 |\nabla(u_0(\alpha(s)))| \, ds \,, \tag{2.21}$$

wobei $\alpha(s)$ die parametrisierte aktive Kontur darstellt. Die Variable ω_1 wird als Gewichtung der externen Energie eingeführt. Es ist leicht einzusehen, dass das

¹In diesem Zusammenhang hat die Kurve ein Objekt segmentiert, wenn die Kurve genau die Kanten des gesuchten Objektes beschreibt.

Energiefunktional $E_{\text{ext}}(\alpha(s))$ klein wird, genau dann, wenn die aktive Kontur $\alpha(s)$ mit den Kanten des Bildes übereinstimmt.

Die interne Energie: Die interne Energie der aktiven Kontur berechnet sich im Gegensatz zur externen Energie aus der aktuellen Form der Snake. Hier bieten sich die Größen Kantenlänge und Krümmung der aktiven Kontur an. Wie in Bemerkung 2.1.0.1 dargestellt, erhält man die Länge der explizit dargestellten Funktion über die erste Ableitung bezüglich des Parameters s. Also wird die Länge über $\frac{\partial \alpha(s)}{\partial s}$ berechnet. Ebenso leicht kann die Krümmung der aktiven Kontur über ihre zweite Ableitung berechnet werden und man erhält so einen Wert, der etwas über die Kantenglätte der aktiven Kontur aussagt. Die Krümmung berechnet sich also gerade durch $\frac{\partial^2 \alpha(s)}{\partial s^2}$. Zusammengefasst ergibt sich folgendes Ergebnis für die beiden Größen:

Kantenlänge von
$$\alpha(s)$$
 lang $\Rightarrow \int \left| \frac{\partial \alpha(s)}{\partial s} \right|^2 ds$ groß, (2.22)

Kanten von
$$\alpha(s)$$
 unglatt $\Rightarrow \int \left| \frac{\partial^2 \alpha(s)}{\partial s^2} \right|^2 ds$ groß. (2.23)

Die Kantenlänge und die Krümmung der aktiven Kontur werden nun mit den Parametern ω_2 und ω_3 gewichtet. Folgendes Energiefunktional wird für die innere Energie der aktiven Kontur definiert:

$$E_{(int)}(\alpha(s)) = \frac{1}{2} \int \omega_2 \left| \frac{\partial \alpha(s)}{\partial s} \right|^2 + \omega_3 \left| \frac{\partial^2 \alpha(s)}{\partial s^2} \right|^2 ds.$$
 (2.24)

Durch die Parameter ω_2 und ω_3 kann Einfluss auf die minimierende aktive Kontur genommen werden. Wählt man $\omega_2 << \omega_3$, so ist das Minimum eine Kontur mit minimaler Krümmung, also eine sehr glatte Kontur. Für den Fall $\omega_3 << \omega_2$ hingegen ergibt sich eine möglichst kurze Kontur mit großen Krümmungen. Offensichtlich gilt nun, dass das Energiefuktional $E_{\rm int}(\alpha(s))$ klein wird, genau dann, wenn die Kontur gut zu den eingestellten Parametern des Segments passt.

Zusammengefasst gilt also für das Energiefunktional in (2.19):

$$E_{\text{snake}}(\alpha(s)) = E_{\text{ext}}(\alpha(s)) + E_{\text{int}}(\alpha(s))$$

$$= \int -\omega_1 |\nabla(u_0(\alpha(s)))| \, ds + \frac{1}{2} \int \omega_2 \left| \frac{\partial \alpha(s)}{\partial s} \right|^2 + \omega_3 \left| \frac{\partial^2 \alpha(s)}{\partial s^2} \right|^2 \, ds \,. \tag{2.25}$$

Der Term $E_{\text{snake}}(\alpha(s))$ soll nun bezüglich $\alpha(s)$ minimiert werden, um die energieminimierende aktive Kontur zu finden. Es ergibt sich also folgendes Optimierungsproblem:

$$E_{\text{snake}}(\alpha(s)) \to \min$$
 (2.26)

2.3. LEVEL SET METHODE

Die Techniken zur Minimierung eines Funktionals stellt uns die Variationsrechnung zur Verfügung. Diese Techniken werden im weiteren Verlauf der Arbeit wiederholt benötigt, so dass diese in Abschnitt 2.4 vorgestellt werden.

Ein offensichtliches Problem der Idee, das gesuchte Segment mit Hilfe einer parametrisierten Kurve darzustellen, ist, dass die Topologie der zu segmentierenden Region im Voraus bekannt sein muss, da die Kurve durch ihre explizite Betrachtung als parametrisierte Kurve während ihrer Entwicklung nicht ihren Zusammenhang verlieren und sich aufteilen kann. Dies könnte unerwünscht sein, wenn zum Beispiel nach Blutgefäßen in der Leber in einem MRT-Bild gesucht wird, weil die Topologie der Blutgefäße im Voraus selten bekannt ist. Als Lösung dieses Problems wurde die Idee der Snakes zunächst an dieses angepasst und es wurden die sogenannten T-Snakes (Topologisch-adaptierte Snakes) entwickelt. Diese werden zum Beispiel in [OP03] von S. Osher und N. Paragios näher beleuchtet.

Ein weiteres Problem, dass auch durch die T-Snakes nicht gelöst wurde, ist die Möglichkeit, das Verfahren ohne eine erhebliche Erhöhung des Rechenaufwandes auf höhere Dimensionen zu verallgemeinern.

S. Osher und J.A. Sethian haben 1988 in [OS88] als erste einen Algorithmus vorgeschlagen bei dem diese beiden Probleme nicht auftreten: die sogenannte Level Set Methode. Hierzu betrachteten sie die Kontur als spezielle Nullstellenmenge einer höherdimensionalen Fläche wie in Abbildung 2.1 dargestellt. In der Praxis wird diese höherdimensionale Fläche meist als Zeitfunktion betrachtet. Somit ist es möglich, dass sich die Kontur mit der Zeit ändert und so die gewünschte Form und Topologie erhält wie in Abbildung 2.2 angedeutet.

2.3 Level Set Methode

Bei der Level Set Methode wird die aktive Kontur als Nullstellenmenge einer höherdimensionalen Funktion φ dargestellt. Dadurch wird die Kontur in einem festen Eulerschen Koordinatensystem implizit dargestellt ohne eine explizite Parametrisierung zu verwenden. Im Folgenden wird die Level Set Methode, analog zu S. Osher und J.A. Sethian in [OS88], zunächst für den zweidimensionalen Fall eingeführt und im weiteren Verlauf auf den dreidimensionalen Fall erweitert.

Hierzu sei vorerst $\Omega \subset \mathbb{R}^2$ offen und beschränkt sowie $\partial\Omega$ der Rand von Ω . Ein zweidimensionales Bild wird wieder durch die Funktion $u_0 : \Omega \to \mathbb{R}$ beschrieben. Sei nun $\omega \subseteq \Omega$ offen. Dann betrachte die aktive Kontur C als Rand der offenen Teilmenge ω . Also $C = \partial\omega$.

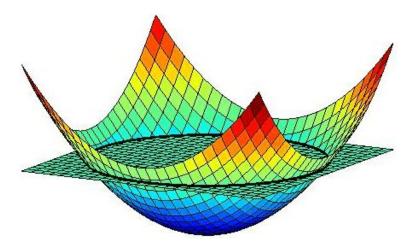


Abbildung 2.1: Die Kontur als Schnitt- beziehungsweise Nullstellenmenge einer höherdimensionalen Fläche mit dem Bild.

Wie oben bereits erwähnt ist die Idee der Level Set Methode, die aktive Kontur C als Nullstellenmenge einer höherdimensionalen Funktion φ zu betrachten. Dazu wird φ wie folgt definiert:

$$\varphi: \quad \Omega \to \mathbb{R}$$

$$(x_1, x_2) \mapsto \pm d,$$

$$(2.27)$$

wobei d den Abstand von $(x_1, x_2) \in \Omega$ zum Rand $\partial \omega$ zum Zeitpunkt t = 0 beschreibt. Das Vorzeichen von d wird positiv gewählt, wenn der Punkt (x_1, x_2) außerhalb von ω liegt und negativ, falls der Punkt innerhalb von ω liegt. Die Vorteile dieser Methode, die aktive Kontur als Nullstellenmenge einer höherdimensionalen Fläche zu betrachten, sind vielfältig. Zum einen kann, wie schon beschrieben, auf eine Parametrisierung der Kontur verzichtet und sie somit implizit betrachtet werden. Dadurch ist es möglich, dass sich die Topologie der Kontur im Verlauf ändert und so die gewünschte Form erhält. Zum anderen ergeben sich aus der Darstellung der Funktion $\varphi(x_1, x_2) = \pm d$ einige geometrische Eigenschaften. So lassen sich zum Beispiel der Normalenvektor und die Krümmung eines Punktes $x_0 = (x_1, x_2) \in C$ leicht berechnen. Der Normalenvektor eines Punktes $x_0 \in C$ berechnet sich als normierter Gradient der vorzeichenbehafteten Abstandsfunktion. Also durch:

$$N(x_0) = \frac{\nabla \varphi(x_0)}{|\nabla \varphi(x_0)|}, \qquad (2.28)$$

wobei der Nabla-Operator

$$\nabla \varphi(x_0) = \left(\frac{\partial \varphi}{\partial x_1}(x_0), \frac{\partial \varphi}{\partial x_2}(x_0)\right)^T \tag{2.29}$$

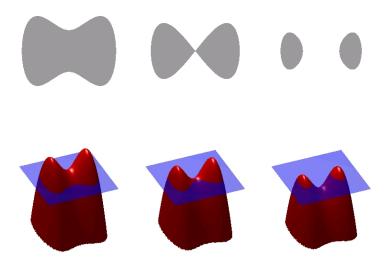


Abbildung 2.2: Evolution der aktiven Kontur (oben dargestellt) als Schnitt der höherdimensionalen Funktion (rot) mit dem Bild (blau) zu verschiedenen Zeitpunkten.

Quelle: [Wik].

den Gradienten von φ im Punkt x_0 liefert. Ebenso einfach lässt sich die Krümmung k im Punkt $x_0 \in C$ wie folgt berechnen:

$$k(x_0) = \operatorname{div} \left(\frac{\nabla \varphi(x_0)}{|\nabla \varphi(x_0)|} \right). \tag{2.30}$$

Bei einer zweidimensionalen Kontur und einer dreidimensionalen Funktion $\varphi(x_1, x_2)$ ergibt sich für die Krümmung

$$k = \frac{\varphi_{x_1 x_1} \varphi_{x_2}^2 - 2\varphi_{x_1} \varphi_{x_2} \varphi_{x_1 x_2} + \varphi_{x_2 x_2} \varphi_{x_1}^2}{(\varphi_{x_1}^2 + \varphi_{x_2}^2)^{\frac{2}{3}}},$$
(2.31)

wobei $\varphi_{x_i x_j} = \frac{\partial^2 \varphi}{\partial x_i \partial x_j}$ die partielle Ableitung beschreibt.

Ein weiterer wichtiger Vorteil ist, wie im Folgenden dargestellt, dass sich diese Methode erweitern lässt. Somit ist es leicht möglich, auch Konturen in höheren Dimensionen zu beschreiben. Sei nun also $\Omega \in \mathbb{R}^3$ offen und beschränkt sowie $\partial \Omega$ der Rand von Ω . Die Volumendaten, zum Beispiel Computer-Tomographiedaten oder Magnet-Resonanz-Tomographiedaten, werden durch die Funktion

$$u_0: \Omega \to \mathbb{R}$$
 (2.32)

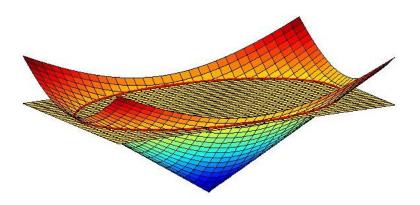


Abbildung 2.3: Die vorzeichenbehaftete Abstandsfunktion eines Kreises.

beschrieben. Sei nun $\omega \subseteq \Omega$ offen. Dann wird die aktive Kontur C als Rand der offenen Teilmenge ω betrachtet. Also $C = \delta \omega$. Analog zum zweidimensionalen Fall wird wieder die höherdimensionale Funktion $\varphi(x_1, x_2, x_3)$ definiert.

Definition 2.3.0.11

Es sei

$$\varphi: \quad \Omega \to \mathbb{R}$$

$$(x_1, x_2, x_3) \mapsto \pm d,$$

$$(2.33)$$

wobei d den Abstand von einem Punkt (x_1, x_2, x_3) zur Kontur C beschreibt. Wiederum wird das Vorzeichen von d positiv für $(x_1, x_2, x_3) \in \Omega \setminus \overline{\omega}$ gewählt und negativ, falls $(x_1, x_2, x_3) \in \omega$ gilt. φ wird in diesem Fall als vorzeichenbehaftete Abstandsfunktion bezeichnet.

Bemerkung 2.3.0.4

Durch die Definition der vorzeichenbehafteten Abstandsfunktion gilt:

$$(x_1, x_2, x_3) \in C \Leftrightarrow \varphi(x, y, z) = 0. \tag{2.34}$$

Bemerkung 2.3.0.5

Wie im zweidimensionalen Fall ergeben sich auch im dreidimensionalen Fall folgende geometrische Eigenschaften:

$$N(x_0) = \frac{\nabla \varphi(x_0)}{|\nabla \varphi(x_0)|} \tag{2.35}$$

beschreibt die Flächennormale in einem Punkt $x_0 = (x_1, x_2, x_3)^T \in C$ und

$$k(x_0) = \operatorname{div} \left(\frac{\nabla \varphi(x_0)}{|\nabla \varphi(x_0)|} \right)$$
 (2.36)

2.4. VARIATIONSRECHNUNG

beschreibt die mittlere Krümmung der Fläche in einem Punkt $x_0 \in C$.

Wie oben bereits erwähnt, wird die vorzeichenbehaftete Abstandsfunktion $\varphi(x_1, x_2, x_3)$ als Zeitfunktion betrachtet, um die Bewegung der aktiven Kontur zu beschreiben. Hierzu wird die Funktion φ durch eine zeitliche Komponente erweitert:

$$\varphi: \Omega \times \mathbb{R} \to \mathbb{R} \,. \tag{2.37}$$

Als nächstes wird die Gleichung, die die Evolution der aktiven Kontur C beschreibt definiert. Wie in [MOS90] von W. Mulder, S. Osher und J.A. Sethian für den zweidimensionalen Fall beschrieben, ergibt sich folgende partielle Differentialgleichung, wenn die Fläche in Richtung ihrer Flächennormale evaluiert werden soll:

$$\frac{\partial \varphi}{\partial t} + F|\nabla \varphi| = 0, \qquad (2.38)$$

wobei F die Ausbreitungsgeschwindigkeit darstellt. Hierauf wird im nächsten Kapitel beim sogenannten "edge stopping" Verfahren noch explizit eingehen.

2.4 Variationsrechnung

Wie schon erwähnt, werden im Verlauf der Arbeit Minimierungsprobleme der Form

$$\min_{A} \mathcal{E}(a(x)) \tag{2.39}$$

gelöst. Das heißt, aus einer Menge A von Funktionen wird diejenige gesucht die ein gegebenes Energiefunktional $\mathcal{E}:A\to\mathbb{R}$ minimiert. Probleme dieser Art werden mit Hilfe der Variationsrechnung gelöst. Die Variationsrechnung wurde 1744 von Leonhard Euler ins Leben gerufen, als er die berühmte "Euler-Gleichung" bewies, welche 1762 von Lagrange für Funktionen mehrerer Veränderlicher verallgemeinert wurde.

Nachfolgend wird lediglich die Euler-Lagrange Gleichung vorgestellt und ihr Zusammenhang mit dem Minimierungsproblem kurz erläutert, da eine vollständige Einführung in die Thematik der Variationsrechnung an dieser Stelle zu umfangreich wäre. Für ein tiefer gehendes Verständnis sei der interessierte Leser auf die entsprechende Spezialliteratur verwiesen, wie beispielsweise [HSZ03] von W. Hackbusch, H.R. Schwarz und E. Zeidler oder [Her04] von N. Herrmann.

Sei also $\Omega \subseteq \mathbb{R}^N, \, a:\Omega \to \mathbb{R}$ und

$$\mathcal{E}(a) = \int_{\Omega} E(a(x), \nabla a(x), x) dx \quad \text{mit } x \in \Omega.$$
 (2.40)

Gesucht ist nun diejenige Funktion a, die das Funktional $\mathcal{E}(a)$ minimiert. Für ein Minimum α des Funktionals gilt nach den Erkenntnissen von Euler und Lagrange notwendigerweise die sogenannte Euler-Lagrange Gleichung:

$$0 = \sum_{j=1}^{N} \frac{\partial}{\partial x_j} \frac{\partial E}{\partial (\partial_j a)}(A) - \frac{\partial E}{\partial a}(A), \qquad (2.41)$$

wobei $\partial_j := \frac{\partial}{\partial x_j}$ und $A = (a, \nabla a, x)$ gilt.

Bemerkung 2.4.0.6

Die Euler-Lagrange Gleichung stellt nur ein notwendiges Kriterium für das Minimierungsproblem (2.39) dar. Sie ist jedoch äquivalent zu dem folgenden Problem:

$$\mathcal{E}(a(x)) = \text{station}\ddot{a}r.$$
 (2.42)

Mit Hilfe der Euler-Lagrange Gleichung kann später ein iteratives Verfahren für Energieminimierende aktive Konturen zu beschreiben werden.

2.4. VARIATIONSRECHNUNG

Kapitel 3

Modelle und Methoden der Segmentierung

Im Vordergrund dieses Kapitels stehen die beiden in der Einleitung erwähnten Segmentierungsverfahren, das sogenannte "edge stopping" Verfahren sowie der "energy minimization" Algorithmus. Zunächst werden diese beiden Verfahren in Anlehnung an die Arbeiten [OS88] von S. Osher und J.A. Sethian sowie [CV99] von T. Chan und L. Vese hergeleitet und vorgestellt. Weiterhin werden in diesem Kapitel die benötigten numerischen Methoden beschrieben, die zum Segmentieren von Volumendaten erforderlich sind.

Nach der Einführung der beiden Verfahren werden in Abschnitt 3.3 die Änderungen, die ich im Laufe der Diplomarbeit entworfen und getestet habe, beschrieben.

3.1 Das "edge stopping" Verfahren

Wie in 2.3 beschrieben, geht man von einer dreidimensionalen vorzeichenbehafteten Abstandsfunktion $\varphi:\Omega\subseteq\mathbb{R}^3\to\mathbb{R}$ aus und erweitert diese um eine zeitliche Komponente. Dementsprechend ist φ definiert als:

$$\varphi: \quad \Omega \times \mathbb{R} \to \mathbb{R}$$

$$(x_1, x_2, x_3, t) \mapsto \pm d.$$

$$(3.1)$$

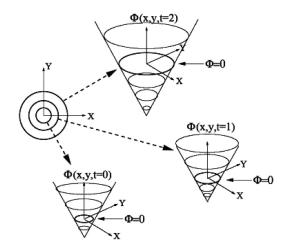


Abbildung 3.1: Einbettung der aktiven Kontur C in die vorzeichenbehaftete Distanzfunktion zu verschiedenen Zeitpunkten. Quelle: [Set99a] von J.A. Sethian.

Bemerkung 3.1.0.7

Zu jedem Zeitpunkt t der Evolution ist die aktive Kontur C als Nullstellenmenge in $\varphi(x_1, x_2, x_3, t)$ eingebettet. Es gilt in jedem Iterationsschritt:

$$C_t = \{(x_1, x_2, x_3) \mid (x_1, x_2, x_3) \in \Omega \text{ und } \varphi(x_1, x_2, x_3, t) = 0\}.$$
 (3.2)

 C_t wird auch als Zero Level Set bezeichnet. Abbildung 3.1 verdeutlicht noch einmal die Einbettung der aktiven Kontur in die vorzeichenbehaftete Distanzfunktion.

Ziel ist es also, die Funktion $\varphi(x_1, x_2, x_3, t)$ iterativ zu entwickeln, so dass am Ende der Iteration das Zero Level Set die Grenzen des gesuchten Segments beschreibt. Hierzu wird zunächst die Evolutionsgleichung (2.38) hergeleitet. Dazu sei $w_x(t)$ mit $t \in [0, \infty)$ der Weg, den ein Punkt auf der aktiven Kontur während der Entwicklung zurücklegt. Da der Punkt während der Entwicklung zu jeder Zeit auf der Kontur C liegt, gilt:

$$\varphi(w_x(t), t) = 0. \tag{3.3}$$

Die Änderung der vorzeichenbehafteten Abstandsfunktion $\varphi(x_1, x_2, x_3, t)$ werden mit der Ableitung nach der Zeit t bestimmt. Mit der Kettenregel folgt wie in [MOS90] also:

$$\frac{\partial \varphi}{\partial t} + \frac{\partial}{\partial t} w_x \cdot \nabla \varphi = 0. \tag{3.4}$$

Um die Kontur C in Richtung ihrer Einheitsnormalen N zu entwickeln, muss nach der Kurventheorie gelten:

$$\frac{\partial}{\partial t}w_x(t) = F \cdot N. \tag{3.5}$$

Mit (2.35) folgt also

$$\frac{\partial}{\partial t}w_x(t) = F \cdot \frac{\nabla \varphi}{|\nabla \varphi|}.$$
(3.6)

F bezeichnet auch hier wieder die Geschwindigkeit der Evolution von C. Wird nun die Gleichung (3.6) in Gleichung (3.4) eingesetzt, so erhält man:

$$\frac{\partial \varphi}{\partial t} + F \cdot \frac{\nabla \varphi}{|\nabla \varphi|} \nabla \varphi = \frac{\partial \varphi}{\partial t} + F|\nabla \varphi| = 0. \tag{3.7}$$

Da die Evolution der aktiven Kontur C hergeleitet wurde, wird als nächstes die Geschwindigkeit F der Evolution definiert. In diesem Zusammenhang bezeichnet man F auch als "Speed Funktion" oder "Ausbreitungsgeschwindigkeit". Diesbezüglich wird F, wie von R. Malladi, J.A. Sethian und B. Vemuri in [MSV95] vorgeschlagen, in zwei Komponenten unterteilt:

$$F = F_A + F_G. (3.8)$$

 F_A dient hier als konstanter Term, der die aktive Kontur je nach Vorzeichen zur stetigen Ausbreitung beziehungsweise Verkleinerung anregt. Der Term F_G hängt von der aktuellen Form der Kontur ab und sorgt für eine Glättung der Kontur in Regionen mit einer starken mittleren Krümmung.

Wie R. Malladi, J.A. Sethian und B. Vemuri in [MSV95] ist der konstanten Term definiert als:

$$F_A = 1. (3.9)$$

Der Term F_G , der die Kontur glättet, wird analog zu M. Wasilewski in [Wasa] in Abhängigkeit von der mittleren Krümmung k und einer kleinen negativen Konstante ϵ definiert. Also:

$$F_G = -\epsilon k \,. \tag{3.10}$$

Somit wird die Ausbreitungsgeschwindigkeit der Kontur in Regionen mit einer hohen mittleren Krümmung verringert. Für die Evolution der aktiven Kontur erhält man also zunächst:

$$\frac{\partial \varphi}{\partial t} + (1 - \epsilon k) |\nabla \varphi| = 0. \tag{3.11}$$

In Analogie zu den "klassischen" aktiven Konturen kann die Ausbreitungsgeschwindigkeit beziehungsweise die "Speed Funktion" auch als interne Kraft interpretiert werden, die nur von der aktuellen Form der aktiven Kontur abhängt. Nun wird die Ausbreitungsgeschwindigkeit noch an die Volumendaten gekoppelt. Dazu wird,

3.1. DAS "EDGE STOPPING" VERFAHREN

wie der Name des Verfahrens bereits sagt, eine stopping Funktion $g(|\nabla u_0|)$ definiert, die die Ausbreitungsgeschwindigkeit verringert, falls die Kanten des gesuchten Segmentes erreicht wurden. Die Funktion wird in Abhängigkeit des Gradienten ∇u_0 definiert, da die Kanten in den Volumendaten durch einen hohen Gradienten gekennzeichnet sind. Allgemein muss die Funktion $g(|\nabla u_0|)$ also folgende Eigenschaften haben:

- auf dem Intervall $[0, \infty)$ muss die Funktion $g(|\nabla u_0|)$ positiv und regulär sein,
- die Funktion $g(|\nabla u_0|)$ muss auf dem Intervall $[0, \infty)$ monoton fallend sein, also muss gelten $g'(|\nabla u_0|) \leq 0$,
- da die Funktion auf dem Intervall $[0, \infty)$ also positiv und monoton fallend ist, muss gelten $\lim_{t\to\infty} g(t) = 0$.

Die Funktion $g(|\nabla u_0|)$ kann also zum Beispiel durch

$$g(|\nabla u_0|) = \frac{1}{1 + |\nabla(G_\sigma(x_1, x_2, x_3) \star u_0(x_1, x_2, x_3))|^p}, \text{ mit } p \ge 1$$
 (3.12)

oder durch

$$g(|\nabla u_0|) = e^{-|\nabla(G_\sigma(x_1, x_2, x_3) \star u_0(x_1, x_2, x_3))|}$$
(3.13)

definiert werden. Hierbei bezeichnet $G_{\sigma}(x,y,z) \star u_0(x,y,z)$ die Faltung der Volumendaten mit einem Gaussfilter¹. Dies bedeutet, dass die Volumendaten geglättet werden, um Rauschen und Voxelfehler zu minimieren. Beide Funktionen erfüllen die Eigenschaften einer allgemeinen stopping Funktion, dass heißt, sie haben in Regionen mit hohem Gradienten Werte nahe Null und in homogenen Regionen Werte nahe Eins. Die aktive Kontur breitet sich somit in homogenen Regionen mit "normaler" Geschwindigkeit aus, wohingegen die Kontur an den Kanten eines Objekts stark gebremst wird. Die zweite Funktion (3.13) fällt im Gegensatz zur ersten Funktion (3.12) deutlich schneller gegen Null. Demzufolge werden auch schwächere Kanten erkannt und die Ausbreitungsgeschwindigkeit in diesen Regionen verringert. Jedoch ist die Funktion (3.13) dadurch auch anfälliger für Bildstörungen wie Rauschen und Voxelfehler.

Die medizinischen Volumendaten zeichnen sich im Allgemeinen dadurch aus, dass sie einen großen² Grauwertbereich haben. Deshalb werden die beiden *stopping*

¹Der Gaussfilter ist definiert durch: $G_{\sigma} = \sigma^{-\frac{1}{2}} e^{-\frac{|x^2+y^2+z^2|}{4\sigma}}$. An dieser Stelle können auch andere Filter, wie der Median-Filter, verwendet werden. Dem Leser sei hier [Jäh05] von B. Jähne empfohlen.

²In diesem Zusammenhang bezeichnet ein großer Grauwertbereich mehr als 256 verschiedene Grauwerte.

KAPITEL 3. MODELLE UND METHODEN DER SEGMENTIERUNG

Funktionen auf den visuellen Grauwertbereich der angezeigten Volumendaten angepasst:

$$g_1(|\nabla u_0|) = \frac{1}{1 + |\nabla(G_{\sigma}(x) \star vis(u_0(x)))|^p}, \text{ mit } p \ge 1$$
 (3.14)

und

$$g_2(|\nabla u_0|) = e^{-|\nabla(G_\sigma(x) \star vis(u_0(x)))|},$$
 (3.15)

wobei $x = (x_1, x_2, x_3)$ gilt und die Funktion $vis(u_0)$ die Transformation der Volumendaten auf den visuellen Bereich beschreibt.

Die stopping Funktion wird nun mit der Ausbreitungsgeschwindigkeit $F = F_A + F_G$ durch Multiplikation zusammengeführt, so dass folgende Gleichung die Evolution der aktiven Kontur beschreibt:

$$\frac{\partial \varphi}{\partial t} + g(|\nabla u_0|)(1 - \epsilon k)|\nabla \varphi| = 0$$

$$\Leftrightarrow \frac{\partial \varphi}{\partial t} = -g(|\nabla u_0|)(1 - \epsilon k)|\nabla \varphi|.$$
(3.16)

Um die Veränderung der aktiven Kontur C zu berechnen, muss diese partielle Differentialgleichung gelöst werden. Hierzu muss der Anteil $F_G = -\epsilon k$ zunächst erweitert werden, weil die Krümmung nur für Punkte auf der Kontur, also auf dem Zero Level Set, sinnvoll definiert ist. Diesbezüglich wird folgenden Ansatz von R. Malladi, J.A. Sethian und B. Vemuri aus [MSV95] benutzt, um eine erweiterte Ausbreitungsgeschwindigkeit zu definieren.

Definition 3.1.0.12

Sei $Q \notin C$, also $\varphi(Q) \neq 0$ und $P \in C$ der Punkt auf der Kontur mit dem kleinsten Abstand zu Q, also $\varphi(P)=0$ und

$$P = \min_{R \in C} |R - Q|. \tag{3.17}$$

Dann definiere die erweiterte Ausbreitungsgeschwindigkeit durch

$$F_G(Q) = F_G(P). (3.18)$$

Analog dazu ist die erweiterte stopping Funktion $g(|\nabla u_o|)$ definiert.

In dieser Arbeit wird durchgängig mit diskreten Volumendaten gearbeitet, so dass noch geeignete Hilfsmittel benötigt werden, die im Folgenden vorgestellt werden. Hierzu sei also

$$\hat{u_0}: \quad \hat{\Omega} \to G$$

$$x \mapsto \hat{u_0}(x)$$
(3.19)

3.1. DAS "EDGE STOPPING" VERFAHREN

ein digitales Bild³. Wobei $\hat{\Omega} = \{0, \dots, n_{x_1}\} \times \{0, \dots, n_{x_2}\} \times \{0, \dots, n_{x_3}\}$ die Dimensionen des Bildraums, $x \in \hat{\Omega}$ einen diskreten Koordinatenvektor und $G \in \{0, \dots, x_G\}$ den diskreten Grauwert beschreibt. Um die vorzeichenbehaftete Abstandsfunktion $\varphi(x_1, x_2, x_3, t)$ zu diskretisieren sei Δt der Zeitschritt und $(x_1, x_2, x_3) \in \hat{\Omega}$ die Gitterpunkte der Volumendaten.

$$\varphi_{x_1, x_2, x_3}^n = \varphi(x_1, x_2, x_3, n\Delta t) \tag{3.20}$$

für $n \in \mathbb{N}$, $n \ge 0$ bezeichnet im Folgenden eine Approximation von $\varphi(x_1, x_2, x_3, t)$. Die Differnzenquotienten werden mit Hilfe der Finiten-Differenzenquotienten approximiert.

Definition 3.1.0.13

Mit obiger Notation sind die Finiten-Differenzenquotienten wie folgt definiert:

$$\Delta_{+}^{x_{1}} \varphi_{x_{1},x_{2},x_{3}} := \varphi_{x_{1}+1,x_{2},x_{3}} - \varphi_{x_{1},x_{2},x_{3}}
\Delta_{+}^{x_{2}} \varphi_{x_{1},x_{2},x_{3}} := \varphi_{x_{1},x_{2}+1,x_{3}} - \varphi_{x_{1},x_{2},x_{3}}
\Delta_{+}^{x_{3}} \varphi_{x_{1},x_{2},x_{3}} := \varphi_{x_{1},x_{2},x_{3}+1} - \varphi_{x_{1},x_{2},x_{3}}$$
(3.21)

beschreiben die Finiten-Vorwärtsdifferenzenquotienten nach x_1, x_2, x_3 und

$$\Delta_{-}^{x_{1}} \varphi_{x_{1},x_{2},x_{3}} := \varphi_{x_{1},x_{2},x_{3}} - \varphi_{x_{1}-1,x_{2},x_{3}}
\Delta_{-}^{x_{2}} \varphi_{x_{1},x_{2},x_{3}} := \varphi_{x_{1},x_{2},x_{3}} - \varphi_{x_{1},x_{2}-1,x_{3}}
\Delta_{-}^{x_{3}} \varphi_{x_{1},x_{2},x_{3}} := \varphi_{x_{1},x_{2},x_{3}} - \varphi_{x_{1},x_{2},x_{3}-1}$$
(3.22)

beschreiben die Finiten-Rückwärtsdifferenzenquotienten.

Zusätzlich werden in dieser Arbeit sowohl beim "edge stopping" Verfahren als auch beim später vorgestellten Verfahren der Energieminimierung die sogenannten zentralen Differenzenquotienten benötigt.

Definition 3.1.0.14

Mit obiger Notation beschreiben

$$\Delta^{x_1} \varphi_{x_1, x_2, x_3} := \frac{1}{2} \cdot (\varphi_{x_1 + 1, x_2, x_3} - \varphi_{x_1 - 1, x_2, x_3})$$

$$\Delta^{x_2} \varphi_{x_1, x_2, x_3} := \frac{1}{2} \cdot (\varphi_{x_1, x_2 + 1, x_3} - \varphi_{x_1, x_2 - 1, x_3})$$

$$\Delta^{x_3} \varphi_{x_1, x_2, x_3} := \frac{1}{2} \cdot (\varphi_{x_1, x_2, x_3 + 1} - \varphi_{x_1, x_2, x_3 - 1})$$
(3.23)

die zentralen Differenzenquotienten.

 $^{^3}$ Ein digitales Bild bezeichnet die Digitalisierung eines Bildes u_0 durch Abtastung und Quantisierung. Die Digitalisierung eines Bildes ist stark abhängig vom gewählten Aufnahmegerät.

Bemerkung 3.1.0.8

Die in dieser Arbeit benötigten höheren partiellen Ableitungen ergeben sich durch hintereinander ausführen der zentralen Differenzenquotienten. Man erhält zum Beispiel:

$$\Delta^{x_1 x_1} \varphi_{x_1, x_2, x_3} = \frac{1}{4} \cdot (\varphi_{x_1 + 2, x_2, x_3} - \varphi_{x_1 - 2, x_2, x_3} - 2 \cdot \varphi_{x_1, x_2, x_3})$$

$$\Delta^{x_1 x_2} \varphi_{x_1, x_2, x_3} = \frac{1}{4} \cdot (\varphi_{x_1 + 1, x_2 + 1, x_3} - \varphi_{x_1 + 1, x_2 - 1, x_3})$$

$$- \varphi_{x_1 - 1, x_2 + 1, x_3} + \varphi_{x_1 - 1, x_2 - 1, x_3}).$$
(3.24)

Die diskretisierte vorzeichenbehaftete Abstandsfunktion φ_{x_1,x_2,x_3}^n wird im Laufe der Iteration selten exakt Null sein. Daher ist ein geeignetes Hilfsmittel nötig, um zu entscheiden, wann ein Voxel zum Zero Level Set, also zur Kontur C gehört. Dies ist notwendig, damit das Zero Level Set der Funktion zum Zeitpunkt n approximiert werden kann. Die in [Wasa] von M. Wasilewski vorgestellte Methode definiert einen Voxel als Teilmenge vom Zero Level Set, wenn es einen Vorzeichenwechsel in positiver Richtung der Abstandsfunktion in einer 26er-Nachbarschaft gibt. Mathematisch also:

Definition 3.1.0.15

Für einen Voxel $(x_1, x_2, x_3) \in \hat{\Omega}$ gilt: $(x_1, x_2, x_3) \in C_n$ genau dann, wenn

$$\max \left[\varphi_{(x_{1},x_{2},x_{3})}^{n}, \varphi_{(x_{1}+1,x_{2},x_{3})}^{n}, \varphi_{(x_{1},x_{2}+1,x_{3})}^{n}, \varphi_{(x_{1},x_{2},x_{3}+1)}^{n}, \varphi_{(x_{1}+1,x_{2}+1,x_{3})}^{n}, \varphi_{(x_{1}+1,x_{2},x_{3}+1)}^{n}, \varphi_{(x_{1},x_{2}+1,x_{3}+1)}^{n}, \varphi_{(x_{1},x_{2}+1,x_{3}+1)}^{n} \right] > 0$$

$$(3.25)$$

und

$$\min \left[\varphi_{(x_{1},x_{2},x_{3})}^{n}, \varphi_{(x_{1}+1,x_{2},x_{3})}^{n}, \varphi_{(x_{1},x_{2}+1,x_{3})}^{n}, \right. \\
\left. \varphi_{(x_{1},x_{2},x_{3}+1)}^{n}, \varphi_{(x_{1}+1,x_{2}+1,x_{3})}^{n}, \varphi_{(x_{1}+1,x_{2},x_{3}+1)}^{n}, \right. \\
\left. \varphi_{(x_{1},x_{2}+1,x_{3}+1)}^{n}, \varphi_{(x_{1}+1,x_{2}+1,x_{3}+1)}^{n} \right] < 0.$$
(3.26)

Wählt man zum Beispiel die Kanten eines Objekts als Startkontur und berechnet die vorzeichenbehaftete Distanzfunktion, so ergibt sich mit der obigen Definition des Zero Level Sets die in Abbildung 3.2 (b) dargestellte aktive Kontur C_0 . Unterabbildung (e) zeigt eine Vergrößerung des entstandenen Zero Level Sets. Deutlich zu erkennen ist, dass mit der obigen Definition auch Pixel außerhalb des Objekts zum Zero Level Set gehören. Hierdurch werden zum einen die Kanten des Objekts schneller übersprungen und zum anderen entfernt sich die Funktion φ schneller

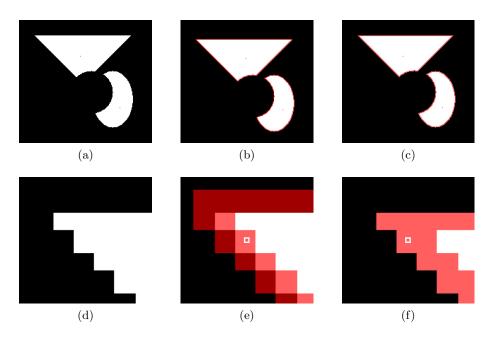


Abbildung 3.2: Verschiedene Definitionen des Zero Level Sets.

von der initialisierten vorzeichenbehafteten Distanzfunktion. Die Funktion stellt also keine vorzeichenbehaftete Abstandsfunktion mehr dar, wodurch die geometrischen Vorteile dieser Darstellung entfallen und es häufiger nötig ist, die Funktion zu reinitialisieren. Deshalb wird in dieser Arbeit folgende Definition der Zero Level Set Menge verwendet:

Definition 3.1.0.16

Für einen Voxel $(x_1, x_2, x_3) \in \hat{\Omega}$ gilt: $(x_1, x_2, x_3) \in C_n$ genau dann, wenn

$$\varphi(x_1, x_2, x_3, t) <= 0 \tag{3.27}$$

und für mindestens einen Nachbarn $(y_1, y_2, y_3) \in \hat{\Omega}$ von (x_1, x_2, x_3) bezüglich einer 26er-Nachbarschaft gilt:

$$\varphi(y_1, y_2, y_3, t) > 0. (3.28)$$

Die Abbildungen 3.2 (c) und (f) machen deutlich, dass die beschriebenen Voxel, ausserhalb des Objekts nicht mehr im Zero Level Set auftreten.

Die in dieser Arbeit benötigten Hilfsmittel, um mit diskreten Volumendaten zu arbeiten, sind nun bekannt. Folglich wird weiter auf die Evolutionsgleichung der aktiven Kontur eingegangen.

KAPITEL 3. MODELLE UND METHODEN DER SEGMENTIERUNG

Erinnerung: Folgende partielle Differentialgleichung beschreibt die Evolution der aktiven Kontur:

 $\frac{\partial \varphi}{\partial t} = -g(|\nabla \hat{u}_0|)(1 - \epsilon k)|\nabla \varphi|), \qquad (3.29)$

wobei $g(|\nabla \hat{u_0}|)$ die erweiterte stopping Funktion und $(1 - \epsilon k)|\nabla \varphi|$ die erweiterte Ausbreitungsgeschwindigkeit wie in (3.1.0.12) darstellt.

Mit der manuell initialisierten Startkontur und der zugehörigen vorzeichenbehafteten Abstandsfunktion φ_0 muss also eine partielle Differentialgleichung 1. Ordnung mit der Anfangsbedingung $\varphi(x_1, x_2, x_3, 0) = \varphi_0$ gelöst werden. Hier bietet sich das einfache eulersche Polygonzugverfahren an.

Definition 3.1.0.17

Gegeben sei das Anfangswertproblem

$$\frac{\partial y}{\partial x} = y' = f(x, y), \quad y(x_0) = y_0. \tag{3.30}$$

Um eine Näherung y_n für den Wert $y(x_n)$ der Lösung y im Punkt $x_n = x_{n-1} + h$ zu einer gegebenen Schrittweite h zu bestimmen, lautet das eulersche Polygonzugverfahren oder auch explizite Euler-Verfahren wie folgt:

$$\frac{y_n - y_{n-1}}{h} = f(x_{n-1}, y_{n-1}). (3.31)$$

Die Approximation des Wertes $y(x_n)$ wird also iterativ berechnet.

Angewendet auf die Evolutionsgleichung (3.29) erhält man also folgende Gleichung, um die Evolution der aktiven Kontur numerisch zu approximieren:

$$\frac{\varphi_{x_{1},x_{2},x_{3}}^{n+1} - \varphi_{x_{1},x_{2},x_{3}}^{n}}{\Delta t} = -g(|\nabla \hat{u_{0}}||(1 - \epsilon k)|\nabla \varphi|))$$

$$\Leftrightarrow \varphi_{x_{1},x_{2},x_{3}}^{n+1} = \varphi_{x_{1},x_{2},x_{3}}^{n} - \Delta t \left[g(|\nabla \hat{u_{0}}||(1 - \epsilon k)|\nabla \varphi|)\right].$$
(3.32)

Nun sind noch die richtigen Hilfsmittel nötig, um die beiden Terme $F_A|\nabla\varphi|$ und $F_G|\nabla\varphi|$ der Ausbreitungsgeschwindigkeit numerisch zu berechnen.

Der konstante Term $F_A|\nabla\varphi|$ mit $F_A=1$, der die aktive Kontur zur stetigen Ausbreitung anregen soll, wird, wie von J.A. Sethian und J. Strain in [SJ92] vorgeschlagen, mit Hilfe der sogenannten *upwind schemes* berechnet, da die Berechnung des Terms mit Finiten Differenzen dazu neigt, zu oszillieren. Man erhält also:

$$F_{A}|\nabla\varphi| = \left[\max(\Delta_{-}^{x_{1}}\varphi_{x_{1},x_{2},x_{3}}^{n},0)^{2} + \min(\Delta_{+}^{x_{1}}\varphi_{x_{1},x_{2},x_{3}}^{n},0)^{2} + \max(\Delta_{-}^{x_{2}}\varphi_{x_{1},x_{2},x_{3}}^{n},0)^{2} + \min(\Delta_{+}^{x_{2}}\varphi_{x_{1},x_{2},x_{3}}^{n},0)^{2} + \max(\Delta_{-}^{x_{3}}\varphi_{x_{1},x_{2},x_{3}}^{n},0)^{2} + \min(\Delta_{+}^{x_{3}}\varphi_{x_{1},x_{2},x_{8}}^{n},0)^{2}\right]^{\frac{1}{2}}.$$

$$(3.33)$$

Listing 3.1: Global Extension Algorithmus

```
Initialize \varphi_{x_1,x_2,x_3}^0 = \varphi_0, n = 0

for fixed number of iterations do

for each (x_1,x_2,x_3) \in C_n do

Compute F_A |\nabla \varphi| + F_G |\nabla \varphi| by the equations (3.33) and (3.34)

Multiply F_A |\nabla \varphi| + F_G |\nabla \varphi| by g(|\nabla \hat{u_0}|)

end

for each (x_1,x_2,x_3) \notin C_n do

Find Point (y_1,y_2,y_3) \in C closest to (x_1,x_2,x_3)

Extend F_A |\nabla \varphi| + F_G |\nabla \varphi| and g(|\nabla \hat{u_0}|)

end

Compute \varphi^{n+1} by (3.32)

n = n+1
```

Der Term $F_G|\nabla\varphi| = -\epsilon k|\nabla\varphi|$, der für die Glättung der aktiven Kontur sorgt, wird mit Hilfe der zentralen Differenzenquotienten wie folgt berechnet:

$$F_G|\nabla\varphi| = -\epsilon k \left[(\Delta^{x_1}\varphi_{x_1,x_2,x_3}^n)^2 + (\Delta^{x_2}\varphi_{x_1,x_2,x_3}^n)^2 + (\Delta^{x_3}\varphi_{x_1,x_2,x_3}^n)^2 \right]^{\frac{1}{2}}.$$
 (3.34)

Der Algorithmus zur Bestimmung der Veränderung der aktiven Kontur arbeitet wie in Listing 3.1 dargestellt.

Dem Algorithmus ist zu entnehmen, dass in jeder Iteration offensichtlich alle Voxel der Volumendaten betrachtet werden. Demzufolge werden auch Voxel, die weit von der Kontur entfernt sind, in jeder Iteration aktualisiert. Da sich die Nullstellenmenge beziehungsweise das Zero Level Set der aktiven Kontur bei jeder Iteration jedoch nur in einer kleinen Umgebung des Zero Level Sets ändert, genügt es nur die Voxel in der unmittelbaren Umgebung des aktuellen Level Sets zu betrachten. Hierzu wird um jeden Voxel der aktiven Kontur eine Kugel mit dem Radius δ gelegt. Die Menge der Voxel, die innerhalb einer solchen Kugel liegen, wird als Nachbarschafts-Band oder Narrow Band bezeichnet.

Definition 3.1.0.18

Es sei $C_t = (x_1, x_2, \dots, x_k)$ mit $x_i \in \hat{\Omega} \ \forall \ i \in \{1, \dots k\}$ das Zero Level Set zum Zeitpunkt t. Also $\varphi(x_{i_1}, x_{i_2}, x_{i_3}, t) = 0 \ \forall \ i \in \{1, \dots k\}$. Dann definiere das *Narrow Band* von C_t mit Radius δ wie folgt:

$$NB(C_t) = \{ y \in \hat{\Omega} \mid \exists x \in C_n, \text{ mit } y \in B_{\delta}(x) \},$$
 (3.35)

wobei $B_{\delta}(x) = \{ z \in \hat{\Omega} \mid |x - z| \le \delta \}.$

Listing 3.2: Narrow Band Algorithmus

```
Initialize \varphi^0_{x_1,x_2,x_3} = \varphi_0, \ n = 0

for fixed number of iterations do

Compute NB(C_n)

for fixed number of iterations do

for each (x_1,x_2,x_3) \in C_n do

Compute F_A |\nabla \varphi| + F_G |\nabla \varphi| by the equations (3.33) and (3.34)

Multiply F_A |\nabla \varphi| + F_G |\nabla \varphi| by g(|\nabla \hat{u_0}|)

end

for each (x_1,x_2,x_3) \in NB(C_n) do

Find Point (y_1,y_2,y_3) \in C closest to (x_1,x_2,x_3)

Extend F_A |\nabla \varphi| + F_G |\nabla \varphi| and g(|\nabla \hat{u_0}|)

end

Compute \varphi^{n+1} by (3.32)

n = n+1

end

end
```

Wenn also bei jeder Iteration nur die Punkte innerhalb des Narrow Bands aktualisiert werden und dieses nach einer vom Benutzer festgelegten Anzahl von Iterationen reinitialisiert wird, ergibt sich der in Listing 3.2 dargestellte Algorithmus.

Durch die Eigenschaft des Narrow-Band Algorithmus, nur noch Voxel in einer vom Benutzer festgelegten Umgebung des Zero Level Sets zu betrachten und zu aktualisieren, verringert sich die Komplexitätsklasse wie in Abschnitt 5.1 gezeigt von $O(kn^5)$ auf $O(kn^5+(l+k)n^4)$. Der Hauptanteil der Berechnung liegt nun in der Bestimmung des Voxels, der sich auf der aktiven Kontur C befindet und minimalen Abstand zum betrachteten Voxel hat. In Abschnitt 3.3 werde ich den Standardalgorithmus sowie ein Approximationsverfahren vorstellen, um den Voxel mit minimalem Abstand zu bestimmen.

Der Nachteil des "edge stopping" Verfahrens ist offensichtlich. Damit Objekte segmentiert werden können, ist man von Regionen mit hohem Gradienten abhängig. Folglich können wir mit dem Verfahren nur Objekte mit starken Kanten finden sowie segmentieren. Bei Objekten mit schwachen Objektübergängen beziehungsweise Kanten läuft die aktive Kontur C, wie in Abbildung 3.3 zu sehen, bei zu vielen Iterationen über die Objektgrenzen hinaus.

Ein weiterer Nachteil ist, dass die stopping Funktion $g(|\nabla \hat{u_0}|)$ nie exakt Null ist und somit auch starke Kanten eines Objekts bei genügend vielen Iterationen über-

$3.1.\ DAS$ "EDGE STOPPING" VERFAHREN

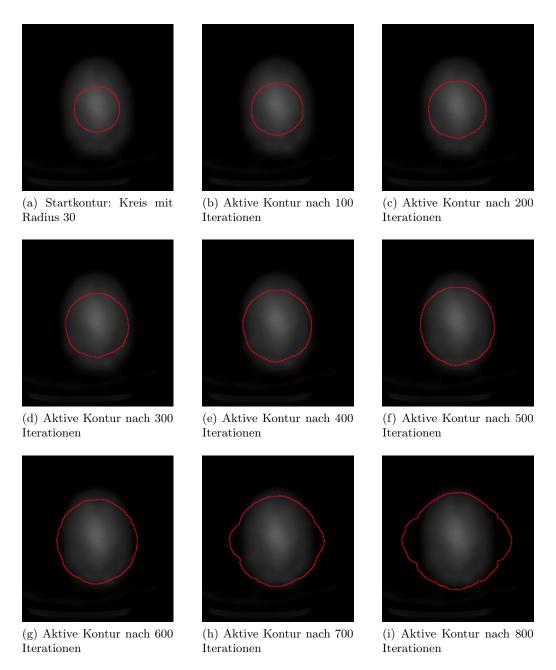


Abbildung 3.3: Evolution der aktiven Kontur bei einem Objekt mit kleinem Gradienten. Das Verfahren überspringt die Kanten des Objekts, weil die Objektübergänge zu schwach sind.

sprungen werden. Hier ist eine entsprechend gute Startkontur notwendig, wie in Abbildung 3.4, um das gewünschte Segment zu finden.

Eine Alternative zum "edge stopping" Verfahren bietet der "energy minimization" Algorithmus, der anstatt die Kanten des Objekts mit Hilfe einer *stopping* Funktion zu bestimmen, auf der Minimierung eines geeigneten Energiefunktionals basiert.

3.2 Der "energy minimization" Algorithmus

In diesem Abschnitt wird der sogenannte "energy minimization" Algotithmus vorgestellt und erläutert. Wie im vorigen Abschnitt angedeutet, beruht dieses Verfahren nicht auf einer stopping Funktion, sondern auf der Minimierung eines geeigneten Energiefunktionals. Dies hat den Vorteil, dass man bei der Segmentierung mit dem "energy minimization" Algorithmus nicht von Regionen mit einem hohen Gradienten abhängig ist. Im Folgenden wird das sogenannte Chan-Vese Energiefunktional von T. Chan und L. Vese aus [CV99] und [CV01] hergeleitet und der Zusammenhang zum klassischen Mumford-Shah Funktional der Segmentierung von D. Mumford und J. Shah aus [MS89] erläutert.

3.2.1 Herleitung des Chan-Vese Funktionals

Definition 3.2.1.1

Sei $\Omega \subset \mathbb{R}^3$ offen und beschränkt, $\partial \Omega$ der Rand von Ω und $u_0 : \Omega \to \mathbb{R}$ beschreibt die Volumendaten. Sei nun $\omega \subseteq \Omega$ offen und $\partial \omega = C$ die betrachtete aktive Kontur. Dann definiere mit

$$inside(C) := \omega,$$

 $outside(C) := \Omega \setminus \overline{\omega}$ (3.36)

das Innere sowie das Äußere der aktiven Kontur C.

Beschränkt man sich zunächst auf kontinuierliche Volumendaten u_0 , die aus genau zwei Bereichen mit jeweils konstanten Grauwerten bestehen und bezeichnet diese unterschiedlichen aber konstanten Grauwerte mit u_0^0 und u_0^1 . Das gesuchte Segment hat den Rand C_s und den Grauwert u_0^1 . Das heißt $\omega = inside(C_s)$ hat den Grauwert u_0^1 und $\Omega \setminus \overline{\omega} = outside(C_s)$ hat den Grauwert u_0^0 .

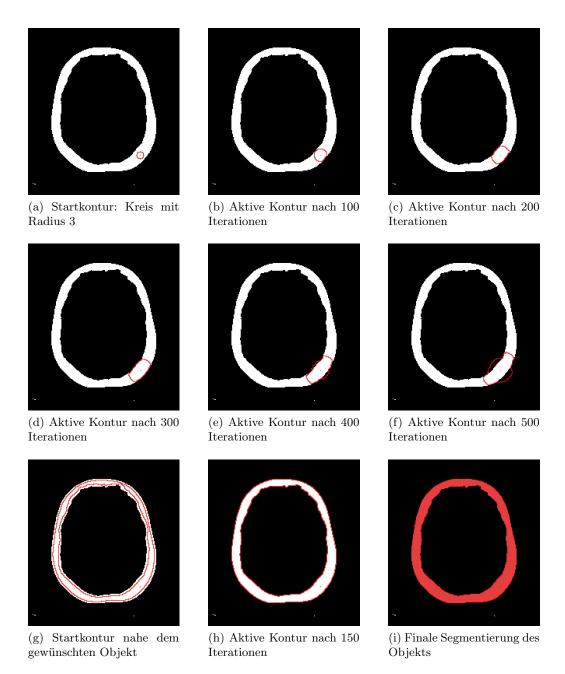


Abbildung 3.4: (a) - (f): Evolution der aktiven Kontur bei einem Objekt mit großem Gradienten, aber schlecht gewählter Startkontur. Das Verfahren überspringt die Kanten des Objekts, da die *stopping* Funktion an den Kanten nicht exakt Null wird

(g) - (i) Segmentierung des gewünschten Objekts bei hinreichend gut gewählter Startkontur.

Für den beschriebenen Spezialfall ergibt sich nun folgendes Energiefunktional:

$$F_1(C) + F_2(C) = \int_{inside(C)} |u_0(x_1, x_2, x_3) - c_1|^2 dx_1 dx_2 dx_3 + \int_{outside(C)} |u_0(x_1, x_2, x_3) - c_2|^2 dx_1 dx_2 dx_3,$$
(3.37)

wobei c_1, c_2 die mittleren Grauwerte von u_0 innerhalb beziehungsweise außerhalb der aktiven Kontur C beschreiben. Man sieht leicht ein, dass die aktive Kontur C_s (der Rand von ω) das gegebene Funktional minimiert:

$$\inf_{C} \{ F_1(C) + F_2(C) \} \approx 0 \approx F_1(C_s) + F_2(C_s). \tag{3.38}$$

Bemerkung 3.2.1.1

Es ist leicht einzusehen, dass der Rand von ω , also die Kontur C_s , das Funktional $F_1(C)+F_2(C)$ minimiert. Liegt die aktive Kontur zum Beispiel außerhalb des gesuchten Segmentes, so gilt $F_1(C)>0$ und $F_2(C)\approx 0$. Liegt die Kontur im Inneren des gesuchten Segments, so folgt $F_1(C)\approx 0$ aber $F_2(C)>0$. Wenn die aktive Kontur sowohl im Inneren als auch im Äußeren des gesuchten Segments liegt, so gilt $F_1(C)>0$ und $F_2(C)>0$. Schließlich gilt sowohl $F_1(C)\approx 0$ als auch $F_2(C)\approx 0$, wenn $C=C_s$ gilt, also wenn die aktive Kontur C genau die Kanten des gesuchten Segments beschreibt. Anschaulich sind alle vier Fälle in Abbildung 3.5 dargestellt.

In Analogie zum Mumford-Shah Energiefunktional der Segmentierung aus [MS89] von D. Mumford und J. Shah, werden nun noch Regularisierungsterme hinzugefügt, um zusätzlich die Oberfläche der aktiven Kontur C und/oder das Volumen inside(C) zu minimieren. Man erhält also wie T. Chan und L. Vese in [CV01] folgende Definition des Energiefunktionals:

Definition 3.2.1.2

Das Chan-Vese Energiefunktional der Segmentierung ist definiert als:

$$E_{\text{CV}}(C, c_1, c_2) = \mu \cdot \text{Area}(C) + \nu \cdot \text{Volume}(inside(C))$$

$$+ \lambda_1 \int_{inside(C)} |u_0(x_1, x_2, x_3) - c_1|^2 dx_1 dx_2 dx_3$$

$$+ \lambda_2 \int_{outside(C)} |u_0(x_1, x_2, x_3) - c_2|^2 dx_1 dx_2 dx_3.$$
(3.39)

Hierbei bezeichnen $\mu \geq 0, \nu \geq 0, \lambda_1 \geq 0, \lambda_2 \geq 0$ fest gewählte Parameter, mit denen der Benutzer den Einfluss der einzelnen Terme steuern kann.

$3.2.\ \ DER\ "ENERGY\ MINIMIZATION"\ ALGORITHMUS$

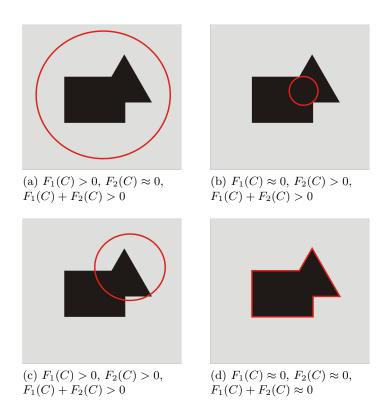


Abbildung 3.5: Die vier möglichen Positionen einer Kurve und die Auswirkungen auf die Energien F_1 und F_2 .

Für das *Chan-Vese* Energiefunktional betrachtet man nun, zur Segmentierung, das Minimierungsproblem:

$$\inf_{c_1, c_2, C} F_{\text{CV}}(C, c_1, c_2). \tag{3.40}$$

3.2.2 Beziehung zum Mumford-Shah Funktional

Im Vergleich zur Definition des *Chan-Vese* Energiefunktionals ist das klassische *Mumford-Shah* Funktional der Segmentierung, aus [MS89] von D. Mumford und J. Shah, wie folgt definiert:

Definition 3.2.2.1

Das Mumford-Shah Energiefunktional der Segmentierung ist definiert als:

$$\begin{split} E_{\text{MS}}(u,C) &= \mu \cdot \text{Area}(C) \\ &+ \lambda \int_{\Omega} |u_0(x_1, x_2, x_3) - u(x_1, x_2, x_3)|^2 dx_1 dx_2 dx_3 \\ &+ \int_{\Omega \setminus C} |\nabla u(x_1, x_2, x_3)|^2 dx_1 dx_2 dx_3 \,, \end{split}$$
(3.41)

wobei $u_0: \Omega \to \mathbb{R}^3$ die gegebenen Volumendaten darstellt, $\mu \geq 0$ und $\lambda \geq 0$ festgewählte Parameter sind und $u(x_1, x_2, x_3)$ die Grauwertfunktion beschreibt, die aus glatten Regionen R_i und starken Kanten C besteht.

Eine reduzierte oder vereinfachte Form des Minimierungsproblems

$$\inf_{(u,C)} F_{\text{MS}}(u,C) \tag{3.42}$$

ist die Beschränkung auf stückweise konstante Funktionen $u(x_1, x_2, x_3)$. So kann für u angenommen werden, dass $u = a_i$ auf jeder Zusammenhangskomponente R_i von $\Omega \setminus C$, mit $a_i \in \mathbb{R}$, konstant ist. In diesem Fall muss, wie von D. Mumford und J. Shah in [MS89] gezeigt, a_i der mittlere Grauwert der Zusammenhangskomponente R_i sein.

Bemerkung 3.2.2.1

Mit der Beschränkung auf stückweise konstante Funktionen wird das Minimierungsproblem

$$\inf_{(u,C)} F_{\text{MS}}(u,C) \tag{3.43}$$

auch minimal partition problem genannt.

3.2. DER "ENERGY MINIMIZATION" ALGORITHMUS

Wie man nun leicht einsieht, ist dass Chan-Vese Energiefunktional (3.39) mit den Parametern $\nu = 0$ und $\lambda_1 = \lambda_2 = \lambda$

$$E(C, c_1, c_2) = \mu \cdot \text{Area}(C)$$

$$+ \lambda \int_{inside(C)} |u_0(x_1, x_2, x_3) - c_1|^2 dx_1 dx_2 dx_3$$

$$+ \lambda \int_{outside(C)} |u_0(x_1, x_2, x_3) - c_2|^2 dx_1 dx_2 dx_3,$$
(3.44)

ein Spezialfall des minimal partition problem, wenn nach der besten Approximation $u(x_1, x_2, x_3)$ von $u_0(x_1, x_2, x_3)$ gesucht wird, wobei $u(x_1, x_2, x_3)$ genau zwei Werte annimt:

$$u = \begin{cases} average(u_0) \text{ inside } C\\ average(u_0) \text{ outside } C \end{cases}$$
(3.45)

und lediglich eine Kante C hat, die von der aktiven Kontur repräsentiert wird. Dann gilt offensichtlich

$$E_{\text{MS}}(u,C) = \mu \cdot \text{Area}(C) + \lambda \int_{\Omega} |u_0(x_1, x_2, x_3) - u(x_1, x_2, x_3)|^2 dx_1 dx_2 dx_3 + \int_{\Omega \setminus C} |\nabla u(x_1, x_2, x_3)|^2 dx_1 dx_2 dx_3 = \mu \cdot \text{Area}(C) + \lambda \int_{inside(C)} |u_0(x_1, x_2, x_3) - average(u_0) \text{ inside } C|^2 dx_1 dx_2 dx_3 + \lambda \int_{outside(C)} |u_0(x_1, x_2, x_3) - average(u_0) \text{ outside } C|^2 dx_1 dx_2 dx_3 + const$$
(3.46)

und somit die Äquivalenz der beiden Minimierungsprobleme.

3.2.3 Das *Chan-Vese* Funktional und Level Set Methoden

Die aktive Kontur C wird nun wieder als Zero Level Set einer höher dimensionalen Funktion $\varphi(x_1, x_2, x_3)$ betrachtet, wobei $\varphi(x_1, x_2, x_3)$ wieder die vorzeichenbehaftete Abstandsfunktion, wie in (2.3.0.11) definiert, ist. Es gilt also:

•
$$C = \partial \omega = \{(x_1, x_2, x_3) \in \Omega \mid \varphi(x_1, x_2, x_3) = 0\},\$$

- $inside(C) = \omega = \{(x_1, x_2, x_3) \in \Omega \mid \varphi(x_1, x_2, x_3) < 0\},\$
- $outside(C) = \Omega \setminus \overline{\omega} = \{(x_1, x_2, x_3) \in \Omega \mid \varphi(x_1, x_2, x_3) > 0\}.$

Um das Chan-Vese Energiefunktional mit Level Set Methoden zu formulieren, wird nun die aktive Kontur C durch die vorzeichenbehaftete Abstandsfunktion $\varphi(x_1, x_2, x_3)$ ersetzt und analog zu T. Chan, B. Merriman, S. Osher und H.-K. Zhao in [CMOZ96] vorgegangen.

Durch ersetzen der aktiven Kontur C in (3.39) erhält man zunächst:

$$E_{\text{CV}}(\varphi, c_1, c_2) = \mu \cdot \text{Area}(\varphi(x_1, x_2, x_3) = 0) + \nu \cdot \text{Volume}(inside(\varphi(x_1, x_2, x_3) = 0)) + \lambda_1 \int_{inside(\varphi(x_1, x_2, x_3) = 0)} |u_0(x_1, x_2, x_3) - c_1|^2 dx_1 dx_2 dx_3$$
(3.47)
+ \lambda_2 \int_{outside(\varphi(x_1, x_2, x_3) = 0)} |u_0(x_1, x_2, x_3) - c_2|^2 dx_1 dx_2 dx_3.

Als nächstes wird die Oberfläche Area $(\varphi=0)$ und das Volumen Volume $(inside(\varphi=0))$ der aktiven Kontur separat betrachtet. Um die Oberfläche und das Volumen der aktiven Kontur C zu beschreiben, wird die Heaviside Funktion H(x) und die Dirac-Delta Funktion $\delta(z)$ benutzt.

Definition 3.2.3.1

Die Funktion

$$H(z) = \begin{cases} 1, & \text{falls } z \ge 0, \\ 0, & \text{falls } z < 0 \end{cases}$$
 (3.48)

wird als Heaviside Funktion bezeichnet und die Differentiation der Heaviside Funktion nach der Variablen z

$$\delta(z) = \frac{\partial}{\partial z} H(z) = \begin{cases} \infty, & \text{falls } z = 0, \\ 0, & \text{falls } z < 0 \text{ oder } z > 0 \end{cases}, \text{ mit } \int_{-\infty}^{\infty} \delta(z) \, dz = 1 \quad (3.49)$$

wird Dirac-Delta Funktion genannt.

Bemerkung 3.2.3.1

Für die Funktion $H(\varphi(x_1, x_2, x_3))$ gilt:

$$H(\varphi(x_1, x_2, x_3)) = 1 \Leftrightarrow (x_1, x_2, x_3) \in outside(\varphi(x_1, x_2, x_3)) = 0$$
 (3.50)

sowie

$$H(\varphi(x_1, x_2, x_3)) = 0 \Leftrightarrow (x_1, x_2, x_3) \in inside(\varphi(x_1, x_2, x_3)) = 0.$$
 (3.51)

3.2. DER "ENERGY MINIMIZATION" ALGORITHMUS

Das Volumen der aktiven Kontur C kann also durch einfaches Integrieren der Funktion 1 - H(z) über dem Gebiet Ω bestimmt werden. Also gilt:

Volume(
$$\varphi(x_1, x_2, x_3) = 0$$
) = $\int_{\Omega} (1 - H(\varphi(x_1, x_2, x_3))) dx_1 dx_2 dx_3$. (3.52)

Offensichtlich wird die aktive Kontur C durch den Gradienten $\delta(z) = \frac{\partial}{\partial z} H(z)$ der Funktion H(z) beschrieben. Die Oberfläche der aktiven Kontur C kann also durch Integration der Funktion $\delta(z)$ bestimmt werden:

Area
$$(\varphi(x_1, x_2, x_3) = 0) = \int_{\Omega} |\nabla(H(\varphi(x_1, x_2, x_3)))| dx_1 dx_2 dx_3$$

$$= \int_{\Omega} h(\varphi(x_1, x_2, x_3)) |\nabla\varphi(x_1, x_2, x_3)| dx_1 dx_2 dx_3.$$
(3.53)

Ebenso ist es mit Hilfe der Heaviside Funktion H(z) möglich, die beiden Terme

$$\int_{inside(\varphi(x_1, x_2, x_3) = 0)} |u_0(x_1, x_2, x_3) - c_1|^2 dx_1 dx_2 dx_3$$
 (3.54)

und

$$\int_{outside(\varphi(x_1,x_2,x_3)=0)} |u_0(x_1,x_2,x_3) - c_2|^2 dx_1 dx_2 dx_3, \qquad (3.55)$$

über ganz Ω definieren. Es ergeben sich:

$$\int_{inside(\varphi(x_1, x_2, x_3) = 0)} |u_0(x_1, x_2, x_3) - c_1|^2 dx_1 dx_2 dx_3$$

$$= \int_{\Omega} |u_0(x_1, x_2, x_3) - c_1|^2 \cdot (1 - H(\varphi(x_1, x_2, x_3))) dx_1 dx_2 dx_3$$
(3.56)

und

$$\int_{outside(\varphi(x_1, x_2, x_3) = 0)} |u_0(x_1, x_2, x_3) - c_2|^2 dx_1 dx_2 dx_3$$

$$= \int_{\Omega} |u_0(x_1, x_2, x_3) - c_1|^2 \cdot H(\varphi(x_1, x_2, x_3)) dx_1 dx_2 dx_3.$$
(3.57)

Somit kann nun das Chan-Vese Energiefunktional über ganz Ω definiert werden:

$$E_{\text{CV}}(\varphi, c_1, c_2) = \mu \int_{\Omega} \delta(\varphi(x_1, x_2, x_3)) |\nabla \varphi(x_1, x_2, x_3)| dx_1 dx_2 dx_3$$

$$+ \nu \int_{\Omega} (1 - H(\varphi(x_1, x_2, x_3))) dx_1 dx_2 dx_3$$

$$+ \lambda_1 \int_{\Omega} |u_0(x_1, x_2, x_3) - c_1|^2 \cdot (1 - H(\varphi(x_1, x_2, x_3))) dx_1 dx_2 dx_3$$

$$+ \lambda_2 \int_{\Omega} |u_0(x_1, x_2, x_3) - c_1|^2 \cdot H(\varphi(x_1, x_2, x_3)) dx_1 dx_2 dx_3.$$
(3.58)

Bemerkung 3.2.3.2

Mit Hilfe der Heaviside Funktion kann ebenfalls die Lösung u des Mumford-Shah minimal partition problem in (3.41) als

$$u(x_1, x_2, x_3) = c_1(1 - H(\varphi(x_1, x_2, x_3))) + c_2H(\varphi(x_1, x_2, x_3))$$
(3.59)

geschrieben werden. Wählt man die Funktion $\varphi(x_1, x_2, x_3)$ nun fest und minimiert das *Chan-Vese* Energiefunktional (3.58), bezüglich der beiden Konstanten c_1 und c_2 , so gilt zunächst:

$$\inf_{c_1} E_{CV}(\varphi, c_1, c_2) = \inf_{c_1} \int_{\Omega} |u_0(x_1, x_2, x_3) - c_1|^2 \cdot (1 - H(\varphi(x_1, x_2, x_3))) \, dx_1 dx_2 dx_3$$
(3.60)

und somit

$$\frac{\partial}{\partial c_1} \int_{\Omega} |u_0(x_1, x_2, x_3) - c_1|^2 \cdot (1 - H(\varphi(x_1, x_2, x_3))) \, dx_1 dx_2 dx_3$$

$$= c_1 \int_{\Omega} 1 - H(\varphi(x_1, x_2, x_3)) \, dx_1 dx_2 dx_3$$

$$- \int_{\Omega} u_0(x_1, x_2, x_3) \, (1 - H(\varphi(x_1, x_2, x_3))) \, dx_1 dx_2 dx_3$$
(3.61)

. .

Also:

$$c_1(\varphi(x_1, x_2, x_3)) = \frac{\int_{\Omega} u_0(x_1, x_2, x_3) \cdot (1 - H(\varphi(x_1, x_2, x_3))) dx_1 dx_2 dx_3}{\int_{\Omega} (1 - H(\varphi(x_1, x_2, x_3))) dx_1 dx_2 dx_3}, \quad (3.62)$$

falls $\int_{\Omega} (1 - H(\varphi(x_1, x_2, x_3))) dx_1 dx_2 dx_3 \neq 0$. Dies gilt, wenn zum Beispiel das Volumen der aktiven Kontur $Volume(\varphi(x_1, x_2, x_3) = 0) > 0$ ist. Analog ergibt sich für c_2 :

$$c_2(\varphi(x_1, x_2, x_3)) = \frac{\int_{\Omega} u_0(x_1, x_2, x_3) \cdot H(\varphi_0(x_1, x_2, x_3)) dx_1 dx_2 dx_3}{\int_{\Omega} H(\varphi(x_1, x_2, x_3)) dx dy dz},$$
(3.63)

falls $\int_{\Omega} H(\varphi(x_1, x_2, x_3)) dx dy dz \neq 0$. Offensichtlich sind c_1 und c_2 , also die mittleren Grauwerte von ω und $\Omega \setminus \overline{\omega}$.

Bemerkung 3.2.3.3

Mit den bisher hergeleiteten Formeln, kann das Chan-Vese Energiefunktional auch als Funktional der charakteristischen Funktion $H(\varphi(x_1, x_2, x_3))$ des Gebietes ω

3.2. DER "ENERGY MINIMIZATION" ALGORITHMUS

geschrieben werden. Das neue Energiefunktional \mathcal{E} hat dann folgende Form:

$$\mathcal{E}(\chi_{\omega}) = \mu \int_{\Omega} |\nabla \chi_{\omega}(x_{1}, x_{2}, x_{3})| dx_{1} dx_{2} dx_{3}$$

$$+ \nu \int_{\Omega} \chi_{\omega}(x_{1}, x_{2}, x_{3}) dx_{1} dx_{2} dx_{3}$$

$$+ \lambda_{1} \int_{\Omega} \left(|u_{0}(x_{1}, x_{2}, x_{3}) - c_{1}(\chi_{\omega}(x_{1}, x_{2}, x_{3}))|^{2} \right)$$

$$(1 - \chi_{\omega}(x_{1}, x_{2}, x_{3})) dx_{1} dx_{2} dx_{3}$$

$$+ \lambda_{2} \int_{\Omega} |u_{0}(x_{1}, x_{2}, x_{3}) - c_{2}(\chi_{\omega}(x_{1}, x_{2}, x_{3}))|^{2} \chi_{\omega}(x_{1}, x_{2}, x_{3}) dx_{1} dx_{2} dx_{3},$$

$$(3.64)$$

wobei $\chi_{\omega} = H(\varphi)$ die charakteristische Funktion von ω ist. Das neue Minimierungsproblem lautet dann:

$$\inf_{\chi_{\omega}} \mathcal{E}(\chi_{\omega}), \quad \chi_{\omega}(x_1, x_2, x_3) \in \{0, 1\}.$$
(3.65)

In dieser Arbeit wird jedoch die Level Set Funktion φ für die Darstellung der aktiven Kontur C benutzt, weil es sowohl geometrische als auch numerische Vorteile mit sich bringt. Das Problem könnte jedoch auch dadurch gelöst werden, in dem \mathcal{E} über alle charakteristischen Funktionen minimiert wird.

Um nun das Minimum des Chan-Vese Energiefunktionals (3.58) zu bestimmen, nutzt man die Ergebnisse der Variationsrechnung. Hierzu ist es jedoch notwendig, zunächst einmal die Heaviside Funktion H(z) zu glätten. Sei also $H_{\epsilon} \in C^2(\Omega)$ eine Approximation der Heaviside Funktion H(z) aus (3.48) und $H'_{\epsilon} = \delta_{\epsilon}$ eine Approximation der Dirac-Delta Funktion (3.49), so dass

$$\lim_{\epsilon \to 0} H_{\epsilon}(z) = H(z) \tag{3.66}$$

und

$$\lim_{\epsilon \to 0} H'_{\epsilon}(z) = \delta(z) \tag{3.67}$$

gilt. Beispiele für Approximationen der Heaviside Funktion werden in (3.2.4) angegeben. Das resultierende geglättete *Chan-Vese* Energiefunktional ist wie folgt

definiert:

$$\begin{split} E_{\text{CV}}^{\epsilon}(\varphi,c_{1},c_{2}) &= \mu \int_{\Omega} \delta_{\epsilon}(\varphi(x_{1},x_{2},x_{3})) |\nabla \varphi(x_{1},x_{2},x_{3})| \, dx_{1} dx_{2} dx_{3} \\ &+ \nu \int_{\Omega} (1 - H_{\epsilon}(\varphi(x_{1},x_{2},x_{3}))) \, dx_{1} dx_{2} dx_{3} \\ &+ \lambda_{1} \int_{\Omega} |u_{0}(x_{1},x_{2},x_{3}) - c_{1}|^{2} (1 - H_{\epsilon}(\varphi(x_{1},x_{2},x_{3}))) \, dx_{1} dx_{2} dx_{3} \\ &+ \lambda_{2} \int_{\Omega} |u_{0}(x_{1},x_{2},x_{3}) - c_{1}|^{2} H_{\epsilon}(\varphi(x_{1},x_{2},x_{3})) \, dx_{1} dx_{2} dx_{3} \, . \end{split}$$

$$(3.68)$$

Wie oben bereits erwähnt, ist man nun an dem Minimum des *Chan-Vese* Energiefunktionals interessiert. Daraus resultierend sucht man also:

$$\inf_{\varphi,c_1,c_2} E_{\text{CV}}^{\epsilon}(\varphi,c_1,c_2). \tag{3.69}$$

Hierzu wählt man die mittleren Grauwerte c_1 und c_2 fest und minimiert das Chan-Vese Energiefunktional $E_{\text{CV}}^{\epsilon}(\varphi, c_1, c_2)$ bezüglich der vorzeichenbehafteten Abstandsfunktion $\varphi(x_1, x_2, x_3)$. Für das Minimum des Energiefunktionals muss, wie in 2.4 beschrieben, die sogenannte Euler-Lagrange Gleichung (2.41) gelten.

Hier sei nun E^{ϵ} wie folgt definiert:

$$E^{\epsilon}(\varphi) = \mu \cdot \delta_{\epsilon}(\varphi(x_{1}, x_{2}, x_{3})) |\nabla \varphi(x_{1}, x_{2}, x_{3})| + \nu \cdot (1 - H_{\epsilon}(\varphi(x_{1}, x_{2}, x_{3}))) + \lambda_{1} \cdot |u_{0}(x_{1}, x_{2}, x_{3}) - c_{1}|^{2} \cdot (1 - H_{\epsilon}(\varphi(x_{1}, x_{2}, x_{3}))) + \lambda_{2} \cdot |u_{0}(x_{1}, x_{2}, x_{3}) - c_{1}|^{2} \cdot H_{\epsilon}(\varphi(x_{1}, x_{2}, x_{3})) .$$

$$(3.70)$$

Um die Euler-Lagrange Gleichung zu erfüllen, werden die folgenden Ableitungen benötigt:

$$\frac{\partial E^{\epsilon}}{\partial \varphi}, \qquad \sum_{j=1}^{3} \frac{\partial}{\partial x_{j}} \frac{\partial E^{\epsilon}}{\partial (\partial_{j} \varphi)}, \tag{3.71}$$

die im Folgenden kurz hergeleitet werden.⁴ Man erhält mit den Regeln der Differentiation:

$$\frac{\partial E^{\epsilon}}{\partial \varphi} = \mu \delta_{\epsilon}'(\varphi) |\nabla \varphi| - \nu \delta_{\epsilon}(\varphi) - \lambda_1 |u_0 - c_1|^2 \delta_{\epsilon}(\varphi) + \lambda_2 |u_0 - c_2|^2 \delta_{\epsilon}(\varphi), \quad (3.72)$$

sowie

$$\frac{\partial E^{\epsilon}}{\partial(\partial_{j}\varphi)} = \mu \delta_{\epsilon}(\varphi) \frac{\partial \varphi}{\partial x_{j}} \frac{1}{|\nabla \varphi|}$$

$$\Rightarrow \frac{\partial}{\partial x_{j}} \frac{\partial E^{\epsilon}}{\partial(\partial_{j}\varphi)} = \mu \frac{\partial \varphi}{\partial x_{j}} \frac{1}{|\nabla \varphi|} \left(\frac{\partial}{\partial x_{j}} \delta_{\epsilon}(\varphi) \right) + \mu \delta_{\epsilon}(\varphi) \left(\frac{\partial}{\partial x_{j}} \left(\frac{\partial \varphi}{\partial x_{j}} \frac{1}{|\nabla \varphi|} \right) \right) \tag{3.73}$$

 $^{^4}$ Zur besseren Übersicht werden die Koeffizienten x_1, x_2 und x_3 bei der Herleitung weggelassen.

für $j \in \{1, 2, 3\}$. Also folgt eingesetzt in die Euler-Lagrange Gleichung (2.41):

$$0 = \sum_{j=1}^{3} \frac{\partial}{\partial x_{j}} \frac{\partial E^{\epsilon}}{\partial (\partial_{j}\varphi)} - \frac{\partial E^{\epsilon}}{\partial \varphi}$$

$$\Leftrightarrow 0 = \sum_{j=1}^{3} \left[\mu \frac{\partial \varphi}{\partial x_{j}} \frac{1}{|\nabla \varphi|} \left(\frac{\partial}{\partial x_{j}} \delta_{\epsilon}(\varphi) \right) + \mu \delta_{\epsilon}(\varphi) \left(\frac{\partial}{\partial x_{j}} \left(\frac{\partial \varphi}{\partial x_{j}} \frac{1}{|\nabla \varphi|} \right) \right) \right]$$

$$- \mu \delta'_{\epsilon}(\varphi) |\nabla \varphi| + \nu \delta_{\epsilon}(\varphi) + \lambda_{1} |u_{0} - c_{1}|^{2} \delta_{\epsilon}(\varphi) - \lambda_{2} |u_{0} - c_{2}|^{2} \delta_{\epsilon}(\varphi)$$

$$\Leftrightarrow 0 = \mu \sum_{j=1}^{3} \left[\delta'_{\epsilon}(\varphi) \left(\frac{\partial \varphi}{\partial x_{j}} \right)^{2} \frac{1}{|\nabla \varphi|} + \delta_{\epsilon}(\varphi) \frac{\partial}{\partial x_{j}} \left(\frac{\partial \varphi}{\partial x_{j}} \right) \right]$$

$$- \mu \delta'_{\epsilon}(\varphi) |\nabla \varphi| + \nu \delta_{\epsilon}(\varphi) + \lambda_{1} |u_{0} - c_{1}|^{2} \delta_{\epsilon}(\varphi) - \lambda_{2} |u_{0} - c_{2}|^{2} \delta_{\epsilon}(\varphi)$$

$$\Leftrightarrow 0 = \mu \delta'_{\epsilon}(\varphi) \frac{\sum_{j=1}^{3} \left(\frac{\partial \varphi}{\partial x_{j}} \right)^{2}}{|\nabla \varphi|} + \mu \delta_{\epsilon}(\varphi) \sum_{j=1}^{3} \left[\frac{\partial}{\partial x_{j}} \left(\frac{\partial \varphi}{\partial x_{j}} \right) \right]$$

$$- \mu \delta'_{\epsilon}(\varphi) |\nabla \varphi| + \nu \delta_{\epsilon}(\varphi) + \lambda_{1} |u_{0} - c_{1}|^{2} \delta_{\epsilon}(\varphi) - \lambda_{2} |u_{0} - c_{2}|^{2} \delta_{\epsilon}(\varphi)$$

$$\Leftrightarrow 0 = \mu \delta'_{\epsilon}(\varphi) \frac{|\nabla \varphi|^{2}}{|\nabla \varphi|} + \mu \delta_{\epsilon}(\varphi) \cdot \operatorname{div}\left(\frac{\nabla \varphi}{|\nabla \varphi|} \right)$$

$$- \mu \delta'_{\epsilon}(\varphi) |\nabla \varphi| + \nu \delta_{\epsilon}(\varphi) + \lambda_{1} |u_{0} - c_{1}|^{2} \delta_{\epsilon}(\varphi) - \lambda_{2} |u_{0} - c_{2}|^{2} \delta_{\epsilon}(\varphi)$$

$$\Leftrightarrow 0 = \delta_{\epsilon}(\varphi) \left[\mu \cdot \operatorname{div}\left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) + \nu + \lambda_{1} |u_{0} - c_{1}|^{2} - \lambda_{2} |u_{0} - c_{2}|^{2} \right]$$

als notwendige Bedingung für das Minimum des Minimierungsproblems

$$\inf_{\omega} E_{\text{CV}}^{\epsilon}(\varphi, c_1, c_2). \tag{3.75}$$

Ein iteratives Verfahren für die Berechnung des Minimums wird jetzt gewonnen, indem wie schon beim "edge stopping" Verfahren eine neue Zeitvariable t eingeführt wird und die notwendige Bedingung der Euler-Lagrange Gleichung (3.74) mit der Ableitung von $\varphi(x_1, x_2, x_3, t)$ nach der neuen Variablen t gleichgesetzt wird. Man erhält also, wie T.Chan und L. Vese in [CV01], die folgende partielle Differentialgleichung, die die Evolution der aktiven Kontur C beschreibt:

$$\frac{\partial \varphi}{\partial t} = \delta_{\epsilon}(\varphi) \left[\mu \cdot \operatorname{div} \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) + \nu + \lambda_1 |u_0 - c_1|^2 - \lambda_2 |u_0 - c_2|^2 \right] = 0.$$
 (3.76)

Durch eine Diskretisierung sowie Linearisierung der partiellen Differentialgleichung mit Hilfe der Finiten-Vorwärtsdifferenzen und des expliziten Euler-Verfahrens

$$\frac{\partial}{\partial t}\varphi(x_1, x_2, x_3, t) = \frac{\varphi^{n+1} - \varphi^n}{\Delta t}$$
(3.77)

erhält man schließlich wie T. Chan und L. Vese in [CV01]

$$\frac{\varphi^{n+1} - \varphi^n}{\Delta t} = \delta_{\epsilon}(\varphi) \left[\mu \cdot \operatorname{div} \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) + \nu + \lambda_1 |u_0 - c_1|^2 - \lambda_2 |u_0 - c_2|^2 \right]
\Leftrightarrow \varphi^{n+1} = \varphi^n + \Delta t \cdot \delta_{\epsilon}(\varphi) \left[\mu \cdot k + \nu + \lambda_1 |u_0 - c_1|^2 - \lambda_2 |u_0 - c_2|^2 \right] ,$$
(3.78)

da div $\left(\frac{\nabla \varphi}{|\nabla \varphi|}\right)$ gerade die mittlere Krümmung k liefert. Diese kann man mit den beim "edge stopping" Verfahren vorgestellten Methoden der zentralen Finiten Differenzenquotien durch

$$k = \frac{\begin{pmatrix} \varphi_{x_1x_1}(\varphi_{x_2}^2 + \varphi_{x_3}^2) + \varphi_{x_2x_2}(\varphi_{x_1}^2 + \varphi_{x_3}^2) + \varphi_{x_3x_3}(\varphi_{x_1}^2 + \varphi_{x_2}^2) \\ -2(\varphi_{x_1}\varphi_{x_2}\varphi_{x_1x_2} + \varphi_{x_1}\varphi_{x_3}\varphi_{x_1x_3} + \varphi_{x_2}\varphi_{x_3}\varphi_{x_2x_3}) \end{pmatrix}}{(\varphi_{x_1}^2 + \varphi_{x_2}^2 + \varphi_{x_3}^2)^{3/2}}$$
(3.79)

berechnen.

Man erhält diesbezüglich folgenden Algorithmus zur Bestimmung der Veränderung der aktiven Kontur C mit der Anfangsbedingung $\varphi(x_1, x_2, x_3, 0) = \varphi_0$:

Listing 3.3: Energy Minimization Algorithm

```
Initialize \varphi_{x_1,x_2,x_3}^0 = \varphi_0, n = 0
for fixed number of iterations do
Compute c_1(\varphi^n) and c_2(\varphi^n) by (3.62) and (3.63)
Compute Curvature k by (2.36)
Compute \varphi^{n+1} by (3.78)
n = n+1
end
```

3.2.4 Approximationen der Heaviside Funktion

Weiter benötigt werden die Approximationen der Heaviside Funktion und der Dirac-Delta Funktion. T. Chan, B. Merriman, S.Osher und H.-K. Zhao schlagen in [CMOZ96] zunächst folgende Approximation der Heaviside Funktion vor:

Definition 3.2.4.1

Die Funktion

$$H_{1,\epsilon}(z) = \begin{cases} 1, & \text{falls } z > \epsilon, \\ 0, & \text{falls } z < -\epsilon, \\ \frac{1}{2} + \frac{z}{2\epsilon} + \frac{1}{2\pi} \sin\left(\frac{\pi z}{\epsilon}\right), & \text{falls } |z| \le \epsilon \end{cases}$$
 (3.80)

3.2. DER "ENERGY MINIMIZATION" ALGORITHMUS

approximiert die Heaviside Funktion. Offensichtlich gilt:

$$\delta_{1,\epsilon} = \frac{\partial}{\partial z} H_{1,\epsilon}(z) = \begin{cases} 0, & \text{falls } |z| > \epsilon, \\ \frac{1}{2\epsilon} \left(1 + \cos\left(\frac{\pi z}{\epsilon}\right) \right), & \text{falls } |z| \le \epsilon \end{cases} \in C^{1}(\Omega)$$
 (3.81)

approximiert die Dirac-Delta Funktion und:

$$\lim_{\epsilon \to 0} H_{1,\epsilon}(z) = H(z) \tag{3.82}$$

sowie

$$\lim_{\epsilon \to 0} \frac{\partial}{\partial z} H_{1,\epsilon}(z) = \delta(z). \tag{3.83}$$

Wie in Abbildung 3.6 und in Gleichung (3.81) zu sehen, ist die Approximation der Dirac-Delta Funktion nur auf dem Intervall $[-\epsilon, \epsilon]$ ungleich Null. Somit hat die Evolutionsgleichung (3.78) nur Einfluss auf einen kleinen Bereich um das Zero Level Set herum. Demzufolge ist die Segmentierung bei der gegebenen Approximation der Heaviside Funktion stark von der Initialisierung der aktiven Kontur abhängig, weil das Verfahren mit der Approximation der Heaviside dazu neigt gegen ein lokales Minimum zu konvergieren, wie in Abbildung A.8 (e) - (h) zu sehen. Aus diesem Grund haben T. Chan und L. Vese in [CV01] folgende Approximation der Heaviside Funktion vorgeschlagen:

Definition 3.2.4.2

Die Funktion

$$H_{2,\epsilon}(z) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{z}{\epsilon}\right) \in C^{\infty}(\Omega)$$
 (3.84)

approximiert die Heaviside Funktion. Es gilt:

$$\delta_{2,\epsilon} = \frac{\partial}{\partial z} H_{2,\epsilon}(z) = \frac{\epsilon}{\pi} \frac{1}{\epsilon^2 + z^2} \quad \in C^{\infty}(\Omega)$$
 (3.85)

approximiert die Dirac-Delta Funktion und:

$$\lim_{\epsilon \to 0} H_{2,\epsilon}(z) = H(z) \tag{3.86}$$

sowie

$$\lim_{\epsilon \to 0} \frac{\partial}{\partial z} H_{2,\epsilon}(z) = \delta(z). \tag{3.87}$$

Eine weitere mögliche Approximation, die in dieser Arbeit benutzt wird, ist die Folgende:

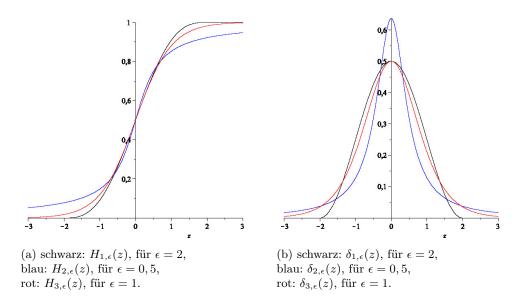


Abbildung 3.6: Verschiedene Approximationen der Heaviside Funktion (links) und der zugehörigen Approximationen der Dirac-Delta Funktion (rechts).

Definition 3.2.4.3

Die Funktion

$$H_{3,\epsilon}(z) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{z}{\epsilon}\right) \in C^{\infty}(\Omega)$$
 (3.88)

approximiert die Heaviside Funktion. Es ist leicht einzusehen, dass

$$\delta_{3,\epsilon} = \frac{\partial}{\partial z} H_{3,\epsilon}(z) = \frac{1}{2\epsilon} \frac{1}{2\epsilon} \tanh^2 \left(\frac{z}{\epsilon}\right) \in C^{\infty}(\Omega)$$
 (3.89)

ebenfalls die Dirac-Delta Funktion approximiert und

$$\lim_{\epsilon \to 0} H_{3,\epsilon}(z) = H(z) \tag{3.90}$$

sowie

$$\lim_{\epsilon \to 0} \frac{\partial}{\partial z} H_{3,\epsilon}(z) = \delta(z) \tag{3.91}$$

gilt.

Mit dieser Approximation der Heaviside Funktion neigt das Verfahren ebenfalls dazu, gegen ein lokales Minimum zu konvergieren. Jedoch hat diese Approximation der Dirac-Delta Funktion einen größeren Einfluss auf den Bereich um das Zero Level Set als die in (3.80) definierte Approximation.

3.3. WEITERENTWICKLUNG DER MODELLE

Bemerkung 3.2.4.1

Beispiele für das unterschiedliche Verhalten des Verfahrens bei verschiedenen Approximationen der Heaviside Funktion werden in Abbildung A.8 gegeben.

3.3 Weiterentwicklung der Modelle

In diesem Abschnitt werden die im Laufe der Diplomarbeit entstandenen Erweiterungen und Optimierungen der Modelle vorgestellt und beschrieben.

3.3.1 Weiterentwicklung des "edge stopping" Verfahrens

Wie bereits in Abschnitt 3.1 beschrieben ist ein Nachteil des "edge stopping" Verfahrens, dass die stopping Funktion $g(|\nabla \hat{u_o}|)$ nie exakt Null ist und somit auch starke Kanten eines Objekts bei genügend vielen Iterationen übersprungen werden. Hier wurde zunächst die folgende stopping Funktion definiert:

$$g_3(|\nabla u_0|) = \frac{1}{1 + |\nabla(G_{\sigma}(x) \star vis(u_0(x)))|^p + (\hat{u_0} - grey)^2}, \text{ mit } p \ge 1, \quad (3.92)$$

wobei $vis(u_0(x))$, wie schon zuvor, die Transformation der Volumendaten auf den visuellen Bereich beschreibt und grey ein vom Benutzer gewählter Grauwert ist. Offensichtlich erfüllt die stopping Funktion $g_3(|\nabla \hat{u_0}|)$ die geforderten Eigenschaften. Es gilt:

- auf dem Intervall $[0,\infty)$ ist die Funktion $g_3(|\nabla \hat{u_0}|)$ positiv und regulär,
- es gilt $g_3'(|\nabla \hat{u_0}|) \leq 0$ sowie
- $\lim_{|\nabla \hat{u_0}| \to \infty} g_3(|\nabla \hat{u_0}|) = 0.$

Mit den Eigenschaften dieser stopping Funktion kann der Benutzer einen Grauwert festlegen, so dass die stopping Funktion in Regionen in denen der gewählte Grauwert stark von dem Grauwert des Bildes $\hat{u_0}$ abweicht, schneller gegen Null fällt. Weicht der vom Benutzer gewählte Grauwert jedoch auch stark von dem mittleren Grauwert des gesuchten Objekts ab, breitet sich die aktive Kontur in diesen Regionen nur sehr langsam oder gar nicht aus. Deshalb ist es sinnvoll den Grauwert grey nicht vom Benutzer wählen zu lassen, sondern bei jeder Iteration des Verfahrens

Listing 3.4: Berechnung der diskreten vorzeichenbehafteten Distanzfunktion

```
for each (x_1, x_2, x_3) \in \Omega do

if (x_1, x_2, x_3) \in \Omega \setminus \overline{\omega} set \varphi(x_1, x_2, x_3) = \infty

if (x_1, x_2, x_3) \in \overline{\omega} set \varphi(x_1, x_2, x_3) = -\infty

end

for each (x_1, x_2, x_3) \in \Omega do

for each (y_1, y_2, y_3) \in C do

Compute dist = |(x_1, x_2, x_3) - (y_1, y_2, y_3)|

if dist < \varphi(x_1, x_2, x_3) set \varphi(x_1, x_2, x_3) = \pm \text{dist}

end

end
```

den mittleren Grauwert innerhalb der Kontur zu berechnen und die Abweichung von diesem in die stopping Funktion einfließen zu lassen. Es ergibt sich also

$$g_4(|\nabla u_0|, u_0) = \frac{1}{1 + |\nabla(G_{\sigma}(x) \star vis(u_0(x)))|^p + (\hat{u_0} - grey^*)^2},$$
mit $p \ge 1$ und $grey^* = \frac{\int_{\omega} u_0(x_1, x_2, x_3) dx_1 dx_2 dx_3}{\int_{\omega} 1 dx_1 dx_2 dx_3}.$

$$(3.93)$$

Bemerkung 3.3.1.1

Tests haben gezeigt, dass das Verwenden dieser stopping Funktion bei selbst erstellten Phantomdaten gut funktioniert, jedoch in realen medizinischen Daten sehr fehleranfällig ist.

Eine andere Idee das Überspringen von starken Kanten zu verhindern, ist die vorzeichenbehaftete Distanzfunktion $\varphi(x_1, x_2, x_3)$ während der Iteration zu reinitialisieren. Bei einer geeignet gewählten Schrittweite Δt , können starke Kanten durch die stetige Reinitialisierung der vorzeichenbehafteten Distanzfunktion nicht mehr übersprungen werden.

Der erste naheliegende Algorithmus zur Reinitialisierung der vorzeichenbehafteten Distanzfunktion ist in Listing 3.4 dargestellt.

Bemerkung 3.3.1.2 (Laufzeitanalyse)

Der Einfachheit halber wird zur Analyse der Laufzeit eine Größe der Volumendaten von $n \times n \times n$ angenommen. Die Anzahl der Voxel auf der aktiven Kontur C sei dann $O(n^2)$. Der erste Schritt zur Bestimmung der diskreten vorzeichenbehafteten Abstandsfunktion betrachtet jeden Voxel genau einmal und setzt die Distanz

Listing 3.5: Berechnung einer approximierten vorzeichenbehafteten Distanzfunkti-

```
init empty List todo; init empty field marked
for each x \in C do
     \mathbf{set} \ \varphi(x) = 0
     add x to todo
     add x to marked
end
while todo \neq \emptyset do
     for each x \in \text{todo } \mathbf{do}
          set \varphi(y) = \varphi(x) + 1 for all y \in N_8(x) \& y \notin \text{marked}
          if y \notin \text{marked}
               add y to todo
               add y to marked
          end
          remove x from todo
     end
end
```

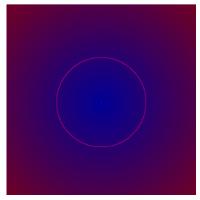
auf $\pm \infty$. Also benötigt der Schritt $O(n^3)$ Zeiteinheiten. Im zweiten Schritt wird nun jeder Voxel $(x_1, x_2, x_3) \in \Omega$ mit jedem Voxel $(y_1, y_2, y_3) \in C$ verglichen und deren Abstand berechnet. Hierzu werden also $O(n^5)$ Zeiteinheiten gebraucht. Die Reinitialisierung der diskreten vorzeichenbehafteten Distanzfunktion $\varphi(x_1, x_2, x_3)$ hat also eine Gesamtlaufzeit von

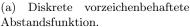
$$O(n^3) + O(n^5) = O(n^5)$$
. (3.94)

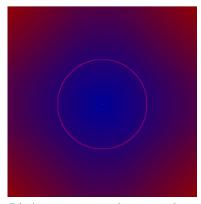
Aufgrund der hohen Gesamtlaufzeit der Reinitialisierung bietet es sich an, die vorzeichenbehaftete Distanzfunktion nicht exakt zu bestimmen, sondern sie lediglich zu approximieren. Die Berechnung der Approximation der vorzeichenbehafteten Abstandsfunktion im dreidimensionalen Fall, bezüglich einer 8er-Nachbarschaft N_8 , funktioniert wie in Listing 3.5 dargestellt.

Bemerkung 3.3.1.3 (Laufzeitanalyse)

Auch hier wird wieder eine Größe der Volumendaten von $n \times n \times n$ sowie die Größe der aktiven Kontur C als $O(n^2)$ angenommen. Im ersten Schritt wird jeder Voxel der aktiven Kontur C, also jeder Voxel $(x_1, x_2, x_3) \in C$, genau einmal betrachtet. Der erste Schritt benötigt also $O(n^2)$ Zeiteinheiten. Sei nun k die Anzahl der Nachbarn bezüglich der gewählten Nachbarschaft. Also k = const. Danach wird im zweiten Schritt für jeden Voxel $(x_1, x_2, x_3) \in \Omega$ jeder der k-Nachbarn genau







(b) Approximation der vorzeichenbehafteten Abstandsfunktion.

Abbildung 3.7: Darstellung zweier verschiedener Verfahren zur Berechnung der vorzeichenbehafteten Distanzfunktion. Der Farbverlauf von Blau nach Rot kennzeichnet die Pixel mit kleinem oder negativen Abstand zur aktiven Kontur hin zu den Pixeln mit einem großen Abstand zur aktiven Kontur.

einmal betrachtet. Für den zweiten Schritt werden also $O(kn^3)$ Zeiteinheiten gebraucht. Die Gesamtlaufzeit des Algorithmus zur Berechnung der approximierten vorzeichenbehafteten Distanzfunktion beträgt also:

$$O(n^2) + O(kn^3) = O(kn^3) = O(n^3)$$
, da k fest. (3.95)

Der Rechenaufwand hat sich also von $O(n^5)$ auf $O(n^3)$ verbessert. In Abbildung 3.7 sind die unterschiedlichen Distanzfunktionen für einen Kreis zu sehen. Insbesondere kann in einer kleinen Umgebung der aktiven Kontur C kaum einen Unterschied festgestellt werden.

Ein weiteres Problem der Segmentierung mit dem "edge stopping" Verfahren ist der hohe Rechenaufwand bei der Bestimmung des Punktes $(x_1, x_2, x_3) \in C$ mit minimalem Abstand zum Punkt $(y_1, y_2, y_3) \in \Omega$. Dieser Punkt wird bei der Berechnung der erweiterten Ausbreitungsgeschwindigkeit (3.1.0.12) benötigt. Der Standardalgorithmus zur Bestimmung des Voxels auf der aktiven Kontur C mit minimalem Abstand ist in Listing 3.6 beschrieben.

Damit die erweiterte Ausbreitungsgeschwindigkeit berechnet werden kann, muss jeder Voxel $(y_1, y_2, y_3) \in \Omega$ mit jedem Voxel $(x_1, x_2, x_3) \in C$ verglichen werden. Auch hier genügt es, wie bei der Approximation der vorzeichenbehafteten Abstandsfunktion, den Voxel nur approximativ zu bestimmen.

Es bietet sich also eine Art Zwiebelschalen Algorithmus an, um iterativ alle Voxel mit minimalem Abstand zu bestimmen. Man beginnt mit allen Voxeln auf der aktiven Kontur, also allen Voxeln $(x_1,x_2,x_3) \in C$. Nun werden zu diesen Voxeln alle Nachbarn (y_1,y_2,y_3) bezüglich der 26er-Nachbarschaft betrachtet, genauer gesagt, alle Voxel $(y_1,y_2,y_3) \in N_{26}(x_1,x_2,x_3)$. Für diese Voxel wird nun $closest_Point(y_1,y_2,y_3) = (x_1,x_2,x_3)$ gesetzt, also ist der Punkt $(x_1,x_2,x_3) \in C$ der Voxel mit minimalem Abstand zum Voxel $(y_1,y_2,y_3) \in N_{26}(x_1,x_2,x_3)$. Nachdem alle Voxel $(x_1,x_2,x_3) \in C$ betrachtet wurden, wird der Schritt iterativ für deren Nachbarn fort geführt und so für jeden Voxel $(y_1,y_2,y_3) \in \Omega$ eine Approximation des Voxels $(x_1,x_2,x_3) \in C$ mit minimalem Abstand zu (y_1,y_2,y_3) berechnet. In Listing 3.7 ist dieser Algorithmus nochmals dargestellt.

Bemerkung 3.3.1.4

Bei dem Standardalgorithmus zu Berechnung des Voxels $(x_1, x_2, x_3) \in C$ mit minimalem Abstand zum Voxel $(y_1, y_2, y_3) \in \Omega$ werden $O(n^2)$ Zeiteinheiten benötigt, wenn eine Größe $O(n^2)$ der aktiven Kontur C angenommen wird. Da dieser Algorithmus für alle Voxel $(y_1, y_2, y_3) \in \Omega$ durchgeführt werden muss, ergibt sich bei einer Größe der Volumendaten von $n \times n \times n$ eine Gesamtlaufzeit von $O(n^5)$ zur Bestimmung der Voxel mit minimalem Abstand. Somit braucht auch die Berechnung der erweiterten Ausbreitungsgeschwindigkeit für alle Voxel $(y_1, y_2, y_3) \in \Omega$ $O(n^5)$ Zeiteinheiten.

Mit dem sogenannten Zwiebelschalen Algorithmus wird jeder Voxel $(y_1, y_2, y_3) \in \Omega$ genau 26-mal betrachtet. Es ergibt sich also eine Gesamtlaufzeit von $O(n^3)$ zur Berechnung der in Definition 3.1.0.12 vorgestellten erweiterten Ausbreitungsgeschwindigkeit.

Abbildung 3.8 zeigt jeweils die erweiterte Ausbreitungsgeschwindigkeit bei gegebener aktiver Kontur C. Es ist leicht einzusehen, dass sich in einer kleinen Umgebung der aktiven Kontur C, trotz der Verbesserung des Rechenaufwands von $O(n^5)$ auf $O(n^3)$ durch Approximation, kaum ein Unterschied feststellen lässt.

Um die Idee des Zwiebelschalen Algorithmus auf auch auf den Narrow Band Algorithmus (siehe Listing 3.2) anzuwenden, kann der in Listing 3.9 beschriebene Algorithmus genutzt werden, wobei NB_size die Größe des Narrow Bands beschreibt.

Mit der gleichen Idee kann auch das Narrow Band der aktiven Kontur C bestimmt werden. Beim Standardalgorithmus wird im ersten Schritt die Distanz jedes Voxels zur aktiven Kontur C berechnet und im zweiten Schritt fällt die Entscheidung, ob ein Voxel zum Narrow Band gehört. Dies geschieht über eine einfache Abfrage, ob die Distanz kleiner oder gleich der Größe des Narrow Bands ist. Die Laufzeit ergibt sich also aus $O(n^5)$ für den ersten Schritt und $O(n^3)$ für den zweiten Schritt.

Listing 3.6: Berechnung des Voxel mit minimalem Abstand

```
init dist(y_1, y_2, y_3) = \infty, closest\_Point(y_1, y_2, y_3) = \emptyset

for each (x_1, x_2, x_3) \in C do

Compute distance between |(x_1, x_2, x_3) - (y_1, y_2, y_3)|

if |(x_1, x_2, x_3) - (y_1, y_2, y_3)| < dist(y_1, y_2, y_3)

set dist(y_1, y_2, y_3) = |(x_1, x_2, x_3) - (y_1, y_2, y_3)|

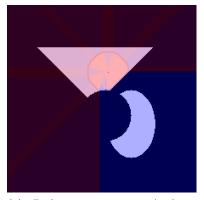
set closest\_Point(y_1, y_2, y_3) = (x_1, x_2, x_3)

end if
```

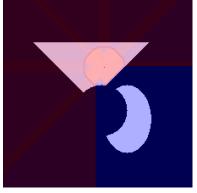
Listing 3.7: Approximierte Berechnung der Voxel mit minimalem Abstand

```
init closest \ Point(y) = \emptyset \ \forall (y) \in \Omega
init empty List todo; init empty field marked
for each (x) \in C do
    set closest Point(x) = (x)
    add x to todo
    add x to marked
end
while todo \neq \emptyset do
    for each (x) \in \text{todo } \mathbf{do}
         set closest\_Point(y) = x for all y \in N_{26}(x) \& y \notin marked
         if y \notin \text{marked}
              add y to todo
              add y to marked
         end
         remove x from todo
    end
end
```

3.3. WEITERENTWICKLUNG DER MODELLE



(a) Diskrete erweiterte Ausbreitungsgeschwindigkeit.



(b) Approximation der erweiterten Ausbreitungsgeschwindigkeit.

Abbildung 3.8: Darstellung zweier verschiedener Verfahren zur Berechnung der erweiterten Ausbreitungsgeschwindigkeit. Der Farbverlauf von Blau nach Rot kennzeichnet die Pixel mit kleiner Ausbreitungsgeschwindigkeit hin zu den Pixeln mit einem großer Ausbreitungsgeschwindigkeit.

Der Algorithmus zur Approximation aller Voxel des Narrow Bands ist in Listing 3.9 angegeben. Die Laufzeit des Algorithmus liegt dann nur noch bei $O(n^2)$. Im Algorithmus bezeichnet NB_{size} die Größe des Narrow Bands.

3.3.2 Weiterentwicklung des "energy minimization" Algorithmus

Betrachtet man noch einmal das Chan-Vese Energiefunktional aus (3.58):

$$E_{\text{CV}}(\varphi, c_1, c_2) = \mu \int_{\Omega} \delta(\varphi(x_1, x_2, x_3)) |\nabla \varphi(x_1, x_2, x_3)| \, dx_1 dx_2 dx_3$$

$$+ \nu \int_{\Omega} (1 - H(\varphi(x_1, x_2, x_3))) \, dx_1 dx_2 dx_3$$

$$+ \lambda_1 \int_{\Omega} |u_0(x_1, x_2, x_3) - c_1|^2 (1 - H(\varphi(x_1, x_2, x_3))) \, dx_1 dx_2 dx_3$$

$$+ \lambda_2 \int_{\Omega} |u_0(x_1, x_2, x_3) - c_1|^2 H(\varphi(x_1, x_2, x_3)) \, dx_1 dx_2 dx_3.$$
(3.96)

In dem *Chan-Vese* Energiefunktional kann der Term Area(C) umwandelt und verallgemeinert werden in: Area(C)^p mit $p \ge 1$. Sei N > 1 und $\Omega \subset \mathbb{R}^N$. Dann wählt man entweder $p := 1 \,\forall N$ und erhält obiges Modell oder man kann $p = \frac{N}{N-1}$ wählen.

Listing 3.8: Approximierte Berechnung des Voxel mit minimalem Abstand innerhalb des Narrow Bands

```
init closest\_Point(y) = \emptyset \ \forall (y) \in \Omega
init empty List todo; init empty field marked
for each (x) \in C do
    set closest Point(x) = (x)
    add x to todo, add x to marked
end
while i < NB size
    for each (x) \in \text{todo do}
        set closest\ Point(y) = x for all y \in N_{26}(x) \& y \notin marked
         if y \notin \text{marked}
             add y to todo, add y to marked
        end
        remove x from todo
        i = i + 1
    end
end
```

Listing 3.9: Approximierte Berechnung aller Voxel des Narrow Bands

```
init NB = \emptyset
init empty List todo; init empty field marked

for each (x) \in C do
  add x to NB, add x to marked

end

while i < NB\_size

for each (x) \in todo do
  add y to NB for all y \in N_{26}(x) & y \notin todo

if y \notin todo, add y to marked
  add y to todo, add y to marked
  end
  remove x from todo
  i = i + 1

end

end
```

3.3. WEITERENTWICKLUNG DER MODELLE

Für letzteren Fall wird die Aussage von L.C. Evans und R.F. Gariepy aus [EG92] ausgenutzt die besagt, dass

$$Volume(inside(C)) \le c \cdot Area(C)^{\frac{N}{N-1}}$$
(3.97)

mit einer nur von N abhängigen Konstanten c ist. Es kann also davon ausgegangen werden, dass dadurch Area(C) und Volume(inside(C)) in gewissem Sinne "vergleichbar" werden.

Definition 3.3.2.1

Das verallgemeinerte *Chan-Vese* Energiefunktional der Segmentierung ist also wie folgt definiert:

$$E_{\text{CV*}}(\varphi, c_1, c_2) = \mu \left(\int_{\Omega} \delta(\varphi(x_1, x_2, x_3)) |\nabla \varphi(x_1, x_2, x_3)| \, dx_1 dx_2 dx_3 \right)^{\frac{N}{N-1}}$$

$$+ \nu \int_{\Omega} (1 - H(\varphi(x_1, x_2, x_3))) \, dx_1 dx_2 dx_3$$

$$+ \lambda_1 \int_{\Omega} |u_0(x_1, x_2, x_3) - c_1|^2 (1 - H(\varphi(x_1, x_2, x_3))) \, dx_1 dx_2 dx_3$$

$$+ \lambda_2 \int_{\Omega} |u_0(x_1, x_2, x_3) - c_1|^2 H(\varphi(x_1, x_2, x_3)) \, dx_1 dx_2 dx_3 .$$

$$(3.98)$$

Werden die Heaviside Funktion und die Dirac-Delta Funktion wieder durch eine der im vorigen Abschnitt vorgestellten Approximationen ersetzt, erhält man folgendes Minimierungsproblem:

$$\inf_{\varphi, c_1, c_2} E_{\text{CV}}^{\epsilon}(\varphi, c_1, c_2). \tag{3.99}$$

Hält man wieder die mittleren Grauwerte c_1 und c_2 fest und minimiert $E^{\epsilon}_{\text{CV}*}(\varphi, c_1, c_2)$ bezüglich der vorzeichenbehafteten Abstandsfunktion $\varphi(x_1, x_2, x_3)$, so muss für das Minimum des Energiefunktionals die Euler-Lagrange Gleichung (2.41) gelten. Es folgt wie in [Tak] von R. Takei

$$0 = \sum_{j=1}^{3} \frac{\partial}{\partial x_{j}} \frac{\partial E^{\epsilon}}{\partial (\partial_{j} \varphi)} - \frac{\partial E^{\epsilon}}{\partial \varphi}$$

$$\Leftrightarrow 0 = \delta_{\epsilon}(\varphi) \left[\mu \frac{N}{N-1} \left(\int_{\Omega} \delta_{\epsilon}(\varphi) |\nabla \varphi| \right)^{\frac{1}{N-1}} \cdot \operatorname{div} \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) + \nu + \lambda_{1} |u_{0} - c_{1}|^{2} - \lambda_{2} |u_{0} - c_{2}|^{2} \right]$$
(3.100)

als notwendige Bedingung für das Minimum des Minimierungsproblems

$$\inf_{\varphi} E_{\text{CV}^*}^{\epsilon}(\varphi, c_1, c_2). \tag{3.101}$$

KAPITEL 3. MODELLE UND METHODEN DER SEGMENTIERUNG

Analog zum vorherigen Abschnitt wird nun die neue Zeitvariable t eingeführt und die Bedingung der Euler-Lagrange Gleichung (3.100) mit der Ableitung von φ nach der Variable t gleichgesetzt. Es folgt folgende partielle Differentialgleichung, die die Evolution der aktiven Kontur C bezüglich des neuen Energiefunktionals beschreibt:

$$\frac{\partial \varphi}{\partial t} = \delta_{\epsilon}(\varphi) \left[\mu \frac{N}{N-1} \left(\int_{\Omega} \delta_{\epsilon}(\varphi) |\nabla \varphi| \right)^{\frac{1}{N-1}} \cdot \operatorname{div} \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) + \nu + \lambda_{1} |u_{0} - c_{1}|^{2} - \lambda_{2} |u_{0} - c_{2}|^{2} \right].$$
(3.102)

Da div $\left(\frac{\nabla \varphi}{|\nabla \varphi|}\right)$ wieder die mittlere Krümmung k und $\left(\int_{\Omega} \delta_{\epsilon}(\varphi)|\nabla \varphi|\right)^{\frac{1}{N-1}}$ gerade die Oberfläche der aktiven Kontur C, also Area $(C)^{\frac{1}{N-1}}$, liefert, erhält man schließlich:

$$\varphi^{n+1} = \varphi^{n} + \Delta t \cdot \delta_{\epsilon}(\varphi) \left[\mu k \frac{N}{N-1} \text{Area}(C)^{\frac{1}{N-1}} + \nu + \lambda_{1} |u_{0} - c_{1}|^{2} - \lambda_{2} |u_{0} - c_{2}|^{2} \right].$$
(3.103)

Betrachtet man noch einmal die Verfahrensvorschrift der Evolution des "energy minimization" Algorithmus, also entweder

$$\varphi^{n+1} = \varphi^n + \Delta t \cdot \delta_{\epsilon}(\varphi) \left[\mu \cdot k + \nu + \lambda_1 |u_0 - c_1|^2 - \lambda_2 |u_0 - c_2|^2 \right] , \qquad (3.104)$$

oder

$$\varphi^{n+1} = \varphi^n + \Delta t \cdot \delta_{\epsilon}(\varphi) \left[\mu k \frac{N}{N-1} \operatorname{Area}(C)^{\frac{1}{N-1}} + \nu + \lambda_1 |u_0 - c_1|^2 - \lambda_2 |u_0 - c_2|^2 \right]$$
(3.105)

so fällt auf, dass die Terme $\lambda_1|u_0-c_1|^2$ und $\lambda_2|u_0-c_2|^2$ im Allgemeinen einen deutlich größeren Wertebereich erreichen als die Terme $\mu \cdot k + \nu$ oder $\mu k \frac{N}{N-1} \operatorname{Area}(C)^{\frac{1}{N-1}} + \nu$. Um dieses zu Problem zu lösen und damit die Wahl der Parameter $\lambda_1, \lambda_2, \mu$ und ν zu vereinfachen, wurde folgende Änderung am *Chan-Vese* Energiefunktional vor-

genommen:

$$E_{\text{CV}_{\text{log}}}(\varphi, c_1, c_2) = \mu \int_{\Omega} \delta(\varphi(x_1, x_2, x_3)) |\nabla \varphi(x_1, x_2, x_3)| \, dx_1 dx_2 dx_3$$

$$+ \nu \int_{\Omega} (1 - H(\varphi(x_1, x_2, x_3))) \, dx_1 dx_2 dx_3$$

$$+ \lambda_1 \int_{\Omega} \left(\log(|u_0(x_1, x_2, x_3) - c_1|^2) \right)$$

$$(1 - H(\varphi(x_1, x_2, x_3))) \, dx_1 dx_2 dx_3$$

$$+ \lambda_2 \int_{\Omega} \left(\log(|u_0(x_1, x_2, x_3) - c_1|^2) \right)$$

$$H(\varphi(x_1, x_2, x_3)) \, dx_1 dx_2 dx_3.$$
(3.106)

Bezüglich diesem Energiefunktional erhält man die partielle Differentialgleichung

$$\frac{\partial \varphi}{\partial t} = \delta_{\epsilon}(\varphi) \left[\mu \cdot k + \nu + \lambda_1 \log(|u_0 - c_1|^2) - \lambda_2 \log(|u_0 - c_2|^2) \right], \qquad (3.107)$$

die die Evolution der aktiven Kontur C nach der Zeit t beschreibt.

In allen vorgestellten partiellen Differentialgleichung kommt die mittlere Krümmung k vor. Diese berechnet sich im dreidimensionalen Fall aus

$$k = \frac{\begin{pmatrix} \varphi_{x_1x_1}(\varphi_{x_2}^2 + \varphi_{x_3}^2) + \varphi_{x_2x_2}(\varphi_{x_1}^2 + \varphi_{x_3}^2) + \varphi_{x_3x_3}(\varphi_{x_1}^2 + \varphi_{x_2}^2) \\ -2(\varphi_{x_1}\varphi_{x_2}\varphi_{x_1x_2} + \varphi_{x_1}\varphi_{x_3}\varphi_{x_1x_3} + \varphi_{x_2}\varphi_{x_3}\varphi_{x_2x_3}) \end{pmatrix}}{(\varphi_{x_1}^2 + \varphi_{x_2}^2 + \varphi_{x_3}^2)^{3/2}}$$
(3.108)

mit Hilfe der zentralen Finiten Differenzenquotienten.

Jedoch gilt diese geometrische Eigenschaft nur für den Fall, dass $\varphi(x_1, x_2, x_3)$ die vorzeichenbehaftete Abstandsfunktion ist. Da die Funktion $\varphi(x_1, x_2, x_3)$ während der Iteration des "energy minimization" Algorithmus jedoch schnell von der vorzeichenbehafteten Abstandsfunktion abweicht, ist es sinnvoll auch hier wieder eine Reinitialisierung der vorzeichenbehafteten Abstandsfunktion vorzunehmen. Für die Reinitialisierung der Funktion $\varphi(x_1, x_2, x_3)$ zur vorzeichenbehafteten Distanzfunktion kann wieder der Standardalgorithmus (siehe Listing 3.4) oder der Approximationsalgorithmus (siehe Listing 3.5) zur Berechnung einer Approximation der vorzeichenbehafteten Distanzfunktion benutzt werden.

Betrachtet man einmal ein Schnittbild realer medizinischer Volumendaten, so fällt auf, dass außerhalb des gescannten Objekts ein sehr großer Bereich einen Grauwert nahe Null besitzt. Dieser kann bei der Segmentierung vernachlässigt werden.

Je nach verwendeter Approximation der Heaviside Funktion kann dieser Bereich bei der Segmentierung mit dem "energy minimization" Algorithmus jedoch großen Einfluss auf das Ergebnis der Segmentierung haben. Um dieses Problem zu lösen, kann zum einen die Güte der approximierten Heaviside Funktion verbessert werden, in dem ϵ sehr klein gewählt wird oder zum anderen der zu betrachtende wichtige Bereich, um das gesuchte Objekt, vor der Segmentierung vom Benutzer festgelegt werden. Hierzu definiert der Benutzer, also vor Beginn der Segmentierung, ein Gebiet $\Omega^* \subset \Omega \subset \mathbb{R}^3$ und die Energiefunktionale werden dann auf das Gebiet Ω^* eingeschränkt. Für das *Chan-Vese* Energiefunktional aus (3.58) gilt dann:

$$E_{\text{CV}}(\varphi, c_1, c_2) = \mu \int_{\Omega^*} \delta(\varphi(x_1, x_2, x_3)) |\nabla \varphi(x_1, x_2, x_3)| dx_1 dx_2 dx_3$$

$$+ \nu \int_{\Omega^*} (1 - H(\varphi(x_1, x_2, x_3))) dx_1 dx_2 dx_3$$

$$+ \lambda_1 \int_{\Omega^*} |u_0(x_1, x_2, x_3) - c_1|^2 (1 - H(\varphi(x_1, x_2, x_3))) dx_1 dx_2 dx_3$$

$$+ \lambda_2 \int_{\Omega^*} |u_0(x_1, x_2, x_3) - c_1|^2 H(\varphi(x_1, x_2, x_3)) dx_1 dx_2 dx_3.$$
(3.109)

Analog können auch die Anderen in dieser Arbeit vorgestellten Energiefunktionale auf das Gebiet Ω^* einschränkt werden. Somit erreicht man, dass der Fehler der Segmentierung verringert wird, da nur noch der wesentliche Bereich um das gesuchte Objekt betrachtet wird.

Bemerkung 3.3.2.1

Durch die Wahl eines Gebiets Ω^* vom Benutzer kann der zu betrachtende Bildbereich stark von der Größe des Originalbildbereichs abweichen. Hier ergibt sich zusätzlich zu den angesprochenen Vorteilen auch eine erhebliche Verringerung des Rechenaufwandes.

Eine weitere Idee die während dieser Arbeit entstand, um zum einen den Bereich der Segmentierung auf den wichtigen Bereich um das gewünschte Objekt zu beschränken und zum anderen den Rechenaufwand zu verringern war, die Idee des Narrow Bands vom "edge stopping" Verfahren auf den "energy minimization" Algorithmus zu übertragen. Hierzu wird zu Beginn jeder Iteration ein Narrow Band mit einer vorher gewählten Größe NB_{size} um die aktive Kontur C berechnet. Nun wird das Energiefunktional bezüglich dem so entstandenen Gebiet $\Omega' \subset \Omega$ minimiert und anschließend wieder das aktuelle Narrow Band berechnet. Es ergibt sich also der folgende Algorithmus mit Anfangsbedingung $\varphi(x_1, x_2, x_3, 0) = \varphi_0$:

Listing 3.10: Energy Minimization Algorithm with Narrow Band

```
Initialize \varphi_{x_1,x_2,x_3}^0 = \varphi_0, n = 0
for fixed number of iterations do
Compute Narrow Band NB(C_n) with size NB_{size}
Compute c_1(\varphi^n) and c_2(\varphi^n) by (3.62) and (3.63)
Compute Curvature k by (2.36)
Compute \varphi^{n+1} by (3.78)
n = n + 1
```

Bemerkung 3.3.2.2

Mit diesem Algorithmus ist eine gute bis sehr gute Startkontur, die nahe an dem gewünschten Objekt liegt, notwendig. Trotzdem tendiert das Verfahren auch bei guter Startkontur gegen ein falsches Minimum zu konvergieren. Da die Laufzeit insbesondere bei sehr großen Datensätzen deutlich geringer ist als das Standardverfahren, erscheint hier eine Weiterentwicklung des Verfahrens notwendig und sinnvoll.

Kapitel 4

Implementierung

In diesem Kapitel wird auf die Implementierung der in den vorangegangenen Abschnitten vorgestellten Verfahren eingegangen. Die Darstellung der Ergebnisse erfolgte mit dem Programm YaDiV (Yet another DICOM-Image Viewer), welches im Fachbereich für Graphische Datenverarbeitung der Leibniz Universität Hannover von Dipl.-Math. Karl-Ingo Friese entwickelt wurde. Das Programm liest DICOM-Datensätze¹ ein und stellt die Schnittdaten, wie in Abbildung 4.1 zu sehen, sowohl zweidimensional als auch dreidimensional dar.

Die verwendete Programmiersprache während der Implementierung war Java in der Sprachversion 1.6.0_10 und die benutzte Programmierumgebung war Eclipse SDK in der Version 3.4.0 unter Windows Vista. Das entwickelte Programm funktioniert problemlos unter Windows XP, Windows Vista und Linux², sowie unter 32 und 64 bit. Andere Betriebssysteme wie MacOS X oder weitere Linux Distributionen wurden nicht getestet, wobei kein Grund zur Annahme besteht, dass es unter diesen Betriebssystemen nicht lauffähig sein könnte.

Zur Visualisierung der 3D-Daten wurde Java3D in der Version 1.5.2 verwendet. Da die beschriebenen Segmentierungsverfahren, sowohl das "edge stopping" Verfahren als auch der "energy minimization" Algorithmus, sehr zeitkritisch sind, wurden die Verfahren als eigenständiger Thread implementiert. Somit ist es möglich, nach dem Start eines Segmentierungsverfahrens auch weiterhin die Volumendaten zu betrachten, ohne das die Programmoberfläche blockiert wird.

¹DICOM steht für "Digital Imaging and Communications in Medicine" und beschreibt einen offenen Standard zum Austausch von medizinischen Daten. Ein DICOM-Datensatz enthält neben den Schnittbildern auch zahlreiche Informationen über den Patienten wie zum Beispiel den behandelnden Arzt, Geräteparameter oder das Aufnahmedatum.

²Das Programm wurde unter openSuse 11.031 und Fedora 8 getestet.

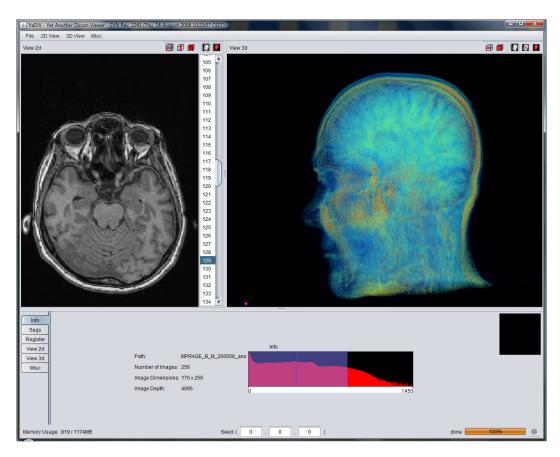


Abbildung 4.1: Die Hauptansicht des Programms YaDiV. Im linken Bereich sind die zweidimensionalen Schnittdaten und im rechten Bereich die dreidimensionale Darstellung der Schnittdaten zu sehen.

Die Kommunikation mit dem Graphical User Interface (GUI) des YaDiV Programms sowie das Ausführen der Algorithmen findet über die Klassen SegGenSnake und SegGenSnakeEnergy statt.

4.1 Die Klasse Distancefunction

Die Klasse Distancefunction repräsentiert die vorzeichenbehaftete Abstandsunktion $\varphi(x_1, x_2, x_3)$ im Dreidimensionalen oder $\varphi(x_1, x_2)$ im Zweidimensionalen und stellt Methoden zur Berechnung der im jeweiligen Algorithmus benötigten Werte zur Verfügung.

4.1.1 Die Attribute der Klasse Distancefunction

phi repräsentiert die diskretisierte vorzeichenbehaftete Abstandsfunktion.

- speed speichert die Werte der Ausbreitungsgeschwindigkeit F, der Punkte $x \in \Omega$.

_x_size, _y_size, _z_size geben die Dimensionen des Volumendatensatzes an.

_x_spacing, _y_spacing, _z_spacing speichern die Pixel spacing Werte.

_3d_distance gibt an, ob es sich um eine dreidimensionale oder zweidimensionale vorzeichenbehaftete Abstandsfunktion handelt.

starting points speichert die Startvoxel der Segmentierung.

4.1.2 Die Methoden der Klasse Distancefunction

Die Klasse Distancefunction besitzt zwei verschiedene Konstruktoren. Der Standardkonstruktor initialisiert die gegebenen Attribute und ruft die Methode generate_distancees auf, um die vorzeichenbehaftete Abstandsfunktion für eine Kugel (3d) oder einen Kreis (2d) zu berechnen. Der zweite Konstruktor setzt die Attribute der Klasse und initialisiert die vorzeichenbehaftete Abstandsfunktion für eine gegebene Segmentierung wie folgt:

$$\varphi(x_1, x_2, x_3) = \begin{cases}
0, & \text{falls } (x_1, x_2, x_3) \in C, \\
-1, & \text{falls } (x_1, x_2, x_3) \in \omega, \\
1, & \text{falls } (x_1, x_2, x_3) \in \Omega \setminus \overline{\omega}.
\end{cases} (4.1)$$

4.2. DIE KLASSEN SEGGENSNAKE UND SEGGENSNAKEENERGY

Damit eine Segmentierung wiederholt oder neugestartet werden kann, besitzt die Klasse zwei Methoden um die Attribute sowie die vorzeichenbehaftete Abstandsfunktion zu reinitialisieren. Hier wird wieder unterscheiden, ob die Segmentierung mit einer Kugel beziehungsweise einem Kreis startet oder mit einer vorgegebenen Segmentierung.

Zur Reinitialisierung der vorzeichenbehafteten Abstandsfunktion während der Segmentierung, stehen die Methoden reinitialize _distances2d und reinitialize _-distances3d zur Verfügung. Hier hat der Benutzer die Wahl zwischen der Berechnung der diskreten vorzeichenbehafteten Abstandsfunktion oder der Berechnung einer Approximation der vorzeichenbehafteten Abstandsfunktion. Weitere Methoden der Klasse Distancefunction sind:

- is_front3D bestimmt, ob ein Punkt $(x_1, x_2, x_3) \in \Omega$ zum Zero Level Set gehört, also ob $(x_1, x_2, x_3) \in C$ gemäß Definition (3.1.0.16).
- is_front2D wie is_front3d, jedoch für den zweidimensionalen Fall bezüglich einer 8er-Nachbarschaft.
- **compute_curvature3d** berechnet die Krümmung $k = \operatorname{div}\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right)$ gemäß (3.79) mit Hilfe der zentralen Finiten Differenzenquotienten.
- **compute_curvature2d** berechnet analog zu compute_curvature3d die Krümmung, jedoch für den zweidimensionalen Fall.
- compute_speed3d berechnet die Ausbreitungsgeschwindigkeit eines Voxels zu gegebener stopping Funktion nach den Gleichungen (3.33) und (3.34).
- **compute_speed2d** berechnet, wie compute_speed3d, die Ausbreitungsgeschwindigkeit für den zweidimensionalen Fall.
- dirac berechnet den Wert der approximierten Dirac-Delta Funktion zur gegebenen Approximation der Heaviside Funktion.

4.2 Die Klassen SegGenSnake und SegGenSnakeEnergy

Über die Klassen SegGenSnake und SegGenSnakeEnergy findet zum einen die Kommunikation mit dem Graphical User Interface statt und zum anderen werden die in der Arbeit vorgestellten Algorithmen ausgeführt.

Die Klasse SegGenSnake bietet die Möglichkeit die Algorithmen 3.1 und 3.2 zu starten und über die Klasse SegGenSnakeEnergy kann der Algorithmus 3.3 ausgeführt werden. Hierzu stehen dem Benutzer zahlreiche Möglichkeiten zur Verfügung,

Einfluss auf das Verhalten des Algorithmus zu nehmen sowie die einzelnen in Abschnitt 3.3 vorgestellten Erweiterungen einzuschalten. Die Ausführung der Algorithmen findet, wie schon erwähnt, als eigenständiger Thread statt. Hierzu wurden jeweils die privaten Klassen

- CreateSnakeThread2d
- CreateSnakeThread3d

innerhalb der Klasse SegGenSnake und

- CreateEnergySnake2dThread
- CreateEnergySnake3dThread

innerhalb der Klasse SegGenSnakeEnergy implementiert. Eine detaillierte Aufzählung der wählbaren Parameter würde an dieser Stelle zu weit führen, daher sei hier auf den Quellcode, das Programm oder die mit Javadoc erstellte Dokumentation auf der beiliegenden CD verwiesen.

Weiterhin wurden die Klassen SnakeDistanceRenderer2d und SnakeSpeedRenderer2d implementiert, um das Verhalten der Algorithmen zu visualisieren und besser nachvollziehen zu können. Die Abbildungen 3.7 und 3.8 zeigen Beispiele der Visualisierung.

Um die Segmentierungsverfahren auf einen vom Benutzer festgelegten Bereich zu begrenzen, wurde die Klasse SnakeBoundingBox implementiert. Die Klasse bietet die Möglichkeit, eine Bounding Box auszuwählen und dessen Größe einzustellen. Die Visualisierung der Bounding Box geschieht über die beiden Klassen Snake BBoxRenderer2d und Snake BBoxRenderer3d.

$4.2.\ \ DIE\ KLASSEN\ SEGGENSNAKE\ UND\ SEGGENSNAKEENERGY$

Kapitel 5

Ergebnisse

In dem folgenden Kapitel wird eine Laufzeitanalyse der vorgestellten Algorithmen beschrieben und einige beispielhafte Ergebnisse präsentiert. Die erzielten Ergebnisse werden zum einen mit denen einer herkömmlichen Segmentierung wie zum Beispiel einer Segmentierung durch den Region-Grow Algorithmus verglichen und zum anderen werden die verschiedenen Verfahren untereinander verglichen. Dabei wird sowohl das "edge stopping" Verfahren als auch der "energy minimization" Algorithmus an selbst erstellten Phantomdaten und an realen Volumendaten aus der medizinischen Praxis vorgeführt und getestet.

5.1 Laufzeitanalyse der Verfahren

5.1.1 Das "edge stopping" Verfahren

Der Einfachheit halber wird zunächst wieder angenommen, dass die Größe der Volumendaten $n \times n \times n$ ist. Um die Funktion $\varphi(x_1, x_2, x_3)$ als vorzeichenbehaftete Distanzfunktion einer Kugel mit gegebenen Radius zu initialisieren, ist eine Laufzeit $O(n^3)$ nötig. Die Anzahl der Voxel auf der Kontur C sei $O(n^2)$. Als nächstes wird die Ausbreitungsgeschwindigkeit $g(u_0)*(F_A|\nabla\varphi|+F_G|\nabla\varphi|)$ für die $O(n^2)$ Voxel der Kontur C bestimmt. Zunächst wird also die stopping Funktion berechnet. Hierzu sind $O(n^3)$ Schritte notwendig. Die Ausbreitungsgeschwindigkeit für alle Voxel der aktiven Kontur C zu berechnen, benötigt danach noch $O(n^2)$ Schritte, da jeder Voxel genau einmal betrachtet wird. Nun wird die Ausbreitungsgeschwindigkeit auf $O(n^3)$ Punkte angewendet, indem der naheste von $O(n^2)$ Voxeln gesucht wird. Hierzu sind mit dem Standardalgorithmus (siehe Listing 3.6) $O(n^5)$ Schritte

5.1. LAUFZEITANALYSE DER VERFAHREN

nötig. Diese Schritte werden insgesamt k-mal durchgeführt. Also ergibt sich für die Gesamtlaufzeit des Algorithmus:

$$O(n^3) + O(n^3) + O(kn^2) + O(kn^5) = O(kn^5).$$
(5.1)

Ersetzt man den Standardalgorithmus zur Bestimmung des nahesten Punktes auf dem Zero Level Set durch den Approximationsalgorithmus aus Listing 3.7, so ergibt sich eine Gesamtlaufzeit von:

$$O(n^3) + O(n^3) + O(kn^2) + O(kn^3) = O(kn^3).$$
 (5.2)

Analog zur obigen Laufzeitanalyse folgt die Analyse des Narrow Band Algorithmus. Hier wird wiederum davon ausgegangen, dass die Anzahl der Voxel im Narrow Band $O(n^2)$ ist. Die Voxel die zum Narrow Band gehören, werden mit dem Standardalgorithmus in der Zeit $O(n^5)$ bestimmt. Danach muss die Ausbreitungsgeschwindigkeit , wie oben, auf $O(n^2)$ Punkte angewendet werden, indem der naheste von $O(n^2)$ Voxeln gesucht wird. Hierzu wird mit dem Standardalgorithmus (siehe Listing 3.6) eine Zeit $O(n^4)$ benötigt. Dies führt auf eine Gesamtlaufzeit des Narrow Band Algorithmus für k Durchläufe und l Narrow Band Iterationen von:

$$O(n^3) + O(n^3) + O(kn^5) + k \cdot (O(ln^2) + O(ln^4)) = O(kn^5) + O((l+k)n^4).$$
 (5.3)

Durch Anwendung der Approximationsalgorithmen 3.9 und 3.8, zur Bestimmung des Narrow Bands und zur Bestimmung des nahesten Punktes auf der aktiven Kontur C, ergibt sich analog eine Gesamtlaufzeit von:

$$O(n^3) + O(n^3) + O(kn^2) + k \cdot (O(ln^2) + O(ln^2)) = O(n^3) + O((l+k)n^2) \,. \eqno(5.4)$$

Ersetzt man nun noch die Initialisierung der aktiven Kontur durch ein beliebiges Objekt, so ergibt sich für die Initialisierung eine Laufzeit von $O(n^5)$ mit dem Standardalgorithmus (siehe Listing 3.4) beziehungsweise eine Laufzeit von $O(n^3)$ mit dem Approximationsalgorithmus aus Listing 3.5.

5.1.2 Der "energy minimization" Algorithmus

Wieder wird von einer Größe $n \times n \times n$ der Volumendaten ausgegangen. Nachdem $\varphi(x_1, x_2, x_3)$ wieder in Zeit $O(n^3)$ als vorzeichenbehaftete Abstandsfunktion einer Kugel mit gegebenem Radius berechnet wurde, werden zunächst $c_1(\varphi)$ und $c_2(\varphi)$ für $O(n^3)$ Voxel berechnet. Dazu benötigt man die Zeit $O(n^3)$. Genauso berechnet sich die Krümmung für alle Voxel in der Zeit $O(n^3)$. Nun muss jeder Voxel k-mal aktualisieren. Die Nullstellenmenge berechnen wir wie oben in $O(n^3)$. Insgesamt folgt eine Laufzeit von:

$$O(n^3) + O(kn^3) + O(kn^3) + O(kn^3) + O(kn^3) = O(kn^3).$$
(5.5)

Wieder wird die Größe $O(n^2)$ der aktiven Kontur C angenommen. Wird nun die vorzeichenbehaftete Distanzfunktion zusätzlich bei jeder Iteration reinitialisiert, so ergibt sich mit dem Standardalgorithmus (siehe Listing 3.4) zur Reinitialisierung folgende Komplexität der Gesamtlaufzeit

$$O(kn^3) + O(kn^5) = O(kn^5). (5.6)$$

Mit dem Approximationsalgorithmus aus Listing 3.5 zur approximativen Reinitialisierung der vorzeichenbehafteten Distanzfunktion ergibt sich eine Gesamtlaufzeit von

$$O(kn^3) + O(kn^3) = O(kn^3). (5.7)$$

Bemerkung 5.1.2.1

Durch die approximative Reinitialisierung ist es gelungen, wie die folgenden Beispiele zeigen werden, die Qualität der Segmentierung zu erhöhen ohne dabei die Komplexität des Verfahrens zu erhöhen.

5.2 Validierung der Verfahren

Aufgrund der besseren Darstellbarkeit der Ergebnisse werden die Verfahren zunächst an Beispielen für den zweidimensionalen Fall dargestellt. Um die Unterschiede sowie Vorteile der einzelnen Verfahren deutlich zu machen, werden die einzelnen Verfahren zuerst an erstellten Phantomdaten demonstriert und danach auf medizinische Bilddaten angewendet. Bei den zweidimensionalen Phantomdaten handelt es sich um ein Dreieck und eine Ellipse, wie in Abbildung 5.1 (a) dargestellt. In Abbildung 5.1 (b) sind die Phantomvolumendaten dargestellt. Nach der Untersuchung der Verfahren an den Phantomdaten, werden die Verfahren an MRT- und CT-Volumendatensätzen aus der medizinischen Praxis veranschaulicht. Die Bilder aller Beispiele finden sich zur bessern Übersicht im Anhang A dieser Diplomarbeit.

5.2.1 Das "edge stopping" Verfahren

Die Laufzeit des Global Extension Algorithmus (siehe Listing 3.1) liegt deutlich über der Laufzeit des Narrow Band Algorithmus (siehe Listing 3.2). Daher wird im ersten Beispiel gezeigt, dass das Ergebnis der Segmentierung nicht von der Wahl des verwendeten Algorithmus abhängt. Somit genügt es, in den darauffolgenden Beispielen nur noch den Narrow Band Algorithmus zu betrachten.

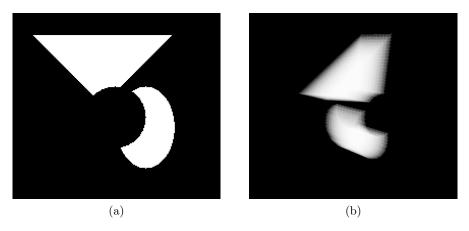


Abbildung 5.1: (a): Die zweidimensionalen Phantomdaten. (b) Die Phantomvolumendaten.

Beispiel 5.2.1.1

Um zu verdeutlichen, dass das Ergebnis der Segmentierung nicht von der Wahl des Verfahrens abhängt, wird der Global Extension Algorithmus mit dem Narrow Band Algorithmus an den in Abbildung 5.1 dargestellten Phantomdaten verglichen. Die zugrundeliegenden Phantomdaten haben eine Größe von 256×256 Pixel. Der Hintergrund hat den Grauwert 0 und die zu segmentierenden Objekte haben den Grauwert 4096.

Zur Segmentierung werden die beiden Standardalgorithmen aus Listing 3.1 und Listing 3.2 ohne eine zusätzliche Reinitialisierung benutzt. Beide Verfahren verwenden die stopping Funktion (3.15), da es in den Phantomdaten kein Bildrauschen gibt. Die gewählte stopping Funktion liegt an den Kanten sehr nahe bei Null und somit werden die Kanten während des Segmentierungsprozesses nicht so schnell übersprungen. Die Variable ϵ die den Einfluss der Krümmung bestimmt, wurde, wie bei den späteren Beispielen, auf 0,025 gesetzt. Als Schrittweite Δt wurde bei dem Global Extension Algorithmus 0,1 und beim Narrow Band Algorithmus 0,05 gewählt. Durch eine kleinere Wahl der Schrittweite wird die Güte der Segmentierung in den meisten Fällen erhöht. Wählt man diese jedoch zu klein, steht der höhere Rechenaufwand in keinem Verhältnis zur bessern Qualität der Segmentierung. Beim Narrow Band Algorithmus wurde zusätzlich, eine Bandbreite von zwei Pixeln und sieben Banditerationen pro Durchlauf gewählt.

Zur Bestimmung des Punktes auf der aktiven Kontur C mit minimalem Abstand und zur Bestimmung des Narrow Bands werden in diesem Beispiel die beiden Standardalgorithmen verwendet. Um die Laufzeit beim Global Extension Algorithmus

Global Extension Algorithmus		Narrow Band Algorithmus	
Iterationen	Laufzeit	Iterationen	Laufzeit
Initialisierung	4,642s	Initialisierung	2,216s
100 It.	559,413s	30 * 7 It.	$195{,}156\mathrm{s}$
200 It.	835,147s	60 * 7 It.	$335{,}147\mathrm{s}$
300 It.	1033,362s	90 * 7 It.	$488{,}157\mathrm{s}$
400 It.	1192,307s	120 * 7 It.	$649,\!440s$
500 It.	1276,387s	150 * 7 It.	$745{,}976\mathrm{s}$
600 It.	1320,743s	180 * 7 It.	$803,\!474s$
		210 * 7 It.	$825{,}957\mathrm{s}$
Gesamtlaufzeit	6222,001s	Gesamtlaufzeit	4045,523s

Tabelle 5.1: Vergleich der Laufzeiten beim Global Extension Algorithmus und beim Narrow Band Algorithmus.

zu verbessern, wurde der Bildbereich durch eine Bounding Box auf 152×152 Pixel verkleinert.

In der Bildfolge von Abbildung A.1 ist leicht zu erkennen, dass das Ergebnis der Segmentierung nach 600 Iterationen beim Global Extension Algorithmus und 210·7 = 1470 Iterationen beim Narrow Band Algorithmus identisch ist. Die höhere Anzahl der benötigten Iterationen beim Narrow Band Algorithmus ergibt sich aus der Wahl der unterschiedlichen Schrittweiten. Der hohe Laufzeitunterschied der beiden Verfahren wird in der Tabelle 5.1 verdeutlicht.

Im nächsten Beispiel werden die Ergebnisse der Segmentierung und die Laufzeiten aus Beispiel 5.2.1.1 mit den Ergebnissen einer Segmentierung des "edge stopping" Verfahrens verglichen, wobei diesmal die im vorangegangenen vorgestellten Approximationsalgorithmen verwendet werden.

Beispiel 5.2.1.2

Zur Segmentierung wird wieder von den zweidimensionalen Phantomdaten (siehe Abbildung 5.1 (a)) ausgegangen. Für den Einfluss der Krümmung durch die Variable ϵ wurde wieder der Wert 0,025 gewählt. Die Bandbreite des Narrow Bands wurde auf zwei Pixel gesetzt und sieben Banditerationen pro Iteration gewählt. Die Schrittweite Δt ist bei beiden Verfahren auf 0,05 gesetzt worden.

Bemerkung 5.2.1.1

Es hat sich im Laufe der Arbeit gezeigt, dass folgende Gleichung erfüllt sein sollte, um die Stabilität des Verfahrens sicher zu stellen.

Anzahl Banditerationen
$$\cdot \Delta t$$
 < Breite des Narrow Bands. (5.8)

NB-Alg ohne Reinitialisierung		NB-Alg mit Reinitialisierung	
Iterationen	Laufzeit	Iterationen	Laufzeit
Initialisierung	0,360s	Initialisierung	0,310s
30 * 7 It.	1,379s	30 * 7 It.	1,413s
60 * 7 It.	1,924s	60 * 7 It.	2,569s
90 * 7 It.	2,148s	90 * 7 It.	$2,\!226s$
120 * 7 It.	2,661s	120 * 7 It.	4,411s
150 * 7 It.	2,860s	150 * 7 It.	3,210s
180 * 7 It.	2,964s	180 * 7 It.	$3,\!823s$
210 * 7 It.	2,990s	210 * 7 It.	2,994s
		240 * 7 It.	3,610
Gesamtlaufzeit	17,286s	Gesamtlaufzeit	24,566s

Tabelle 5.2: Laufzeiten des Narrow Band Algorithmus ohne beziehungsweise mit Reinitialisierung.

Der Narrow Band Algorithmus wurde jetzt mit den Approximationsalgorithmen aus Listing 3.7 und Listing 3.9, zur Bestimmung des Punktes mit minimalen Abstands und zur Bestimmung des Narrow Bands, durchgeführt. Im zweiten Fall wurde zusätzlich eine Reinitialisierung der vorzeichenbehafteten Distanzfunktion $\varphi(x_1, x_2, x_3)$ nach jeweils 150 Iterationen durchgeführt.

Vergleicht man das Ergebnis der Segmentierung aus Abbildung A.2 mit den Ergebnissen der Segmentierung aus Beispiel 5.2.1.1, so ist kaum ein Unterschied feststellbar. Die Laufzeit, dargestellt in Tabelle 5.2, des Algorithmus hat sich jedoch deutlich verbessert. Es kann also behauptet werden, dass das Verfahren mit den Approximationsalgorithmen durch die bessere Laufzeit "anwendbar" geworden ist.

Vergleicht man das Ergebnis der Segmentierungen des Narrow Band Algorithmus einmal ohne und einmal mit Reinitialisierung, so kann kein Unterschied zwischen den segmentierten Objekten erkannt werden. Der Unterschied zwischen den beiden Verfahren wird im folgenden Beispiel verdeutlicht.

Beispiel 5.2.1.3

In diesem Beispiel werden die CT-Daten aus Abbildung 3.4 verwendet. Die Größe der Daten ist 256×256 . In Abbildung 3.4 wurde gezeigt, dass die aktive Kontur C bei zu vielen Iterationen die Kanten des gesuchten Objekts überspringt. Als Lösung wurde gefordert, dass die Startkontur nahe an dem gesuchten Objekt liegen muss. Da diese meist nicht bekannt ist oder der Aufwand, diese zu erstellen, zu groß ist, kann das Verfahren auch mit der Reinitialisierung der vorzeichenbehafteten Distanzfunktion durchgeführt werden. Durch die stetige Reinitialisierung wird, wie

Abbildung A.3 verdeutlicht, das Überspringen von starken Kanten bei zu vielen Iterationen verhindert.

In diesem Beispiel wurde die stopping Funktion (3.14) verwendet und die Variablen wie folgt gewählt:

$$\Delta t = 0.05$$
, $\epsilon = 0.025$, $NB_{size} = 2$, Band Iterationen: 15. (5.9)

Es wurde gezeigt, dass durch die Reinitialisierung das Überspringen von Kanten verhindert werden kann. Jedoch sollte, wenn möglich, immer eine Vorsegmentierung durchgeführt werden, um das Objekt zu segmentieren. Mit einer geeigneten Vorsegmentierung, zum Beispiel durch das Region-Grow Verfahren, werden zum einen die Zeit und zum anderen die Qualität der Segmentierung verbessert. Die Segmentierung des Objekts in diesem Beispiel dauerte 132,568s, wobei die vorzeichenbehaftete Distanzfunktion nach jeweils 100 Iteration neu berechnet wurde. Mit einer geeigneten Vorsegmentierung, wie in Abbildung 3.4, dauerte die Segmentierung bei gleichem Ergebnis 21,082s, da deutlich weniger Iterationen notwendig waren.

Beispiel 5.2.1.4

Bevor das "edge stopping" Verfahren an Volumendaten präsentiert wird, folgt ein letztes zweidimensionales Beispiel. Bei den Daten handelt es sich um ein MRT-Schnittbild aus der medizinischen Praxis. Ziel ist es, die Gehirnflüssigkeit, auch Liquor genannt, im Inneren des Schädels zu segmentieren. Die Größe der gegebenen Schnittbilddaten beträgt 176×256 Pixel.

Der erste Schritt der Segmentierung war eine (Vor-) Segmentierung mit dem Region-Grow Algorithmus, wie in Unterabbildung A.4 (a) zu sehen. Der zweite Schritt war die Initialisierung der vorzeichenbehafteten Distanzfunktion mit dem Approximationsalgorithmus 3.5, bezüglich der Punktmenge aus der (Vor-) Segmentierung. Die so entstandene aktive Kontur C ist in Unterabbildung A.4 (b) dargestellt.

Der dritte und letzte Schritt war die Evolution der aktiven Kontur C mit dem "edge stopping" Verfahren. Da es sich in dem Beispiel um Bilddaten aus der medizinischen Praxis handelt, wurde die stopping Funktion (3.14) verwendet und eine Reinitialisierung der vorzeichenbehafteten Distanzfunktion nach jeweils 130 Iterationen durchgeführt. Die verbleibenden Variablen wurden wie folgt gewählt:

$$\Delta t = 0.05$$
, $\epsilon = 0.025$, $NB_{size} = 2$, Band Iterationen: 18. (5.10)

In der Bildfolge der Abbildung A.4 ist die Evolution der aktiven Kontur C dargestellt. Es ist erkennbar, dass die Kanten des gesuchten Objekts gefunden werden und dass das Ergebnis der Segmentierung die gewünschte Gehirnflüssigkeit ist. Von der Initialisierung der vorzeichenbehafteten Distanzfunktion bis zum endgültigen Segment waren $40 \cdot 18 = 720$ Iterationen notwendig.

5.2. VALIDIERUNG DER VERFAHREN

Iterationen	Laufzeit
Initialisierung	15,195s
20 * 18 It.	220,774s
40 * 18 It.	231,994s
60 * 18 It.	$232{,}209\mathrm{s}$
Gesamtlaufzeit	700,172s

Tabelle 5.3: Laufzeiten des Narrow Band Algorithmus mit Reinitialisierung bei Volumendaten.

Beispiel 5.2.1.5

Als letztes Anwendungsbeispiel des "edge stopping" Verfahrens wird dieses an MRT-Volumendaten aus der medizinischen Praxis getestet. Die Größe der MRT-Volumendaten beträgt $176 \times 256 \times 256$ Voxel. Ziel ist es, wie auch im vorigem Beispiel, die Gehirnflüssigkeit im Inneren des Schädels zu segmentieren.

In den Abbildungen A.5 und A.6 ist die Evolution der aktiven Kontur C dargestellt. Zum leichteren Verständnis wurde die aktive Kontur jeweils in zwei Schnittbildern und einmal dreidimensional dargestellt.

Die Vorgehensweise der Segmentierung verlief analog zu der Segmentierung im zweidimensionalen Fall (Beispiel 5.2.1.4). Der (Vor-) Segmentierung mit dem Region-Grow Algorithmus (A.5 (a) - (c)) folgte die Initialisierung der vorzeichenbehafteten Distanzfunktion (A.5 (d) - (f)). Die Evolution benötigte von der Initialisierung bis zum endgültigen Segment $60 \cdot 18 = 1080$ Iterationen und ist in den Abbildung A.5 und A.6 dargestellt. In Tabelle 5.3 ist die Laufzeit der Evolution angegeben.

5.2.2 Der "energy minimization" Algorithmus

Im vorigen Abschnitt wurde die Funktonalität des "edge stopping" Verfahrens demonstriert, so dass die Demonstration des Verfahrens der Energieminimierung mit dem "energy minimization" Algorithmus folgt. Bevor das Verfahren an Volumendaten aus der medizinischen Praxis getestet wird, wird die Funktionalität zunächst wieder an den Phantomdaten aus Abbildung 5.1 dargestellt und die Vorteile gegenüber dem "edge stopping" Verfahren anhand Beispiel 5.2.2.1 verdeutlicht.

Beispiel 5.2.2.1

In Abbildung 3.3 wurde dargestellt, dass das "edge stopping" Verfahren Objekte nur dann segmentieren kann, wenn die Kanten des gesuchten Objekts durch einen hohen Gradienten gekennzeichnet sind. Das Problem wird in Abbildung A.7 (a)

und (b) noch einmal verdeutlicht. Hierbei wurde der Narrow Band Algorithmus mit einer Reinitialisierung nach jeweils 100 Iterationen und den Variablen

$$\Delta t = 0.05$$
, $\epsilon = 0.025$, $NB_{size} = 2$, Band Iterationen: 18 (5.11)

sowie der stopping Funktion (3.14) durchgeführt.

Mit der gleichen Startkontur und dem "energy minimization" Algorithmus wird das gesuchte Objekt trotz des kleinen Gradienten an den Objektkanten gut segmentiert. Hierbei wurde die Approximation (3.88) der Heaviside Funktion verwendet und die Variablen wurden wie folgt gewählt:

$$\Delta t = 0.05, \quad \epsilon = 1, \quad \lambda_1 = \lambda_2 = 1, \quad \mu = \nu = 0.$$
 (5.12)

In Abbildung A.7 (c) - (f) ist die Evolution der Kurve nach jeweils fünf Iterationen dargestellt. Wird das Ergebnis der Segmentierung nach 15 Iterationen (A.7 (f)) mit der Segmentierung nach 50 Iterationen (A.7 (g)) verglichen, so wird deutlich, dass die aktive Kontur während der Evolution nicht über die Objektgrenzen hinaus läuft.

Bemerkung 5.2.2.1

Die verwendete Approximation der Heaviside Funktion hat in diesem Beispiel keine Auswirkungen auf das Ergebnis der Segmentierung. Das gleiche Segment hätte auch eine Segmentierung ergeben, wenn die Approximation (3.80) oder (3.84) der Heaviside Funktion benutzt worden wäre.

Beispiel 5.2.2.2

In diesem Beispiel wird verdeutlicht, welche Auswirkungen die Wahl der approximierten Heaviside Funktion auf das Ergebnis der Segmentierung hat. Hierbei wird das Verfahren der Energieminimierung auf die Phantomdaten 5.1 angewendet, um die unterschiedlichen Ergebnisse der Segmentierung deutlich zu machen.

- 1. Fall: Wählt man die Approximation (3.84) der Heaviside Funktion und startet die Evolution der Kurve mit einem Kreis innerhalb eines der beiden Objekte (siehe Abbildung A.8 (a)), so führt die Minimierung des *Chan-Vese* Energiefunktional auf das globale Minimum. Dies bedeutet, dass das Ergebnis der Segmentierung die beiden Objekte der Phantomdaten sind, wie in Abbildung A.8 (d) dargestellt.
- 2. Fall: Mit der Approximation (3.80) der Heaviside Funktion und der gleichen Startkontur C wie im 1. Fall (siehe Abbildung A.8 (e)), führt die Minimierung des Chan-Vese Energiefunktionals auf das in Unterabbildung (h) dargestellte lokale Minimum.

	lterationen	Laufzeit
1. Fall	9	1,794s
2. Fall	36	6,628s
3. Fall	8	1,655s
4. Fall	14	2,979s

Tabelle 5.4: Laufzeiten und Anzahl der benötigten Iterationen beim "energy minimization" Algorithmus.

- 3. Fall: Auch mit der Approximation (3.88) der Heaviside Funktion und der gleichen Startkontur C (siehe Abbildung A.8 (i)) führt die Minimierung des Chan-Vese Energiefunktionals auf das in Unterabbildung (k) dargestellte lokale Minimum.
- 4. Fall: Nutzt man wie im 3. Fall die Approximation (3.88) der Heaviside Funktion und wählt als Startkontur einen Kreis der beide Objekte umschließt (A.8 (l)), so ergibt sich bei der Minimierung des Energiefunktionals das in Unterabbildung (m) dargestellte lokale Minimum.

Die Variablen wurden in allen vier Fällen wie folgt gewählt:

$$\Delta t = 0, 1, \quad \epsilon = 3, \quad \lambda_1 = \lambda_2 = 1, \quad \mu = \nu = 0.$$
 (5.13)

Eine Reinitialisierung der Distanzfunktion mit dem Approximationsalgorithmus aus Listing 3.5, wurde nach jeder Iteration vorgenommen. Die jeweils benötigte Anzahl der Schritte und die Laufzeiten der vier Fälle sind in Tabelle 5.4 dargestellt.

Bemerkung 5.2.2.2

Da sich die Ergebnisse der Segmentierung im 2. Fall und im 3. Fall nicht unterscheiden, die Anzahl der benötigten Schritte beim 2. Fall jedoch deutlich höher waren, wird die Approximation (3.80) der Heaviside Funktion in den nachfolgenden Beispielen nicht weiter betrachtet.

Beispiel 5.2.2.3

Die Ergebnisse aus dem Beispiel zuvor werden anhand der dreidimensionalen Volumendaten 5.1 (b) für den dreidimensionalen Fall bestätigt. Hierzu werden die Variablen, analog zu Beispiel 5.2.2.2, wie folgt gewählt:

$$\Delta t = 0, 1, \quad \epsilon = 3, \quad \lambda_1 = \lambda_2 = 1, \quad \mu = \nu = 0.$$
 (5.14)

Im 1. Fall wird wiederum das Verhalten des Verfahrens mit der Approximation (3.84) und im 2. Fall mit der Approximation (3.88) der Heaviside Funktion untersucht. Die Ergebnisse der beiden Fälle sind in Abbildung A.9 dargestellt. Es

	lterationen	Laufzeit
1. Fall	2	16,839s
2. Fall	6	45,627s

Tabelle 5.5: Laufzeiten und Anzahl der benötigten Iterationen beim dreidimensionalen "energy minimization" Algorithmus.

ist deutlich erkennbar, dass im 1. Fall erneut das globale Minimum gefunden wird und im 2. Fall ein lokales Minimum segmentiert wird. Die Anzahl der benötigten Iterationen sowie die Laufzeiten sind in der Tabelle 5.5 angeführt.

Bei den folgenden Beispielen aus der medizinischen Praxis wird immer die Approximation (3.88) der Heaviside Funktion benutzt. Da jeweils eine (Vor-) Segmentierung mit dem Region-Grow Algorithmus durchgeführt wird um eine Startkontur nahe dem gesuchten Objekt zu erhalten, ist hier gerade das lokale Minimum des *Chan-Vese* Energiefunktionals von Interesse. In allen Praxisbeispielen wird das Ergebnis der Segmentierung durch Energieminimierung mit dem Ergebnis des Region-Grow Algorithmus verglichen.

Beispiel 5.2.2.4

Ziel der Segmentierung in diesem Beispiel war es, die weiße Substanz^1 des Gehirns zu segmentieren. Die zugrundeliegenden MRT-Volumendaten haben eine Größe von $176 \times 256 \times 256$ Voxeln und sind in Abbildung 5.2 dargestellt.

Die Unterabbildungen (a) und (b) der Abbildung A.10 zeigen das Ergebnis der Segmentierung mit dem Region-Grow Algorithmus. Es ist deutlich zu erkennen, dass ein Großteil der weißen Substanz noch nicht segmentiert ist. In den Bildern (c) und (d) der Abbildung A.10 ist das Ergebnis der Segmentierung mit dem "energy minimization" Algorithmus dargestellt. Hierbei wurden zwei Iterationen des Algorithmus durchgeführt und die Variablen wie folgt festgelegt:

$$\Delta t = 0, 1, \quad \epsilon = 3, \quad \lambda_1 = 1, \quad \lambda_2 = 0, 1, \quad \mu = \nu = 1.$$
 (5.15)

Um das Segment zu erzeugen wurde das Ergebnis des Region-Grow Verfahrens als Startkontur des Minimierungsalgorithmus gewählt. Zur Initialisierung der Startkontur und zur Durchführung von zwei Iterationen benötigte das Verfahren 51,81s. Zwischen den beiden erzeugten Segmenten ist ein deutlicher Qualitätsunterschied.

¹Die weiße Substanz bezeichnet einen Teil des Zentralnervensystems und ist im Gehirn von der sogenannten grauen Substanz umgeben. (vgl. [Wik])

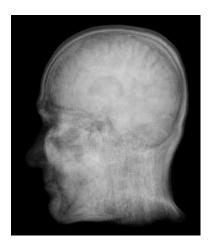


Abbildung 5.2: MRT-Volumendaten eines menschlichen Schädels.

Beispiel 5.2.2.5

Grundlage dieses Beispiels sind wieder die MRT-Volumendaten aus Abbildung 5.2. Ziel ist es, wie in Beispiel 5.2.1.5, die Gehirnflüssigkeit im Inneren des Schädels zu segmentieren. In den Unterabbildungen (a) und (b) ist zunächst einmal wieder das Ergebnis der Region-Grow Segmentierung dargestellt. Auch hier ist wieder deutlich zu sehen, dass ein Teil der Voxel des Segments noch nicht segmentiert ist. Die Unterabbildungen (c) und (d) zeigen das Ergebnis eines Schrittes des Minimierungsverfahrens mit den gewählten Variablen:

$$\Delta t = 0, 1, \quad \epsilon = 3, \quad \lambda_1 = \lambda_2 = 1, \quad \mu = \nu = 0.$$
 (5.16)

Als Startkontur wurde, wie schon in Beispiel 5.2.2.4, das Segment des Region-Grow Verfahrens gewählt, um die Energieminimierung mit einer guten Approximation des gesuchten Objekts zu starten.

Beispiel 5.2.2.6

In den beiden abschließenden Beispielen wird der Knochen in einem CT-Volumendatensatz segmentiert. Um die Qualitätsunterschiede zu verdeutlichen, wird zunächst nur den hinteren Teil des Schädels segmentiert. Die Bildfolge (a), (c) und (e) zeigt das Ergebnis des Region-Grow Algorithmus. Es ist leicht erkennbar, dass in einigen Regionen Voxel segmentiert wurden, die nicht zum Schädel gehören. In anderen Regionen wurden jedoch auch einige Voxel, die zum Schädel gehören, nicht segmentiert. In der Bildfolge (b), (d) und (f) ist das Ergebnis der Energieminimierung nach zwei Iterationen dargestellt. Als Startkontur wurde wieder das mit dem Region-Grow Algorithmus erzeugte Segment gewählt. Man erkennt deutlich, dass die fehlenden Voxel bereits nach zwei Iterationen segmentiert wurden und die Voxel, die nicht zum Schädel gehören nicht mehr segmentiert sind.

Die zugrundeliegenden Volumendaten haben eine Größe von $256 \times 256 \times 113$ Voxel. Um die Segmentierung auf den hinteren Schädel zu beschränken, wurde eine Bounding Box der Größe $166 \times 74 \times 94$ Voxel in das Objekt gelegt.

Abbildung A.13 zeigt das Ergebnis der beiden Verfahren, angewendet auf den gesamten Volumendatensatz. Man erkennt auch hier wieder deutlich die Qualitätsunterschiede der beiden Verfahren. Die Energieminimierung benötigte wieder zwei Iterationen um aus der Vorsegmentierung, mit dem Region-Grow Algorithmus, das endgültige Segment zu erzeugen. Die Variablen in diesem Beispiel wurden wie schon im Beispiel zuvor gewählt:

$$\Delta t = 0, 1, \quad \epsilon = 3, \quad \lambda_1 = \lambda_2 = 1, \quad \mu = \nu = 0.$$
 (5.17)

Die Initialisierung der Startkontur und das Durchführen der zwei Iterationsschritte benötigte 48,583s.

5.2. VALIDIERUNG DER VERFAHREN

Kapitel 6

Fazit und Ausblick

Ziel der vorliegenden Arbeit war die Implementierung, Optimierung und Weiterentwicklung der in Kapitel 3 vorgestellten regionenbasierten Segmentierungsmodelle. Anhand dieser soll es dem Benutzer der Verfahren erleichtert werden, die relevanten Informationen in einem medizinischen Volumendatensatz zu segmentieren. Diesbezüglich war zunächst eine deutliche Verbesserung der Laufzeit der beiden Verfahren nötig, da die Standardverfahren insbesondere bei medizinischen Volumendaten nicht anwendbar sind. Der zweite wichtige Schritt war die Anpassung der beiden Modelle an medizinische Volumendaten, die sich zum Beispiel durch einen sehr großen Grauwertbereich auszeichnen. Im Folgenden werden die erzielten Ergebnisse dargestellt und ein Ausblick auf mögliche zukünftige Arbeiten gegeben.

6.1 Zusammenfassung

Betrachtet man die in Kapitel 5 vorgestellten Ergebnisse am Beispiel der Phantomdaten oder der realen medizinischen Datensätze, so kommt man bezüglich des "edge stopping" Verfahrens resümierend zu zwei Feststellungen. Zum einen ist es während der Arbeit gelungen, durch die neu entwickelten Approximationsalgorithmen die Laufzeit der Algorithmen so weit zu verbessern, dass die vorgestellten Verfahren "praktikabel/anwendbar" geworden sind. Zum anderen wurde es durch die entwickelten Erweiterungen ermöglicht, Objekte zu segmentieren, die mit den Standardalgorithmen nicht ohne weiteres zu segmentieren waren. Hier brachte die stetige Reinitialisierung der Distanzfunktion und die mögliche Verwendung einer Vorsegmentierung deutliche Vorteile gegenüber den Standardalgorithmen. Durch den entwickelten Approximationsalgorithmus (siehe Listing 3.5) konnte auch hier

6.2. AUSBLICK

eine Verbesserung der Laufzeit erreicht werden. Der Ansatz der vorgestellten neuen stopping Funktion (siehe (3.92)) war nur in seltenen speziellen Beispielen eine Verbesserung.

Durch die Reinitialisierung nach einer bestimmten Anzahl von Iterationen ist es möglich, das Hinausspringen der aktiven Kontur auch über starke Kanten zu verhindern. Somit ist es mit dem "edge stopping" Verfahren möglich geworden, Objekte mit einer schlechten Startkontur zu segmentieren. Durch die Möglichkeit der Vorsegmentierung konnte die Anzahl der Iterationen und somit die Laufzeit des Verfahrens bis zum finalen Segment noch einmal deutlich verbessert werden.

Im Gegensatz zum "edge stopping" Verfahren arbeitete der "energy minimization" Algorithmus bereits effizient genug, um in der medizinischen Praxis angewendet werden zu können. Daher hatte an dieser Stelle die Weiterentwicklung des *Chan-Vese* Energiefunktionals eine größere Bedeutung als die Optimierung der Laufzeit des Verfahrens. Hier führte zum einen die approximierte Reinitialisierung der Distanzfunktion nach jeder Iteration und zum anderen die Vorsegmentierung zu einer Steigerung der Qualität der Segmentierung. Durch die Vielzahl der wählbaren Parameter ist hier ein Framework entstanden, dass es ermöglicht, das Verfahren auf eine große Zahl von unterschiedlichsten Datensätzen erfolgreich anzuwenden. Durch die Möglichkeit der Vorsegmentierung des gesuchten Segments, mit Hilfe des Region-Grow Verfahrens, hat sich hier insbesondere die neue Approximation (siehe (3.88)) der Heaviside Funktion als sehr hilfreich erwiesen. Es hat sich gezeigt, dass bei einer "guten" Vorsegmentierung lediglich ein paar Iterationen notwendig sind, um ein lokales Minimum des gewählten Funktionals zu erreichen. Dies ist gleichbedeutend mit einer sehr geringen Zeit, die zur Segmentierung erforderlich ist.

6.2 Ausblick

Nicht untersucht werden konnte in dieser Arbeit, ob die Qualität beziehungsweise Güte der erzielten Segmentierungen lokal ausreichend für die medizinische Praxis ist. Jedoch ist es gelungen die Laufzeiten der Verfahren so weit zu verbessern, dass sie im nächsten Schritt in der medizinischen Praxis hinreichend getestet werden können. Hierzu wäre es sinnvoll, die erzielten Segmentierungen von einem Experten begutachten zu lassen oder die Segmentierungen mit manuell erstellten Segmenten zu vergleichen.

Zur Untersuchung der Qualität der Segmentierung ist, in Zusammenarbeit mit der Neuroradiologie der Medizinischen Hochschule Hannover, in nächster Zeit die Veröffentlichung eines gemeinsamen Papers geplant.

Damit die in dieser Arbeit vorgestellten Verfahren weiter verbessert werden, bieten sich zahlreiche Möglichkeiten, da trotz der erzielten Ergebnisse auch einige Schwachstellen der Verfahren existieren. So könnten zum einen die verwendeten numerischen Verfahren durch bessere Verfahren ersetzt werden. Hier ist es denkbar, dass durch die Verwendung eines Runge-Kutta Verfahrens, anstatt des verwendeten eulerschen Polygonzugverfahrens, zur numerischen Lösung der auftretenden partiellen Differentialgleichungen eine Qualitätsverbesserung mit minimal höherem Rechenaufwand möglich wäre.

Um die Automatisierung der Segmentierung zu erhöhen ist es auch erforderlich, ein geeignetes Abbruchkriterium zu definieren, so dass der Benutzer die benötigte Anzahl der Iterationen nicht mehr erahnen muss. Des Weiteren ist es für die Praxis zwingend notwendig, die Vielzahl der einstellbaren Parameter im Voraus zu schätzen oder idealerweise im Voraus zu bestimmen, da das Ergebnis der Segmentierung stark von den gewählten Parametern abhängt. Insbesondere die Gewichtungsparameter beim Verfahren der Energieminimierung sind schwierig zu bestimmen und meist nur durch mehrere Tests zu bestimmen, was wiederum einige Zeit beansprucht. Aber auch die optimale Schrittweite Δt der Verfahren kann bei nicht geeigneter Wahl zum falschen Ergebnis führen. Bei zu großer Wahl kann es passieren, dass das gesuchte Objekt schlecht oder gar nicht segmentiert wird und bei zu kleiner Wahl ist eine große Zahl von Iterationen notwendig, so dass das Verfahren unbrauchbar wird.

Der regionenbasierte Ansatz der Energieminimierung kann leicht zu einem modellbasierten Ansatz erweitert werden, um die Qualität der Segmentierung zu verbessern. Hier ist es denkbar eine Art Straffunktion einzuführen, die verhindert, dass im Verlauf der Segmentierung zu viele nicht zusammenhängende Segmente erzeugt werden. Eine weitere Idee wäre das Verwenden eines optimalen Modells des gesuchten Objekts. Dann könnte das Verfahren der Energieminimierung so erweitert werden, dass sich die Kontur nicht allzu weit von dem vorgegebenen Modell entfernen kann.

Weiterhin wäre noch zu untersuchen, inwieweit sich die vorgestellten Verfahren in anderen Bereichen der Praxis zur Segmentierung eignen und ob hier eine weitere anwendungsspeziefische Anpassung erforderlich ist.

6.3 Fazit

Es ist in dieser Arbeit gelungen, das "edge stopping" Verfahren durch die neu entwickelten Approximationsalgorithmen, wie zum Beispiel den Zwiebelschalenalgorithmus zur Bestimmung der Punkte auf der aktiven Kontur C mit minimalem

6.3. FAZIT

Abstand, "anwendbar" zu machen. Somit ist es mit diesem Ansatz möglich geworden, Objekte mit starken Kanten in medizinischen Volumendaten zu segmentieren. Zu diesen Objekten kann zum Beispiel der Knochen in einem CT-Datensatz oder die Gehirnflüssigkeit in einem MRT-Datensatz gezählt werden. Weiterhin ist es durch die Weiterentwicklung des *Chan-Vese* Energiefunktionals gelungen, Objekte mit weichen Objektübergängen gut zu segmentieren. Mit diesem Ansatz ist man jetzt in der Lage zum Beispiel die weiße oder graue Masse im Gehirn zu segmentieren oder aber, wie beim "edge stopping" Verfahren, Objekte mit starken Kanten.

Ob jedoch die Qualität lokal ausreichend für die Praxis ist, wird das mit der Neuroradiologie der Medizinischen Hochschule Hannover geplante Paper zeigen.

Anhang A

Bilder der Beispiele

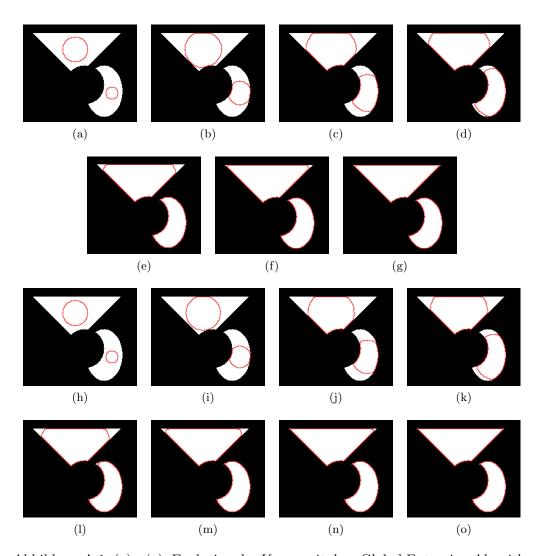


Abbildung A.1: (a) - (g): Evolution der Kurve mit dem Global Extension Algorithmus nach jeweils 100 Iterationen.

(h) - (o): Evolution der Kurve mit dem Narrow Band Algorithmus nach jeweils $30\cdot 7$ Iterationen.

Bilder aus Beispiel (5.2.1.2)

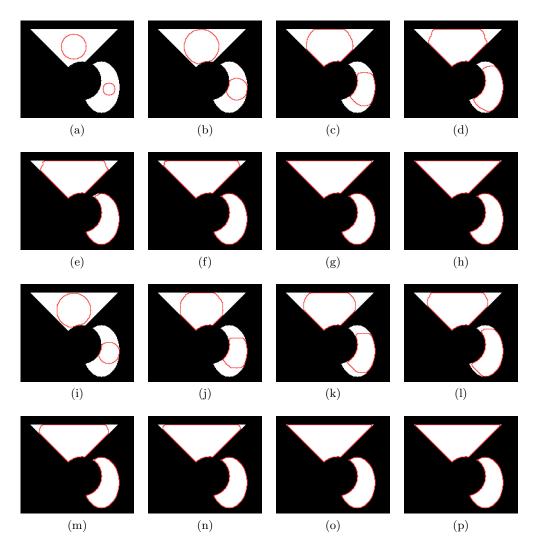


Abbildung A.2: (a) - (h): Evolution der Kurve mit dem Narrow Band Algorithmus nach jeweils $30 \cdot 7$ Iterationen ohne Reinitialisierung der Distanzfunktion. (a), (i) - (p): Evolution der Kurve mit dem Narrow Band Algorithmus nach jeweils $30 \cdot 7$ Iterationen mit Reinitialisierung der Distanzfunktion nach jeweils 150 Iterationen.

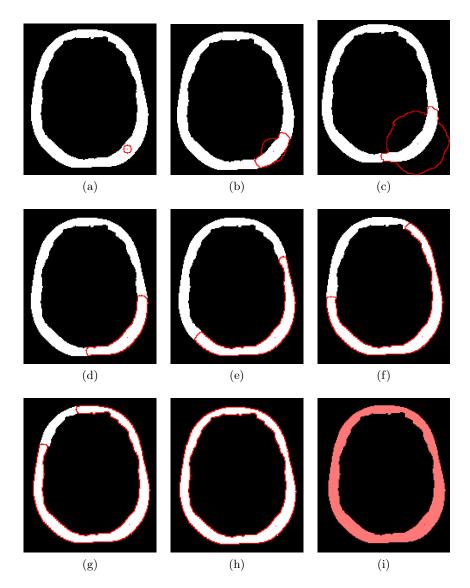


Abbildung A.3: (a) - (c): Evolution der Kurve mit dem Narrow Band Algorithmus nach jeweils $50 \cdot 15$ Iterationen ohne Reinitialisierung der Distanzfunktion. (a), (d) - (i): Evolution der Kurve mit dem Narrow Band Algorithmus nach jeweils $100 \cdot 15$ Iterationen mit Reinitialisierung der Distanzfunktion nach jeweils 150 Iterationen.

Bilder aus Beispiel (5.2.1.4)

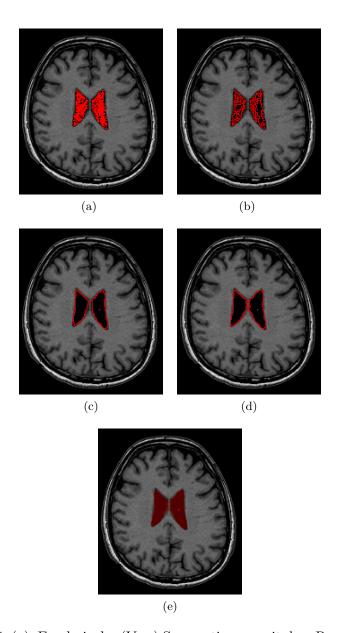


Abbildung A.4: (a): Ergebnis der (Vor-) Segmentierung mit dem Region-Grow Verfahren.

- (b): Darstellung der Aktiven Kontur C bezüglich der (Vor-) Segmentierung.
- (c), (d): Evolution der Kurve nach jeweils $20 \cdot 18$ Iterationen.
- (e): Darstellung des segmentierten Objekts nach $40\cdot 18$ Iterationen.

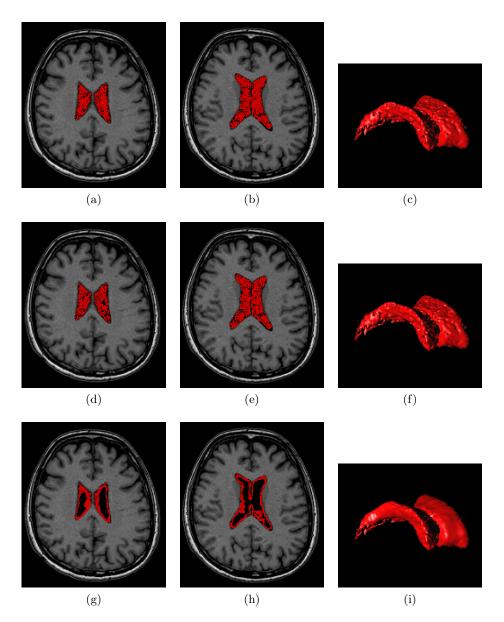


Abbildung A.5: (a) - (c): Ergebnis der (Vor-) Segmentierung mit dem Region-Grow Verfahren.

- (d) (f): Darstellung der aktiven Kontur C bezüglich der (Vor-) Segmentierung.
- (g) . (i): Darstellung der aktiven Kontur nach $20\cdot 18$ Iterationen.

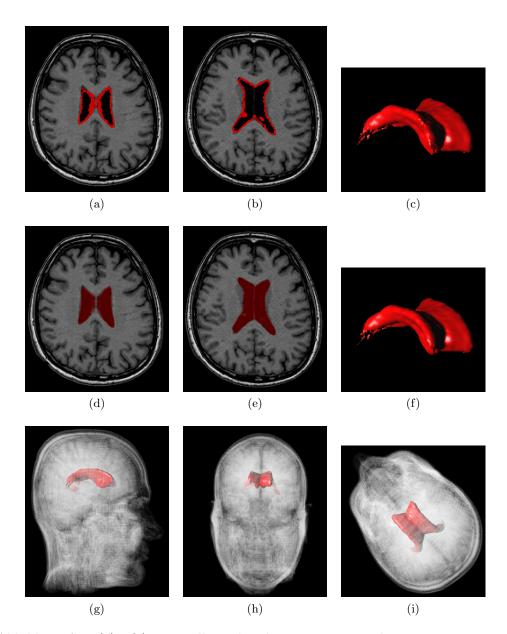


Abbildung A.6: (a) - (c): Darstellung der aktiven Kontur nach $40 \cdot 18$ Iterationen. (d) - (f): Darstellung des Segments nach $60 \cdot 18$ Iterationen .

(g) - (h): Dreidimensionale Darstellung des Segments innerhalb der Volumendaten.

Bilder aus Beispiel (5.2.2.1)

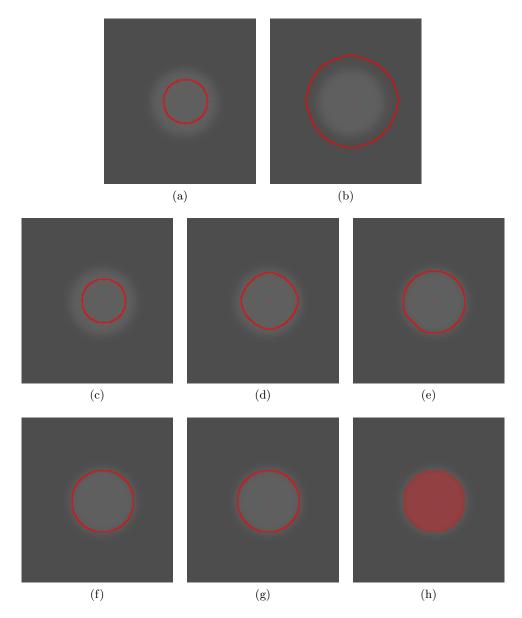


Abbildung A.7: (a) und (b): Evolution der Kurve mit den Narrow-Band Algorithmus.

(c) - (f): Evolution der Kurve mit dem Verfahren der Energieminimierung; (g): Darstellung der aktiven Kontur nach weiteren Iterationen; (f): Darstellung des erzeugten Segments.

Bilder aus Beispiel (5.2.2.2)

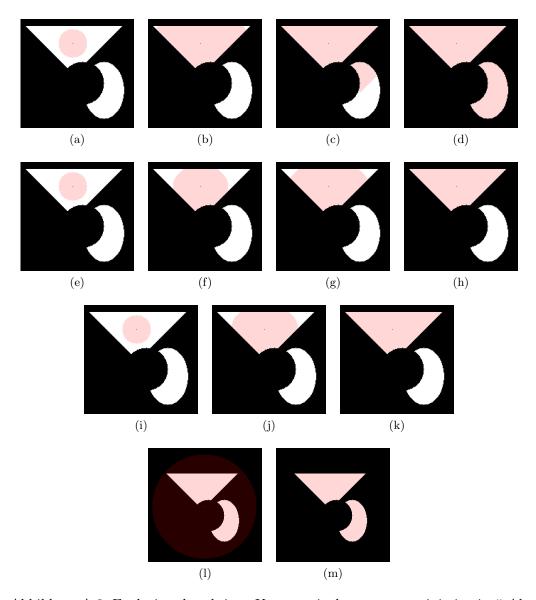


Abbildung A.8: Evolution der aktiven Kontur mit dem "energy minimization" Algorithmus unter Verwendung der verschiedenen Approximationen der Heaviside Funktion beziehungsweise unter Verwendung verschiedener Startkonturen.

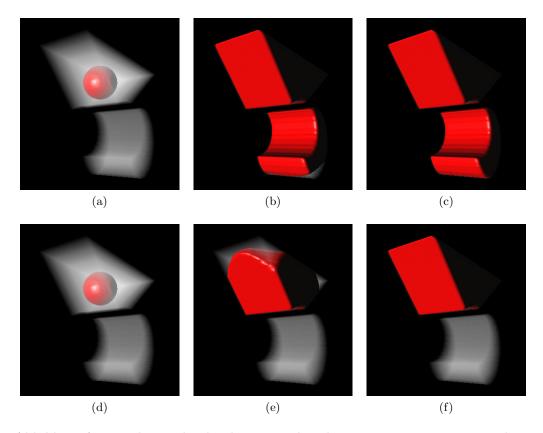


Abbildung A.9: Evolution der dreidimensionalen aktiven Kontur unter Verwendung verschiedener Approximationen der Heaviside Funktion.

Bilder aus Beispiel (5.2.2.4)

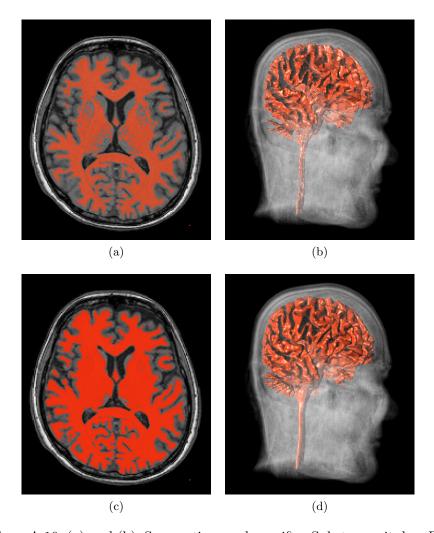


Abbildung A.10: (a) und (b): Segmentierung der weißen Substanz mit dem Region-Grow Algorithmus.

(c) und (d): Ergebnis der Segmentierung mit dem Verfahren der Energieminimierung.

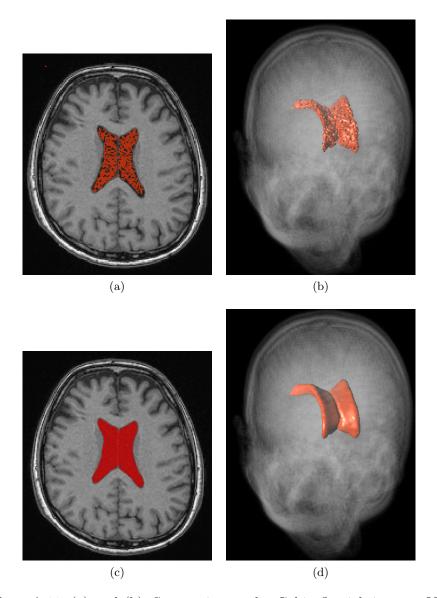


Abbildung A.11: (a) und (b): Segmentierung der Gehirnflüssigkeit unter Verwendung des Region-Grow Algorithmus.

(c) und (d): Ergebnis der Segmentierung mit dem "energy minimization" Algorithmus.

Bilder aus Beispiel (5.2.2.6)

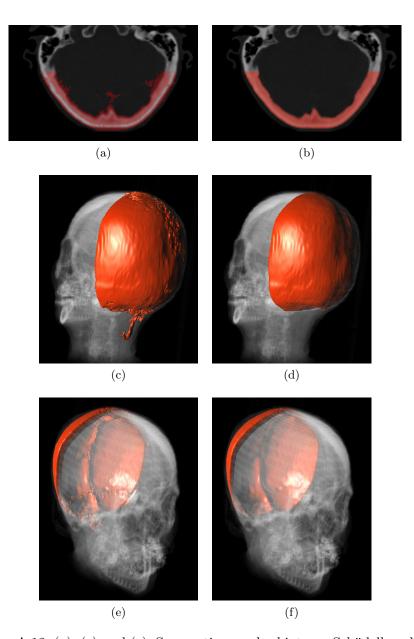


Abbildung A.12: (a), (c) und (e): Segmentierung des hinteren Schädelknochens mit dem Region-Grow Algorithmus.

(b), (d) und (f): Ergebnis der Segmentierung mit dem Verfahren der Energieminimierung.

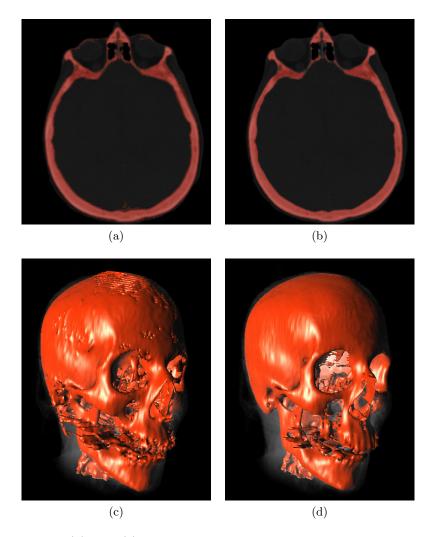


Abbildung A.13: (a) und (c): Segmentierung des Knochens unter Verwendung des Region-Grow Algorithmus.

(b) und (d): Ergebnis der Segmentierung mit dem "energy minimization" Algorithmus.

ANHANG B. NOTATION

Anhang B

Notation

Operatoren

- Betrag von x für $x \in \mathbb{R}$. |x|
- Norm des Vektors x für $x \in \mathbb{R}^N$, $|x| = \sqrt{x_1^2 + \dots x_N^2}$. |x|
- Faltung, zum Beispiel Faltung der Volumendaten mit einem Filter.

Ableitungen

- $\alpha(t)$
- $\alpha'(t)$
- Funktion, $\alpha(t): \mathbb{R} \to \mathbb{R}^N$. $= \frac{\partial \alpha}{\partial t} = (\alpha'_1(t), \dots, \alpha'_N(t))^T, \text{ erste Ableitung.}$ $= \frac{\partial^2 \alpha}{\partial t^2} = (\alpha''_1(t), \dots, \alpha''_N(t))^T, \text{ zweite Ableitung.}$ Funktion, $\varphi(x): \mathbb{R}^N \to \mathbb{R}$. $\alpha''(t)$
- $\varphi(x)$
- $=\frac{\partial \varphi}{\partial x_i}$, partielle Ableitung der Funktion $\varphi(x)$ nach x_i . $\partial_{x_i}\varphi$
- $=(\partial_{x_1}^{\cdot}\varphi,\ldots,\partial_{x_N}),$ Gradient. $\nabla \varphi$
- $=\partial_{x_1}f_1+\ldots+\partial_{x_N}f_N$, Divergenz eines Vektorfeldes $f:\mathbb{R}^N\to\mathbb{R}^N$. $\operatorname{div}(f)$
- $\Delta_{x_i}^{x_i}$ $\Delta_{x_i}^{x_i}$ Finiter Vorwärtsdifferenzenquotient nach x_i .
- Finiter Rückärtsdifferenzenquotient nach x_i .
- Δ^{x_i} zentraler Differenzenquotient nach x_i .

Bilder

- := $[0,1]^N \subset \mathbb{R}^N$, kontinuierlicher Bildraum. Ω
- $:= (x_1, \ldots, x_N) \in \Omega$, Koordinatenvektor.
- kontinuierliche Grauwertfunktion eines Bildes, $u_0: \Omega \to \mathbb{R}$. $u_0(x)$
- := $\{0, \dots, n_{x_1}\} \times \dots \times \{0, \dots, n_{x_N}\}$, diskreter Bildraum.
- $:= (x_1, \ldots, x_N) \in \hat{\Omega}$, diskreter Bildpunkt.
- $102 \\ \hat{u_0}(x)$ diskrete Grauwertfunktion eines Bildes, $\hat{u_0}: \hat{\Omega} \to G$.
 - $:= \{0, \ldots, x_G\},$ diskreter Grauwertbereich.

Segmentierung

```
E_{\rm snake}
             Kass, Witkin, Terzopoulos Modell der Segmentierung.
  E_{\rm int}
             interne Energie der Segmentierung.
  E_{\rm ext}
             externe Energie der Segmentierung.
  E_{\rm CV}
             Chan-Vese Energiefunktional der Segmentierung.
  E_{\rm MS}
             Mumford-Shah Energiefunktional der Segmentierung.
  E_{\text{CV}^*}
             verallgemeinertes Chan-Vese Energiefunktional der Segmentierung.
             := \Omega \to \mathbb{R}, vorzeichenbehaftete Abstandsfunktion (Level Set Funktion).
    \varphi
             := \Omega \times \mathbb{R} \to \mathbb{R}, vorzeichenbehaftete Abstandsfunktion als Zeitfunktion
    \varphi
             (Level Set Funktion).
    F
            := F_A + F_G, Ausbreitungsgeschwindigkeit.
g(|\nabla u_0|)
            : \Omega \rightarrow [0,1], stopping Funktion.
            := \{x \mid x \in \Omega \text{ und } \varphi(x) = 0\}, \text{ aktive Kontur der Segmentierung.}
            =\varphi(x_1,\ldots,x_n,k\Delta t), diskretisierte Level Set Funktion.
            Zeitschritt der Evolution.
NB(C) := \{ y \in \Omega \mid \exists x \in C, \text{ mit } y \in B_{\delta}(x) \}, \text{ Narrow Band um } C.
 B_{\delta}(x)
            := \{z \in \hat{\Omega} \mid |x - z| \le \delta\}, \text{ Kugel um } x.
NB_{size}
            =\delta, Größe des Narrow Bands.
           H(z)
  \delta(z)
             \in C^2\Omega, Approximation der Heaviside Funktion.
   H_{\epsilon}
             \in C\Omega, Approximation der Dirac-Delta Funktion.
```

Anhang C

Beiligende CD

Auf der beiligenden CD ist zu finden:

- Quellcode des YaDiV Programms.
- Ausführbare jar-Datei des YaDiV Programms.
- Die in dieser Arbeit verwendete Literatur (soweit als Datei verfügbar).
- Ein CT-Volumendatensatz eines menschlichen Schädels.
- Dieses Dokument als pdf-Datei.

Abbildungsverzeichnis

1.1	Prinzip der Computer-Tomographie	4
1.2	Beispiel der Min-Max Segmentierung und der Region-Grow Segmentierung.	7
2.1	Die Kontur als Nullstellenmenge.	17
2.2	Evolution der aktiven Kontur	18
2.3	Die vorzeichenbehaftete Abstandsfunktion eines Kreises	19
3.1	Einbettung der Kontur in die vorzeichenbehaftete Distanzfunktion.	24
3.2	Verschiedene Definitionen des Zero Level Sets	30
3.3	Das "edge stopping" Verfahren bei schwachen Objektübergängen. .	34
3.4	Das "edge stopping" Verfahren bei starken Kanten	36
3.5	Die vier möglichen Positionen einer Kurve und die Auswirkungen auf die Energien F_1 und F_2	38
3.6	Verschiedene Approximationen der Heaviside Funktion	49
3.7	Verschiedene diskrete Abstandsfunktionen	53
3.8	Verschiedene erweiterte Ausbreitungsgeschwindigkeiten	56
4.1	Das Programm YaDiV	64
5.1	Die Phantomdaten und Phantomvolumendaten	72
5.2	MRT-Volumendaten eines menschlichen Schädels	80
A.1	Bildfolge zum Beispiel (5.2.1.1)	88
A.2	Bildfolg zum Beispiel (5.2.1.2)	89

ABBILDUNGSVERZEICHNIS

A.3	Bildfolg zum	Beispiel	(5.2.1.3)	3)		 					 90
A.4	Bildfolg zum	Beispiel	(5.2.1.4)	1)		 					 91
A.5	Bildfolg zum	Beispiel	(5.2.1.5)	(1.	Teil)	 					 92
A.6	Bildfolg zum	Beispiel	(5.2.1.5)	(2.	Teil)	 					 93
A.7	Bildfolg zum	Beispiel	(5.2.2.1			 					 94
A.8	Bildfolg zum	Beispiel	(5.2.2.2)	2)		 					 95
A.9	Bildfolg zum	Beispiel	(5.2.2.3)	3)		 					 96
A.10	Bildfolg zum	Beispiel	(5.2.2.4	ł)		 					 97
A.11	Bildfolg zum	Beispiel	(5.2.2.5)	5)		 					 98
A.12	Bildfolg zum	Beispiel	(5.2.2.6	s) (1.	Teil)	 					 99
A.13	Bildfolg zum	Beispiel	(5.2.2.6	i) (2.	Teil)	 					 100

Tabellenverzeichnis

5.1	Vergleich der Laufzeiten beim Global Extension Algorithmus und beim Narrow Band Algorithmus	73
5.2	Laufzeiten des Narrow Band Algorithmus ohne beziehungsweise mit Reinitialisierung.	74
5.3	Laufzeiten des Narrow Band Algorithmus mit Reinitialisierung bei Volumendaten.	76
5.4	Laufzeiten und Anzahl der benötigten Iterationen beim "energy minimization" Algorithmus	78
5.5	Laufzeiten und Anzahl der benötigten Iterationen beim dreidimensionalen "energy minimization" Algorithmus	79

TABELLENVERZEICHNIS

Listings

3.1	Global Extension Algorithmus	32
3.2	Narrow Band Algorithmus	33
3.3	Energy Minimization Algorithm	47
3.4	Berechnung der diskreten vorzeichenbehafteten Distanzfunktion	51
3.5	Berechnung einer approximierten vorzeichenbehafteten Distanzfunktion	52
3.6	Berechnung des Voxel mit minimalem Abstand	55
3.7	Approximierte Berechnung der Voxel mit minimalem Abstand	55
3.8	Approximierte Berechnung des Voxel mit minimalem Abstand innerhalb des Narrow Bands	57
3.9	Approximierte Berechnung aller Voxel des Narrow Bands	57
3.10	Energy Minimization Algorithm with Narrow Band	62

LISTINGS

Literaturverzeichnis

- [AASS95] ADALSTEINSSON, David; ADALSTEINSSON, David; SETHIAN, James A.; SETHIAN, James A.: A fast level set method for propagating interfaces. In: Journal of Computational Physics 118 (1995), 269—277. http://math.berkeley.edu/~sethian/2006/Papers/sethian.adalsteinsson.narrowband.pdf
- [AS04] ALEXANDROV, O.; SANTOSA, F.: A topology-preserving level set method for shape optimization. http://www.math.umn.edu/~santosa/papers/logbar.pdf. Version: September 2004
- [BBW06] BROX, Thomas; BRUHN, Andrés; WEICKERT, Joachim: Variational motion segmentation with level sets. In: in ECCV, Springer, 2006, 471–483. http://www.cs.ualberta.ca/~jag/papersVis2/levsetReadGr/Brox06flowseg.pdf
- [BL03] BISCHOFF, S.; L.KOBBELT: Sub-Voxel topology control for Level-Set surfaces. In: EUROGRAPHICS 03 proceedings 22 (2003), Nr. 3, 273-280. http://www.graphics.rwth-aachen.de/uploads/media/topctrl.pdf
- [Car98] CARMO, M.P. do: Differentialgeometrie von Kurven und Flächen. 3. Auflage. Braunschweig: Vieweg, 1998. ISBN 3-528-27255-4
- [CC93] COHEN, Laurent D.; COHEN, Isaac: Finite element methods for active contour models and balloons for 2-D and 3-D images. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15 (1993), 1131–1147. http://www.ceremade.dauphine.fr/~cohen/mypapers/IsaacPAMI93.ps.gz
- [CMOZ96] CHAN, T.; MERRIMAN, B.; OSHER, S.; ZHAO, Hong-Kai: A variational level set approach to multiphase motion. In: Journal of Computational Physics 127 (1996), 179-195. http://math.uci.edu/~zhao/publication/mypapers/ps/triple.ps
- [Coh91] COHEN, L.D.: On active contour models and balloons. In: Computer Vision, Graphics, and Image Processing 53 (1991), March, Nr. 2, 211-218. ftp://ftp-sop.inria.fr/epidaure/Publications/Cohen/CVGIP91.ps.Z

LITERATURVERZEICHNIS

- [CV99] CHAN, Tony; VESE, Luminita: An active contour model without edges. In: Int. Conf. Scale-Space Theories in Computer Vision, 1999, 141–151. http://www.math.ucla.edu/~lvese/PAPERS/SS99.pdf
- [CV00] Chan, Tony F.; Vese, Luminita A.: Image segmentation using level sets and the piecewise-constant Mumford-Shah model / Tech. Rep. 0014, Computational Applied Math Group. Version: 2000. ftp: //ftp.math.ucla.edu/pub/camreport/cam00-14.ps.gz. 2000. Forschungsbericht
- [CV01] CHAN, Tony; VESE, Luminita: Active contours without edges. In: IEEE Transactions on Image Processing 10 (2001), February, Nr. 2, 266-277. http://www.math.ucla.edu/~lvese/PAPERS/IEEEIP2001. pdf
- [DR05] DROSKE, Marc; RING, Wolfgang: A Mumford-Shah level-set approach for geometric image registration. In: SIAM Appl. Math 2007 (2005). ftp://ftp.math.ucla.edu/pub/camreport/cam05-63.pdf
- [Dös06] DÖSSEL, O.: Bildgebende Verfahren in der Medizin: Von der Technik zur medizinischen Anwendung. 1. Auflage. Berlin: Springer, 2006. ISBN 3-540-66014-3
- [EG92] EVANS, L.C.; GARIEPY, R.F.: Measure Theory and Fine Properties of Functions. Boca Flaton, FL: CRC, 1992
- [Fri06] FRIESE, Karl-Ingo: Folien der Lehrveranstaltung 3D Datenverarbeitung in der Medizin. Universität Hannover, Fachgebiet Graphische Datenverarbeitung, 2006
- [Her04] HERRMANN, N.: Höhere Mathematik. München: Oldenbourg Wissenschaftsverlag, 2004. ISBN 3-486-27498-8
- [HSZ03] HACKBUSCH, W.; SCHWARZ, H.R.; ZEIDLER, E.: Teubner Taschenbuch der Mathematik. Wiesbaden: B.G. Teubner Verlag, 2003. ISBN 3-519-20012-0
- [HXP03] HAN, X.; Xu, C.; Prince, J.L.: A topology preserving level set method for geometric deformable models. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25 (2003), June, Nr. 6, 755-768. http://iacl.ece.jhu.edu/~xhan/papers/PAMI03.pdf
- [Jäh05] Jähne, B.: Digitale Bildverarbeitung. Springer-Verlag, 2005
- [KWT88] KASS, M.; WITKIN, A.; TERZOPOULOS, D.: Snakes: Active Contour models. In: *International Journal of Computer Vision* 1 (1988), Nr. 4, 321–331. http://www.cs.ucla.edu/~dt/papers/ijcv88/ijcv88.pdf

- [Küh05] KÜHNEL, W.: Differentialgeometrie. 3. Auflage. Braunschweig: Vieweg, 2005. ISBN 3-834-80023-6
- [Lie06] LIEDTKE, Prof. Dr.-Ing. C.-E.: Folien der Lehrveranstaltung Digitale Bildverarbeitung. Leibniz Universität Hannover, Institut für Informationsverarbeitung, 2006
- [Mor95] Morneburg, H.: Bildgebende Systeme für die medizinische Diagnostik. 3. Auflage. Publicis Corporate Publishing, 1995. ISBN 3–895–78002–2
- [MOS90] MULDER, W.; OSHER, S.; SETHIAN, J.A.: Computing Interface Motion in Compressible Gas Dynamics. In: Journal of Computational Physics 100 (1990), 209-228. http://www.ta.tudelft.nl/privatepages/w.a.mulder/Papers/Journal/JCP100_MOS.pdf
- [MS85] MUMFORD, D.; SHAH, J.: Boundary detection by minimizing functionals. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Springer, June 1985, S. 22–26
- [MS89] Mumford, D.; Shah, J.: Optimal approximation by piecewise smooth functions and associated variational blems. In: Communications on Pure Applied Mathematics 577-685. http://www.dam.brown.edu/people/ mumford/Papers/DigitizedVisionPapers--forNonCommercialUse/ 89-Mumford-Shah.pdf
- [MSV95] MALLADI, R.; SETHIAN, J.A.; VEMURI, B.: Shape modelling with front propagation: A level set approach. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 17 (1995), Nr. 2, 158–174. ftp://ftp.cis.ufl.edu/pub/faculty/vemuri/pami_fronts.ps
- [MT99] McInerney, T.; Terzopoulos, D.: Topology adaptive deformable Surfaces for medical image volume segmentation. In: *IE-EE Transactions on Medical Imaging* 18 (1999), October, Nr. 10, 840-850. http://reference.kfupm.edu.sa/content/t/o/topology_adaptive_deformable_surfaces_fo_321796.pdf
- [OF01] OSHER, S.; FEDKIW, R.P.: Level set methods: an overview and some recent results. http://www.ipam.ucla.edu/publications/gbm2001/gbmtut_sosher.pdf. Version: 2001
- [OF03] OSHER, S.; FEDKIW, R.: Applied Mathematical Sciences. Bd. 153: Level set methods and dynamic implicit surfaces. Springer-Verlag, 2003. ISBN 0-387-95482-1

LITERATURVERZEICHNIS

- [OP03] OSHER, S.; PARAGIOS, N.: Geometric level set method in imaging, vision, and graphics. Springer-Verlag, 2003. ISBN 0-387-95488-0
- [OS88] OSHER, S.; SETHIAN, J.A.: Fronts propagating with curvature dependent speed: Algorithm based on hamilton-jacobi formulation. In: Journal of Computational Physics 79 (1988), 12-49. http://math.berkeley.edu/~sethian/2006/Papers/sethian.osher.88.pdf
- [Sch07] SCHEUERMANN, Björn: Segmentierung medizinischer Bilddaten mit der Level-Set-Methode, Institut für Mensch-Maschine-Kommunikation, Leibniz Universität Hannover, Studienarbeit, September 2007
- [Set] SETHIAN, J.A.: Level set methods: an initial value formulation. http://math.berkeley.edu/~sethian/2006/Explanations/ level_set_explain.html
- [Set95] SETHIAN, J.A.: Theory, algorithms, and applications of level set methods for propagating interfaces. In: *Acta Numerica* (1995), August. http://math.berkeley.edu/~sethian/2006/Papers/sethian.actanumerica.1995.pdf
- [Set96] SETHIAN, J.A.: A fast marching level set method for monotonically advancing fronts. In: *Proc. Nat. Acad. Sci*, 1996, 1591—1595. http://math.berkeley.edu/~sethian/2006/Papers/sethian.fastmarching.pdf
- [Set98] SETHIAN, J. A.: Adaptive Fast Marching and Level Set Methods for Propagating Interfaces. In: *Acta Math. Univ. Comenianae Vol LXVII* 67 (1998), 3-15. http://math.berkeley.edu/~sethian/2006/Papers/sethian.algorithmy.pdf
- [Set99a] Sethian, J. A.: Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, 1999.
 ISBN 0-521-64557-3
- [Set99b] SETHIAN, J.A.: Fast marching methods. In: SIAM Review 41 (1999), 199-235. http://math.berkeley.edu/~sethian/2006/Papers/sethian.siam_fast.pdf
- [SJ92] SETHIAN, J.A.; J.STRAIN: Crystal growth and dendritic solidification. In: Journal of Computational Physics 98 (1992), 231-253. http://math.berkeley.edu/~sethian/2006/Papers/sethian.crystal.pdf

- [SL07] Song, J.-P.; Li, S.J.: An improved Mumford-Shah model and its applications to image processing with the piecewise constant level set method. In: *ACTA Automatica Sinica* 33 (2007), December, Nr. 12, 1259–1262. http://www.aas.net.cn/qikan/manage/wenzhang/071205.pdf
- [Tak] TAKEI, Ryo: Active contours without edges and image segmentation. http://www.math.ucla.edu/~rrtakei/gradProj/report922.pdf
- [TAYW01] TSAI, A.; A. YEZZI, Jr.; WILLSKY, A.S.: Curve evolution implementation of the Mumford-Shah Functional for Image Segmentation, Denoising, Interpolation, and Magnification. In: *IEEE Transactions on Image Processing* 10 (2001), August, Nr. 8, 1169-1186. http://ssg.mit.edu/group/alumni/atsai/Papers/TIP01.pdf
- [Wasa] WASILEWSKI, M.: Active contours using level sets for medical image segmentation. http://postulate.org/segmentation/segmentation.
- [Wasb] WASILEWSKI, M.: Homepage von Michael Wasilewski. http://www.postulate.org
- [WBMS01] WHITAKER, Ross; BREEN, David; MUSETH, Ken; SONI, Neha: Segmentation of biological Volume Datasets using a Level-Set framework, Volume Graphics 2001. In: In Proceedings of the International Workshop on Volume Graphics 2001, Springer, 2001, 249-263. ftp://ftp.gg.caltech.edu/pub/david/VG01_chapter.pdf
- [Wik] WIKIPEDIA: Wikipedia Die freie Enzyklopädie. http://wikipedia.org
- [Wol06] Wolter, Prof. Dr. F.: Skript der Lehrveranstaltung Geometrisches Modellieren. Leibniz Universität Hannover, Fachgebiet Graphische Datenverarbeitung, 2006
- [YKK⁺97] YEZZI, A.; KICHENASSAMY, S.; KUMAR, A.; OLVER, P.; TAN-NENBAUM, A.: A geometric snake model for segmentation of medical imagery. In: *IEEE Trans. on Medical Imaging* 16 (1997), April, Nr. 2, 199–209. http://www.math.umn.edu/~olver/vi_/medical.pdf