

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



Diplomarbeit

Entwurf gradientenabhängiger 2D- Transferfunktionen für die medizinische Volumenvisualisierung

Diana Stölzel

Angefertigt am Centrum für medizinische Diagnosesysteme und Visualisierung
(MeVis gGmbH) an der Universität Bremen

MeVis

www.mevis.de

Entwurf gradientenabhängiger 2D- Transferfunktionen für die medizinische Volumen- visualisierung

Diplomarbeit

an der Fakultät für Informatik
der Otto-von-Guericke-Universität

von: Diana Stölzel
Matr.-Nr.: 155439

1. Gutachter: Prof. Dr. Bernhard Preim (Otto-von-Guericke-Universität)
2. Gutachter: Dr. Volker Dicken (MeVis gGmbH)

Betreuer: Dr. Volker Dicken (MeVis gGmbH)
Prof. Dr. Bernhard Preim (Otto-von-Guericke-Universität)

Angefertigt am: Centrum für medizinische Diagnosesysteme und Visualisierung
(MeVis gGmbH) an der Universität Bremen

Zeit der Diplomarbeit: 01.04.2004-27.09.04

Selbstständigkeitserklärung

Hiermit versichere ich, Diana Stölzel, die vorliegende Arbeit allein und nur unter Verwendung der angegebenen Quellen angefertigt zu haben.

Diana Stölzel
Magdeburg, 27.09.04

Danksagung

Ich möchte mich hiermit recht herzlich bei meinem Betreuer Dr. Volker Dicken bedanken, der mich während dieser Arbeit ideenreich und intensiv beraten und unterstützt hat. Besonders bedanken möchte ich mich auch bei Prof. Dr. Bernhard Preim, der mir den Kontakt zu MeVis vermittelt hat und mich hervorragend fachlich unterstützte.

Dr. Hoen-Oh Shin und Benjamin King von der Medizinischen Hochschule möchte ich ebenfalls meinen Dank aussprechen. Ihre Tipps und Anregungen zu Beginn dieser Arbeit lieferten wertvolle Ideen.

An dieser Stelle muss auch Stephan Dachwitz (MeVis) erwähnt werden, ohne den die tollen Bilder dieser Arbeit nicht zustande gekommen wären und der stets Verbesserungen am Volumenrenderer vornahm.

Auch die Diskussionen mit Andreas Tappenbeck brachten stets neue Ideen zum Vorschein.

Überdies bedanke ich mich bei Wolf Spindler (MeVis) für seine programmiertechnische Unterstützung und MeVis dafür, dass verschiedene Module zur Bildbearbeitung und Visualisierung genutzt werden konnten.

Abschließend geht ein großer Dank an meine Eltern, die mir das Studium der Computervisualistik ermöglichten und mir immer mit Tat und Kraft zur Seite standen.

I Inhaltsverzeichnis

I	Inhaltsverzeichnis	I
II	Abbildungsverzeichnis	IV
III	Tabellenverzeichnis	VI
IV	Abkürzungsverzeichnis	VII
1	Einleitung	1
2	Medizinische Bildgebung und Visualisierung	3
2.1	Akquirierung medizinischer Volumendaten	3
2.1.1	CT.....	3
2.1.2	MRT	6
2.1.3	Rekonstruktion und Partialvolumeneffekte.....	8
2.1.4	Zusammenfassung.....	9
2.2	Visualisierung medizinischer Volumendaten.....	10
2.2.1	Schichtdatendarstellung	10
2.2.2	Indirektes Volumenrendering.....	11
2.2.3	Direktes Volumenrendering	12
2.3	Direktes Volumenrendering	12
2.3.1	Bildbasiertes Volumenrendering.....	12
2.3.2	Objektbasiertes Volumenrendering.....	13
2.3.3	ShearWarp Algorithmus.....	13
2.3.4	Texturbasiertes Volumenrendering	14
2.3.5	Beleuchtung.....	15
2.3.6	Von der TF zur Lookup-Tabelle	17
2.3.7	Klassifikation der TF.....	18
2.3.8	Der GigaVoxelRenderer	19
2.4	Zusammenfassung.....	19
3	Definition von Transferfunktionen	21
3.1	Eindimensionale Transferfunktionen	21
3.1.1	Interaktiv/Trial-and-Error.....	21
3.1.2	Bildbasiert	24
3.1.3	Weitere eindimensionale Transferfunktionen	26
3.1.4	Zusammenfassung und Vergleich	26
3.2	Mehrdimensionale Transferfunktionen	27
3.2.1	Das Kontur-Spektrum	27
3.2.2	Datenbasierte Transferfunktionen.....	28
3.2.3	Zusammenfassung und Vergleich	32
4	Entwurf gradientenabhängiger 2D-TFs	34
4.1	Bereitstellen von Gradienteninformationen	34
4.2	Generierung eines Intensitäts-Gradienten-Histogramms	35
4.3	Analyse von Intensitäts-Gradienten-Histogrammen	36
4.3.1	Strukturen in Histogrammen medizinischer Datensätze	38
4.3.2	Auswirkungen unterschiedlicher Rekonstruktionsfilter in CT-Daten.....	39
4.3.3	CT-Datensätze mit harten Rekonstruktionskernen	41
4.3.4	Auswertung von MRT-Daten.....	43

4.4	Approximation der Bogenverläufe durch Funktionen	45
4.4.1	Parametrische Kurven	46
4.4.2	Untersuchung einfacherer Funktionen	48
4.5	Entwurf einer Repräsentationsform kantendetektierender TFs.....	50
4.5.1	Repräsentation	50
4.5.2	Steuerung der Opazitätszuordnung	51
4.5.3	Steuerung der Grauwert-/Farbzuordnung.....	53
4.5.4	Behandlung sich überschneidender Bögen.....	55
4.5.5	Zusammenfassung	56
4.6	Interaktionskonzepte	56
4.6.1	Parametrierung der Repräsentation	56
4.6.2	Transparenzen und Farbwerte	58
4.6.3	Unterstützung der TF-Definition.....	58
4.7	Automatische Suche/Parametrierung der Bögen.....	63
4.7.1	Analyse vorhandener Suchalgorithmen für Ellipsen.....	63
4.7.2	Abtragen der Intensitäts- und Gradientenwerte auf Polarkoordinaten.....	64
4.7.3	Ein auf Apriori-Wissen basierender Algorithmus.....	65
4.7.4	Definition von Presets	69
4.8	Zusammenfassung	71
5	Realisierung	73
5.1	Programmierwerkzeuge	73
5.1.1	OpenInventor	73
5.1.2	Die Softwareplattform MeVisLab.....	73
5.1.3	SoView2D-Klasse	75
5.2	Generierung gradientenabhängiger 2D-TFs.....	76
5.2.1	Entwurf einer SoView2D-Klasse für Halbellipsen.....	77
5.2.2	Entwurf einer Listenklasse zur Speicherung der Halbellipsen.....	77
5.2.3	Berechnung der 2D-LUT	78
5.3	Werkzeuge zur vereinfachten Erkennung von Bögen.....	78
5.3.1	JointHistMask.....	79
5.3.2	Kantendetektion	79
5.3.3	Kantendetektion mithilfe einer Segmentierungsmaske.....	80
5.4	Entwurf eines Editors	81
5.5	Zusammenfassung	82
6	Ergebnisse.....	83
6.1	Für den 2D-TFs-Entwurf gut geeignete Datensätze.....	83
6.1.1	Zahndatensatz.....	83
6.1.2	Visible-Human-Kopf.....	84
6.1.3	Plexiglas-Phantomdatensatz	84
6.1.4	Ein Schweinedatensatz mit kontrastierter Leber	86
6.2	Visualisierung einer nicht-segmentierten kontrastierten Leber.....	87
6.2.1	Visualisierung der Gefäße und der Leberoberfläche.....	87
6.2.2	VR der Lebergefäße mit TFs und mit Segmentierungsmasken	88
6.3	Visualisierung der Lungenbronchien und -gefäße	89
6.4	Visualisierung der Knochen	93
6.5	Abdomendatensatz	94
6.6	MRT-Daten	95
7	Zusammenfassung und Ausblick.....	97

8	Literaturverzeichnis	99
A	Anhang	104
	GradLutEditorMacro	104

II Abbildungsverzeichnis

Abbildung 1-1: Vom 3D-Feld zur Volumenvisualisierung.....	1
Abbildung 2-1: Prinzip der Bildentstehung einer CT-Schicht	4
Abbildung 2-2: 3-Tesla-MRT von Philips	6
Abbildung 2-3: Unterschiedlich gewichtete MRT-Kopfaufnahmen.....	7
Abbildung 2-4: Rekonstruktionsfilter für eindimensionale Signale.....	8
Abbildung 2-5: Partialvolumeneffekte.....	9
Abbildung 2-6: Visualisierungspipeline	10
Abbildung 2-7: Schichtdarstellung eines MRT-Kopf-Datensatzes.....	11
Abbildung 2-8: Oberflächendarstellung einer Leber mit Gefäßen.....	12
Abbildung 2-9: Unterteilung des Volumens in Polygonschichten	14
Abbildung 2-10: Ausrichtung der Polygone	15
Abbildung 2-11: Intensitätsbild und zugehöriges Gradientenstärkebild	16
Abbildung 2-12: Isooberflächendarstellung unterschiedlich beleuchtet	17
Abbildung 2-13: Klassifikation der TF.....	18
Abbildung 3-1: Prinzip der Fensterung.....	23
Abbildung 3-2: Fensterung für den Knochenbereich.....	23
Abbildung 3-3: Histogramm hinterlegt als Kontextinformation.....	24
Abbildung 3-4: Design-Galerie.....	25
Abbildung 3-5: Verschiedene stochastisch erzeugte Ergebnisbilder für einen MRT-Kopf.....	25
Abbildung 3-6: Stetig lineare TF und der dazugehörige Datensatz mit Hervorhebung des Rückenmarks.....	26
Abbildung 3-7: Parameter einer Isokontur dargestellt als 2D-Funktionsgraph.....	28
Abbildung 3-8: Modell einer Kante	29
Abbildung 3-9: Intensitätsverlauf $f(x)$ und die zugehörige Ableitungen $f'(x)$ und $f''(x)$	29
Abbildung 3-10: Histogrammvolumen	30
Abbildung 3-11: Kugelförmiges Objekt und dessen Funktionen $g(v)$, $h(v)$, $p(v)$	30
Abbildung 3-12: Objekt mit verschiedenen Gewichtsfunktionen	31
Abbildung 3-13: Applikator und die korrespondierenden Punkte im Histogramm weiß dargestellt.....	31
Abbildung 3-14: Auf einem Histogramm basierender TF-Editor	32
Abbildung 4-1: Visualisierungspipeline für 2D-TFs	34
Abbildung 4-2: Kernelmasken des dreidimensionalen Sobelfilters	35
Abbildung 4-3: 2D-Histogrammgenerierung	36
Abbildung 4-4: 2D-Histogramm.....	36
Abbildung 4-5: Bedeutung der Strukturen.....	37
Abbildung 4-6: Zusammenhang zwischen Intensitäten und Gradientenstärken an einer Kante	37
Abbildung 4-7: Bögen in einem nicht-kontrastiertem Abdomenhistogramm.....	39
Abbildung 4-8: Scharfe Kante und das entsprechende Histogramm.....	40
Abbildung 4-9: Histogramm eines CT-Abdomendatensatzes.....	40
Abbildung 4-10: Histogramm eines CT-Abdomendatensatzes.....	40
Abbildung 4-11: Auswirkungen von Filterungen eines CT-Datensatzes auf das Histogramm.....	42
Abbildung 4-12: Histogramm basierend auf einem realen T1-MRT-Kopfdatensatz.....	43
Abbildung 4-13: Histogramm basierend auf einem realen T2-MRT-Kopfdatensatz:.....	43
Abbildung 4-14: Histogramm basierend auf einem realen geglätteten T1-MRT-Kopfdatensatz:	44
Abbildung 4-15: Histogramm basierend auf einem Phantom MRT-Kopf-Datensatz:.....	44
Abbildung 4-16: Phantomdatensatz mit Inhomogenität.....	45
Abbildung 4-17: einfache 2D-Repräsentation	45
Abbildung 4-18: komplexe Repräsentation	46
Abbildung 4-19: Segmentierung des Luft-Lungengefäße-repräsentierenden Bogens	48
Abbildung 4-20: Segmentierung des Fett-Knochen-repräsentierenden Bogens	49
Abbildung 4-21: Segmentierung des die untere äußere Zahnwurzel repräsentierenden Bogens	49
Abbildung 4-22: Ein Datensatz und die Approximation durch einfache Funktionen	50
Abbildung 4-23: Halbelliptische Bogenflächen.....	51
Abbildung 4-24: Verteilung der Transparenzen	52
Abbildung 4-25: Auswirkungen verschiedener Anwendungsbereiche am Beispiel eines Zahns	52
Abbildung 4-26: Farbverteilungen.....	53
Abbildung 4-27: Farbverteilungen am Beispiel des VRs der Wirbelsäule	53
Abbildung 4-28: Beleuchtung von medizinischen Daten	54
Abbildung 4-29: Mögliche Varianten bei Repräsentationsüberschneidungen.....	55

Abbildung 4-30: Prinzip des Rubberbandings für HE	57
Abbildung 4-31: HE definiert durch zwei Punkte „□“ und zwei Abstände	57
Abbildung 4-32: Interaktion mittels Rampen am Beispiel des Zahndatensatzes	58
Abbildung 4-33: LUT angewendet auf die Schichtdatendarstellung	59
Abbildung 4-34: Selektion eines Voxels und Anzeige des korrespondierenden Bins im Histogramm	59
Abbildung 4-35: Unterstützung der Kantensuche	60
Abbildung 4-36: Mögliche Bereiche zur Untersuchung von Kanten in der Leber	61
Abbildung 4-37: Histogramm eines Leberdatensatzes und gefundener Gefäßbogen	61
Abbildung 4-38: Repräsentationsform entlang des ermittelten Bogens	61
Abbildung 4-39: Gefäßvisualisierung im VR mit einer gradientenabhängigen TF	61
Abbildung 4-40: TFs-Definitionsprozess anhand einer Kantenmaske	62
Abbildung 4-41: Ellipsoidendetektion nach Kim	64
Abbildung 4-42: Ein stark verrauschtes Histogramm	64
Abbildung 4-43: Gradientenstärke und Grauwerte als Polarkoordinaten abgetragen	65
Abbildung 4-44: Gefundene Repräsentation und erzeugte LUT	67
Abbildung 4-45: Automatisch generierte TFs für einen Zahn	67
Abbildung 4-46: Histogramme mit unterschiedlicher Binsize	68
Abbildung 4-47: Automatisch gefundener Bogen in medizinischem Datensatz	69
Abbildung 4-48: Histogramme eines Datensatzes	71
Abbildung 5-1: Die Softwareplattform MeVisLab mit einem Beispielnetzwerk	74
Abbildung 5-2: Panel des <i>ImgLoad</i> -Moduls	74
Abbildung 5-3: Bildverarbeitungsnetzwerk mit unterschiedlichen Ein- und Ausgängen von Modulen	75
Abbildung 5-4: 3D-Koordinatentransformationen	76
Abbildung 5-5: Eine halbelliptische Bogenfläche	77
Abbildung 5-6: Aufbau des Makromoduls zur Kantendetektion	79
Abbildung 5-7: Makromodul für die Kantendetektion	80
Abbildung 5-8: Aufbau des Makros zur Kantendetektion einer Segmentierungsmaske	81
Abbildung 5-9: Beispielnetzwerk für die Erstellung einer LUT	81
Abbildung 6-1: Zahn im Vergleich	83
Abbildung 6-2: VR des Visible-Human-Kopfes mit zweidimensionaler TF	84
Abbildung 6-3: Photographie der Plexiglasscheibe	85
Abbildung 6-4: Die Plexiglasscheibe mit intensitätsbasierter und gradientenbasierter 2D-TF	85
Abbildung 6-5: Lebergefäßvisualisierung eines Schweinedatensatzes	86
Abbildung 6-6: VR einer kontrastierten Leber mit Darstellung der Lebergefäße	87
Abbildung 6-7: Histogramm des Leberdatensatzes	87
Abbildung 6-8: VR von Lebergefäßen	88
Abbildung 6-9: Lebergefäße mit Kontextinformation.	89
Abbildung 6-10: Histogramm mit TF	90
Abbildung 6-11: Schichtdarstellung des Datensatzes mit Färbung der von der TF dargestellten Voxel	90
Abbildung 6-12: VR der Bronchien mit intensitätsbasierter TF	90
Abbildung 6-13: VR der Bronchien mit der in Abbildung 6-10 definierten gradientenabhängigen TF	90
Abbildung 6-14: Definition einer TF für das VR großer und kleiner Bronchien	91
Abbildung 6-15: Histogramm des weich-rekonstruierten Lungendatensatzes	92
Abbildung 6-16: Lungenvisualisierungen	93
Abbildung 6-17: Knochenvisualisierung mittels gradientenabhängiger TFs	94
Abbildung 6-18: VR eines Abdomendatensatzes mittels gradientenbasierter TFs	95
Abbildung 6-19: VR eines realen medizinischen Kopf-Datensatzes	96
Abbildung 6-20: VR eines Phantom-MRT-Kopfdatensatzes	96
Abbildung 7-1: Vom skalaren 3D-Feld zum VR	97

III Tabellenverzeichnis

Tabelle 2-1: HU-Werte für verschiedene Gewebetypen	4
Tabelle 3-1: Verbreitete Konzepte für die Definition elementarer linearer Funktionen	22
Tabelle 4-1: Nachbarschaftliche Beziehungen zwischen einzelnen Geweben.....	38
Tabelle 4-2: Mathematische Beschreibung einfacher Funktionen	48

IV Abkürzungsverzeichnis

1D	eindimensional
2D	zweidimensional
3D	dreidimensional
CT	Computertomographie/Computertomogramm
HU	Hounsfield-Unit
LUT	Lookup-Tabelle
MeVis	Centrum für Medizinische Diagnosesysteme und Visualisierung an der Universität Bremen
MRT	Magnetresonanztomographie
TF	Transferfunktion
GVR	Gigavoxelrenderere, Volumenrenderere
VR	Volumenrenderere
HE	Halbellipse/Halbellipsen

1 Einleitung

Tomographische Verfahren der medizinischen Bildgebung wie die Computertomographie (CT) und die Magnetresonanztomographie (MRT) liefern Bilddatensätze des Menschen, die als 3D-Feld von skalaren Werten aufgefasst werden. Von besonderem Interesse für Mediziner ist das Erkennen und Darstellen anatomischer Strukturen aus diesen Daten. In der klinischen Praxis erfolgt die Auswertung der Daten normalerweise auf 2D-Schichtbildern, wobei Radiologen für die Diagnose eine 3D-Rekonstruktion der Daten mental ableiten.

Aufgrund der Weiterentwicklung der Medizintechnik ist es möglich, genauere und detailliertere Daten zu gewinnen, jedoch ist dies auch mit einem Wachsen der Datenmengen, mit denen Ärzte konfrontiert werden, verbunden. Eine Diagnose erfordert das Betrachten von bis zu mehreren hundert Einzelbildern auf Film. Diese Methode ist nicht nur unwirtschaftlich (hoher Verbrauch an Filmmaterial und großer Zeitaufwand), sondern auch ermüdend und fehlerbehaftet. Daher verlagert sich die Diagnostik auf den Computer. Spezifische Programme ermöglichen verschiedene Visualisierungen wie beispielsweise die Schichtdarstellung oder das Volumenrendering. Das Volumenrendering ist gut geeignet, um anatomische Strukturen im Überblick darzustellen. Dabei ist es möglich, den kompletten 3D-Datensatz zu visualisieren, was eine gute räumliche Orientierung erlaubt. Beim direkten Volumenrendering werden mithilfe einer geeigneten Transferfunktion (TF) unbedeutende Bereiche transparent und die anatomisch wichtigen Strukturen opak (sichtbar) dargestellt. Bisher existieren viele Verfahren zum Entwurf von 1D-TFs, aber nur wenige zur Entwicklung mehrdimensionaler TFs. Die Bedienung mehrdimensionaler ist oft schwer und der Benutzer benötigt fachspezifisches Vorwissen.

Ziel dieser Arbeit ist es, kantendetektierende 2D-TFs zu entwickeln, die dem Benutzer eine einfache Bedienung ermöglichen. Mit wenigen Handgriffen soll eine TF generiert werden und der Benutzer erste Ergebnisse sehen können.

Um diese TFs entwickeln und testen zu können, werden die Bibliotheken von MeVisLab verwendet. MeVisLab ist eine Bildverarbeitungsbibliothek, die speziell für die medizinische Bildverarbeitung entwickelt wurde. Für das direkte Volumenrendering steht in MeVisLab das Volumenrendering-Modul GVR (Gigavoxelrenderer) zur Verfügung, welches die Verwendung 2D-TFs unterstützt.

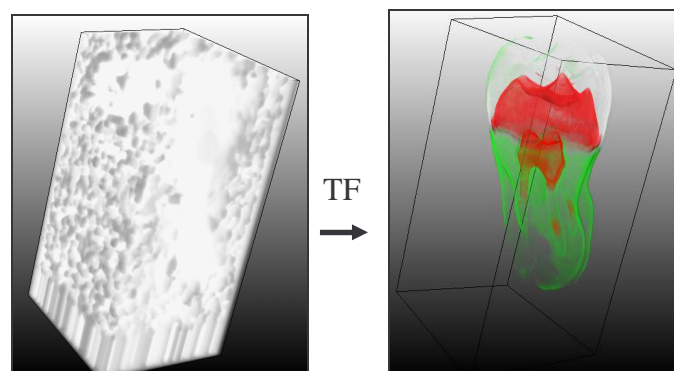


Abbildung 1-1: Vom 3D-Feld zur Volumenvisualisierung

Zusammenfassend ist folgendes Ziel zu nennen:

Entwicklung gradientenabhängiger zweidimensionaler TFs für das Volumenrendering medizinischer Bilddaten

Die vorliegende Arbeit gliedert sich wie folgt:

Kapitel 2 In diesem Kapitel werden Verfahren zur medizinischen Bildgewinnung, die Computertomographie (CT) und die Magnetresonanztomographie (MRT) beschrieben. Im Anschluss werden Volumenvisualisierungsmethoden vorgestellt. Insbesondere wird auf das direkte Volumenrendering eingegangen, das die Grundlage der in dieser Arbeit zu entwickelnden TFs bildet.

Kapitel 3 Hier wird ein Überblick über verschiedene TFs für das Volumenrendering gegeben, beginnend bei einfachen eindimensionalen bis hin zu komplexen mehrdimensionalen TFs.

Kapitel 4 Dieses Kapitel befasst sich mit dem Entwurf 2D-gradientenabhängiger TFs. Unter anderem wird beschrieben, welche Parameter zu einer „guten“ Visualisierung führen. Außerdem werden Möglichkeiten zur Automatisierung dieser TFs vorgestellt.

Kapitel 5 Die Realisierung und Implementierung der vorgestellten TFs wird in diesem Kapitel beschrieben. Dabei wird erläutert, welche Bibliotheken zur Implementierung genutzt wurden.

Kapitel 6 Schließlich werden Ergebnisse vorgestellt, wobei Anwendungsmöglichkeiten vorgestellt und Schwierigkeiten diskutiert werden.

Kapitel 7 Abschließend erfolgt eine Zusammenfassung über das in dieser Arbeit vorgestellte Konzept zum Entwurf 2D-TFs und eine Diskussion über mögliche Erweiterungen.

2 Medizinische Bildgebung und Visualisierung

Die Erforschung der menschlichen Anatomie war bis um 1900 auf das Aufschneiden und Sezieren des menschlichen Körpers beschränkt. Erst durch die Entwicklung bildgebender Verfahren wie Röntgen, CT oder MRT wurde es möglich, in das Innere des menschlichen Körpers zu sehen, ohne ihm größeren Schaden zuzufügen. Dieses Kapitel gibt eine Einführung zu den tomographischen Verfahren, ihren Anwendungsgebieten und ihren Einschränkungen. In [Mor95] können technische Details nachgelesen werden. Im Anschluss daran werden Visualisierungstechniken vorgestellt, die die Darstellung und Interpretation der gewonnenen tomographischen Daten ermöglichen.

2.1 Akquirierung medizinischer Volumendaten

In der Medizin gibt es je nach Visualisierungsziel eine Vielzahl von bildgebenden Verfahren. Mit Hilfe von Mikroskopen ist die Darstellung von kleinsten Zellen im menschlichen Körper möglich. Für die hier vorliegende Arbeit sind jedoch nur die tomographischen Verfahren von Bedeutung, da diese einen Volumendatensatz erzeugen und darauf ein Volumenrendering generiert werden kann. In dieser Arbeit wird der Schwerpunkt auf die Computertomographie und die Magnetresonanztomographie gelegt. Diese beiden Verfahren erzeugen einen skalaren 3D-Datenblock mit hoher räumlicher Auflösung und einem gutem Signal-Rausch-Verhältnis, welches die Grundlagen für eine gute 3D-Visualisierung bilden. Ziel einer 3D-Visualisierung ist die Unterstützung der Mediziner bei der Diagnosefindung, der Therapieplanung und beim Therapiemonitoring.

2.1.1 CT

Die CT-Aufnahme basiert auf Röntgenstrahlung, wobei unterschiedliche Abschwächungen von Röntgenstrahlungen dargestellt werden. Eine Röntgenquelle wirft Strahlen aus. Diese treffen auf die aufzunehmenden Körperbereiche. Dabei durchdringenden die Strahlen verschiedene Gewebearten. Ein Teil der Strahlung wird vom Gewebe absorbiert. Auf der gegenüberliegenden Seite der Röntgenquelle befindet sich ein Messgerät, das die ankommende Strahlung misst und anhand dessen das Bild generiert wird. Knochen haben eine hohe Dichte und damit eine hohe Absorbtiionsrate, Fett eine deutlich geringere.

Gleiche Gewebe haben in unterschiedlichen Aufnahmen die gleiche spezifische Absorbtiionsrate. Dies vereinfacht die Bildgenerierung wesentlich. Ein Nachteil des traditionellen Röntgenverfahrens liegt darin, dass durch die Projektion auf 2D, keine Räumlichkeit dargestellt werden kann. In dem CT-Verfahren liegt die Lösung für dieses Problem.

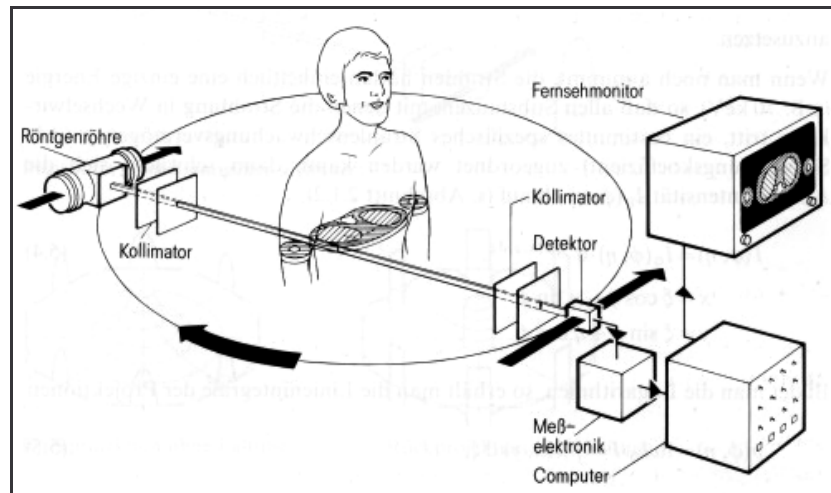


Abbildung 2-1: Prinzip der Bildentstehung einer CT-Schicht. Quelle: [Mor95]

Das CT-Verfahren wurde 1971 von Sir Godfrey N. Hounsfield eingeführt. Das Ziel war eine möglichst überlagerungsfreie Darstellung der Körperregion im Querschnitt, um so die Mehrdeutigkeiten der konventionellen Röntgenbilder zu vermeiden. In Abbildung 2-1 ist das Aufnahmeprinzip dargestellt. Die Röntgenröhre rotiert um die Längsachse der aufzunehmenden Person, währenddessen messen Detektoren auf der gegenüberliegenden Seite die ankommende Strahlung. Die gemessenen Strahlungsintensitäten werden durch elektrische Impulse an einen Computer weitergegeben. Aus den Strahlungsintensitäten wird mittels inverser Radontransformation (benannt nach Johan Radon, der 1917 die Radontransformation präsentierte) eine Schicht rekonstruiert. Jedem Volumenelement in der Schicht wird ein Hounsfieldwert (HU) zugeordnet, welcher sich aus der Dichte μ des Gewebes und der Dichte des Wassers berechnen lässt.

$$HU = \frac{\mu - \mu_{\text{Wasser}}}{\mu_{\text{Wasser}}} * 1000 \quad (2-1)$$

Knochen besitzen einen HU-Wert zwischen 100-3000, Wasser einen Wert von 0 und Luft hat Werte um die -1000. In Tabelle 2-1 sind die Wertebereiche für verschiedene Gewebearten zusammengefasst. [Mor95]

Tabelle 2-1: HU-Werte für verschiedene Gewebetypen

Gewebe	HU-Wert
Luft	-1000
Lunge	-600 ... -700
Fett	-100 ... -200
Wasser	0
Zysten	10 ... 20
Leber	50 ... 60
Nieren	50 ... 70
Knochen	100...3000

Die Kontrastierung von Gefäßen ist für Radiologen von hoher Relevanz. Dabei können intraluminale Pathologien, d.h. krankhafte Veränderungen wie Verschlüsse, Dissektionen oder Rupturen erkannt werden. Bei vielen Untersuchungen ist eine Kontrastierung zur Abgrenzung von Lymphknoten gegen umgebende Gefäße notwendig. Zur Durchführung einer CT-Angiographie (CTA) wird Kontrastmittel intravenös injiziert. Der Radiologe muss die Verteilung des Kontrastmittels im Körper beachten, um sicherzugehen, dass das Kontrastmittel während der Aufnahme in den Gefäßen ist, die für den Arzt zur Diagnostizierung wichtig sind. [Web02]

Vorteile

- Ein Vorteil des CT-Verfahrens liegt in der Normierung der Dichtewerte. Dies erleichtert die Wiedererkennung von Strukturen und die Definition von Transferfunktionen.
- Die kurze Aufnahmezeit ist ein weiterer wesentlicher Vorteil von CT-Aufnahmen. Durch schnelle Aufnahmen entstehen nur geringe Bewegungsartefakte.
- Außerdem erzeugt das CT-Verfahren Datensätze mit einer guten Ortsauflösung.

Nachteile

- Nachteilig bei diesem Verfahren ist, dass der Patient einer Strahlung ausgesetzt wird, welche für den menschlichen Körper in zu hoher Dosis schädlich sein kann. Deshalb dürfen Menschen nur eine begrenzte Zeit der Röntgenstrahlung ausgesetzt werden, was mitunter die Qualität der Bilder beeinflussen kann.
- Beim CT wird zwischen harter und weicher Strahlung unterschieden, wobei harte Strahlung gut für die Knochendarstellung und weiche gut für die Weichteildarstellung (Organe und weibliche Brust) geeignet ist. Die Weichteile sind die Regionen des menschlichen Körpers, in denen unter anderem oft Krebs auftritt und die daraufhin untersucht werden. Jedoch ist gerade die Erzeugung dieser Bilder ein Problem, da weiche Strahlung von den Geweben besser absorbiert wird. Dadurch entstehen bessere Bilder, was aber gleichzeitig bedeutet, dass diese Aufnahmen für den menschlichen Körper schädlicher sind.
- Gefäße lassen sich oft nur durch Kontrastmittelgabe darstellen.
- Ein weiterer Nachteil bei CT-Aufnahmen ist die Artefaktbildung an metallischen Gegenständen. So lassen sich die metallischen Implantate eines Patienten mittels CT nicht gut untersuchen.

Eigenschaften der generierten Datensätze

Die von einem CT generierten 3D-Datensätze liegen in einem regelmäßigem 3D-Gitter vor. Ein Element dieses Gitters wird als Voxel bezeichnet. Es liegen meist anisotrope Datensätze vor, das bedeutet, dass der Schichtabstand größer ist als der Abstand der Pixel einer Schicht zueinander. Typischerweise liegt der Pixelabstand bei 0.7 mm, während der Abstand der Schichten zueinander 2 mm beträgt. Ein Multislice-CTs kann die Schichten mit kleineren Abständen auflösen, so ist ein Schichtabstand von 1 mm und kleiner möglich. Die Auflösung innerhalb einer Schicht beträgt meist 512×512 Voxel. Die Qualität der gewonnenen Datensätze (viel/wenig Rauschen oder weiche/harte Kanten) hängt von der Abtastrate und von den Einstellungen des Radiologen zur Rekonstruktion des Datensatzes ab, siehe Abschnitt 2.1.3.

2.1.2 MRT

Bei der Magnet-Resonanz-Tomographie werden unter Ausnutzung der Wechselwirkungen zwischen einem magnetischen Feld und dem menschlichen Körper Bilder erzeugt.

Das Verfahren basiert auf den magnetischen Eigenschaften der Protonen des Wasserstoff-Atomkerns. Protonen haben einen Eigendrehimpuls (Spin). Diese Eigenrotation des Spins ist nicht sichtbar, jedoch kann die Lage der Rotationsachse des Spins gemessen werden. Mit Hilfe eines starken statischen Magnetfeldes werden die Spins des Wassers des zu untersuchenden Menschen, beeinflusst. Mit Hilfe eines über HF-Spulen (HF=high frequent) eingestrahnten gepulsten HF-Magnetfeldes, werden die Spins angeregt. Beim Abschalten des HF-Magnetfeldes geben sie kleine magnetische Signale ab, die mit Hilfe spezieller Empfänger-spulen gemessen werden. Über zeitweilig zugeschaltete Gradienten-Magnetfelder können diese Signale bestimmten Orten (präziser Voxeln) des Patienten zugeordnet werden. Mit Hilfe leistungsstarker Rechner wird aus diesen zahlreichen Signalen dann das gewünschte Schnittbild erzeugt [KM94], [MWW04].



Abbildung 2-2: 3-Tesla-MRT von Philips. Quelle: [Rad04]

Vorteile

Vorteile dieser Methode liegen darin, dass beim MRT die Weichteile gut dargestellt werden können und dass keine für den Menschen gefährliche Strahlung auftritt.

Nachteile

Je nach Messung einzelner Zeiten der Ausrichtung der Atome können zwei unterschiedliche Arten von Bildern entstehen. Der Arzt kann bei diesen Aufnahmen zwei verschiedene Zeiten messen: zum einen wie lange die Atomkerne brauchen, um sich parallel zum angelegten Magnetfeld auszurichten und zum anderen wie lange es dauert, bis die Atomkerne sich senkrecht zum Magnetfeld ausgerichtet haben. Anhand dieser Messungen werden jeweils unterschiedliche Bilddaten generiert, siehe Abbildung 2-3. Hier wird von T1- und T2-gewichteten Bildern gesprochen. Um ein MRT-Bild zu generieren, werden die Atomkerne einer Schicht mehrmals hintereinander angeregt. Die Zeit, die zwischen zwei aufeinander folgenden Anre-

gungen vergeht, wird als Repititionszeit bezeichnet. [KM94] Die Echozeit ist die Zeit, die vergeht bis das MRT-Signal gemessen wird. Die T1-Gewichtung ist gekennzeichnet durch eine kurze Repititionszeit und eine kurz Echozeit, bei der T2-Gewichtung sind beide Zeiten lang. Bei einer T1-Gewichtung wird Fett hell bei einer T2-Gewichtung wird jedoch Wasser hell dargestellt. Wegen dieser unterschiedlichen Aufnahmebedingungen werden oft mehrere Bilddaten erzeugt, so dass der Radiologe die Bilder untereinander vergleichen kann.

Aufgrund der vielen Parameter und der unterschiedlichen Intensitätsberechnung der Bilder ist die Auswertung von MRT-Daten und auch die Definition von TFs aufwändiger als bei der CT. Der Radiologe benötigt dafür eine größere Erfahrung. Für überweisende Ärzte sind diese Daten oft kaum interpretierbar, da die Intensitäten nicht direkt einzelnen Geweben zugeordnet werden können.

Nachteilig bei diesem Verfahren sind auftretende Inhomogenitäten. Das Magnetfeld besitzt nicht im gesamten Scannbereich die gleiche Stärke, was zu unterschiedlichen Intensitäten in dem Bild führen kann, d.h. wird ein Gewebe in einem Teil des Körpers hell dargestellt, so kann es sein, dass es in einem anderen Teil etwas dunkler dargestellt wird. Die Intensität ist in einigen Bereichen des rekonstruierten Datensatzes inhomogen. Dadurch wird das Definieren von TFs erschwert, da keine konstanten Werte vorliegen.

Auch bei diesem Verfahren treten Probleme bei Patienten mit metallischen Implantaten auf. Ein Patient mit beispielsweise einem Herzschrittmacher oder mit Metallprothesen kann nicht von einem MRT abgetastet werden, da Metall zu starke magnetische Eigenschaften aufweist.

Außerdem ist dieses Verfahren gerätetechnisch aufwändig und auch die Anschaffungskosten sind hoch. Das Gerät ist groß und die Räumlichkeiten müssen wegen der starken Magnetisierung zur Isolierung und zur Abschirmung von äußeren Störfaktoren mit Kupferplatten ausgelegt werden.

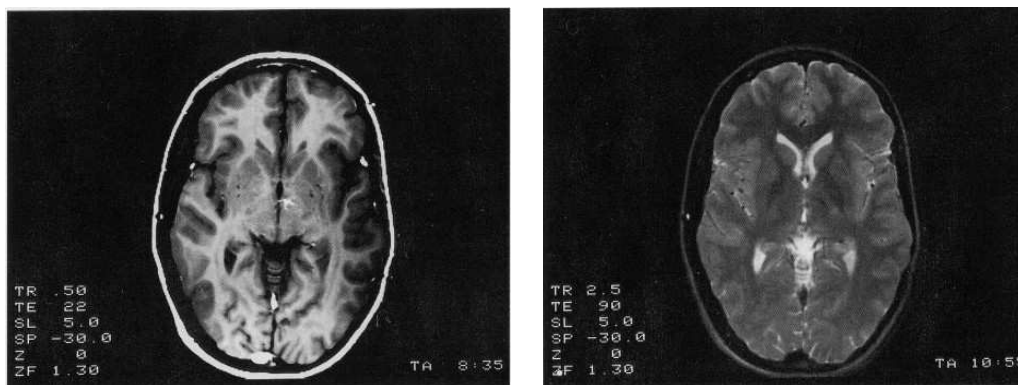


Abbildung 2-3: Unterschiedlich gewichtete MRT-Kopfaufnahmen.

Links: T1-Gewichtung mit einer Repititionszeit von 500 ms und einer Echozeit von 22 ms. Rechts: T2-Gewichtung mit einer Repititionszeit v. 2500 ms und einer Echozeit von 90 ms Quelle: [Morneborg95]

Eigenschaften der generierten Datensätze

Die vom MRT generierten 3D-Datensätze liegen ähnlich wie beim CT in einem regelmäßigen 3D-Gitter vor. Der Pixelabstand in einer Schicht beträgt dabei typischerweise 1.4 mm und der Schichtabstand 4 mm. Beim MRT besitzen die Datensätze meist eine Auflösung von 256×256 Voxel pro Schicht mit ca. 50-80 generierten Schichten.

2.1.3 Rekonstruktion und Partialvolumeneffekte

Um einen 3D-Volumendatensatz zu erzeugen, wird ein kontinuierliches Signal an diskreten Punkten abgetastet. Die registrierten Absorptionsraten werden normiert und analog/digital gewandelt an den Rechner übertragen. Der Rechner berechnet aus den Rohdaten mit Hilfe aufwändiger Algorithmen (Fouriertransformation, Faltung und Rückprojektion) die Intensitätswerte (CT-Werte, MRT-Werte) der einzelnen Bildpunkte (Bilddaten, Pixel bzw. „Voxel“). Dieser Schritt wird als Rekonstruktion der Daten bezeichnet. Um eine möglichst genaue Rekonstruktion eines Signals zu erreichen, müsste laut dem Abtasttheorem [Fol90] eine sinc-Funktion verwendet werden.

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \quad (2-1)$$

Um Gleichung (2-1) anzuwenden, müssten alle Abtastpunkte eines Signals in die Berechnung mit einfließen. Dies ist aber nur schwer berechenbar, daher wird entweder durch die Berechnung der nächstliegenden Nachbarn, durch trilineare Interpolation oder durch Verwendung einfacher Rekonstruktionsfilter wie der Box- oder der Dachfilter (entspricht trilinear Interpolation), der Wert an der gesuchten Position berechnet [Rez01], siehe Abbildung 2-4.

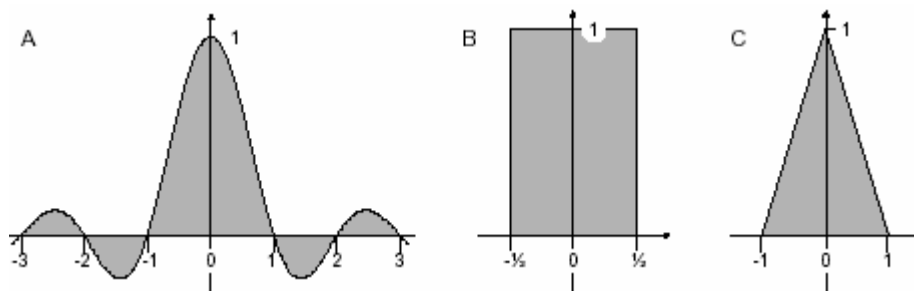


Abbildung 2-4: Rekonstruktionsfilter für eindimensionale Signale: idealer sinc-Filter(A), Boxfilter (B), Dachfilter (C). Quelle: [Rez01]

Auch bei der Rekonstruktion von tomographischen Daten werden Rekonstruktionsfilter verwendet. Ein Patient wird durch einen Tomographen abgetastet und der Radiologe stellt verschiedene Parameter des Rekonstruktionsfilters ein. Dabei stehen ihm verschiedene Einstellungsmöglichkeiten zur Verfügung. Er kann zwischen weicher (Glättung der Kanten), harter (Kantenbetonung) und mittlerer Rekonstruktion wählen. Mit der harten Rekonstruktion erzeugt der Radiologe scharfe Kanten im Bild. Ein Nachteil ist jedoch, dass in hart-rekonstruierten Bildern starkes Rauschen auftritt. Weiche Rekonstruktionen glätten das Rauschen jedoch glätten sie auch die Kanten.

Die Einstellung des Rekonstruktionsfilters ist abhängig vom Visualisierungsziel. So wird weich-rekonstruiert, wenn Niedrigkontrastbereiche begutachtet werden. Wenn starkes Rauschen in diesen Bildern wäre, könnten diese Bereiche nicht voneinander unterschieden werden.

Die Betrachtung hart-rekonstruierter Datensätze erfolgt, wenn kleinste Verzweigungen und Strukturen erkannt werden sollen. Dies wird z.B. oft bei Lungenaufnahmen angewendet, um die kleinen Verästelungen der Lungegefäße zu betrachten oder um Staubablagerungen in der Lunge zu erkennen. Auch bei Knochenfrakturen insbesondere bei Haarrissen werden harte Filter benutzt.

Die Einstellungen des Rekonstruktionsfilters dürfen bei der Definition von gradientenabhängigen TFs nicht außer Acht gelassen werden. Zu untersuchen ist dabei, welchen Einfluss die Filter auf die Volumenvisualisierung nehmen. Lassen sich beispielsweise hart-rekonstruierte Datensätze mit der gleichen Qualität wie weich-rekonstruierte darstellen.

Partialvolumeneffekt

Ein Phänomen, das bei tomographischen Aufnahmeverfahren auftritt, ist der Partialvolumeneffekt. Dieser entsteht erstens durch die Diskretisierung der Daten in Voxel und zweitens durch die Einstellungen des Rekonstruktionsfilters (durch den Radiologen). Die Genauigkeit einer tomographischen Aufnahme ist u.a. abhängig von der endlichen Strahlenbreite eines Abtaststrahls. In der Natur gibt es eine klare Grenze zwischen zwei Materialien, siehe Abbildung 2-5(links). Beim Röntgen eines Objektes kann der Röntgenstrahl zwei Materialien gleichzeitig treffen, d.h. es werden zwei Materialien auf ein Voxel abgebildet. Das Voxel wird von zwei Materialien geteilt, siehe Abbildung 2-5(rechts). Die Intensität des Voxels ergibt sich aus dem Verhältnis und den Intensitäten der beiden beteiligten Materialien.

Der zweite Aspekt, die Rekonstruktion der Bilddaten, wirkt sich auf die Partialvolumeneffekte in der Art aus, dass bei einem weichen Filter der Partialvolumeneffekt noch vergrößert wird. Durch das Glätten der Kante wird der Übergangsbereich zwischen zwei Materialien noch größer. Bei einem harten Filter ist dieser Effekt weniger stark. Dies ist ein wesentlicher Punkt, der bei der Generierung von TFs beachtet werden muss.

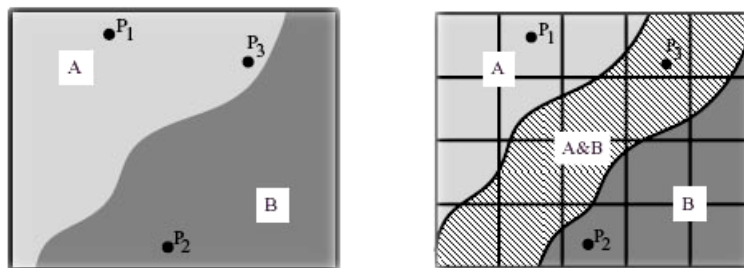


Abbildung 2-5: Partialvolumeneffekte
reale Kante(links), abgetastete Kante(rechts). Quelle:[LFB98]

2.1.4 Zusammenfassung

In diesem Abschnitt wurden zwei tomographische Verfahren vorgestellt, das CT-Verfahren und die MRT. CT findet vor allem bei Knochen- und Thoraxaufnahmen Anwendung, jedoch ist sie auch bei Gefäßaufnahmen, wobei die Gefäße mit Kontrastmittel angereichert sind, geeignet und ein weit verbreitetes Verfahren.

Ein Nachteil der CT liegt darin, dass in den Weichteilen nur ein geringer Intensitätsunterschied erzeugt und somit in den Bildern nur ein schwacher Weichteilkontrast erreicht werden kann. Hier liegt eine der Stärken des MRT. Die Weichteile können gut abgebildet werden. Die MRT ist besonders gut für Gehirnaufnahmen zur Diagnosefindung geeignet, z.B. um dort Tumore zu diagnostizieren, aber auch das Herz, (Hand-, Knie-) Gelenke, Bänder, Sehnen und Knorpel können im MRT gut dargestellt werden.

Bei beiden Verfahren treten Partialvolumeneffekte auf, die die Qualität der Aufnahme beeinflussen. Sie müssen bei der Definition von TFs berücksichtigt werden.

2.2 Visualisierung medizinischer Volumendaten

Die Rohdaten für die 3D-Visualisierung werden von tomographischen Verfahren generiert. Im nächsten Schritt sollen die Daten visualisiert werden. Um von den Rohdaten eine geeignete Ansicht zu erzeugen, muss ein Prozess gestartet werden, der in der Visualisierung als Visualisierungspipeline bezeichnet wird.

Die Visualisierungspipeline für die 3D-Visualisierung, in Abbildung 2-6 dargestellt, beinhaltet drei Stufen:

Eingangsdaten sind Rohdaten aus dem Tomographen. In der Realität liegt eine extrem große Menge an Daten vor, jedoch ist nur ein Teil für die weitere Bearbeitung nötig. Der Datensatz wird auf die gewünschte Größe reduziert. In der ersten Stufe erfolgt eine weitere Filterung der Daten. Hauptaufgabe ist ein erneutes Abtasten (Resampling) und Interpolieren der Daten auf die gewünschte interne Repräsentation oder auf die vom Benutzer spezifizierten Auswahlkriterien. Die nächste Stufe „Abbildung“ ist der Kern der Visualisierungspipeline. Hier werden Strukturen aus den Daten ermittelt. Es wird festgelegt, welche Informationen aus den Daten extrahiert und visualisiert werden sollen. Dafür müssen Visualisierungseigenschaften für die Daten bestimmt werden, d.h. es muss eine Abbildung aus Graphikprimitive (Punkte, Linien, Dreiecke) erfolgen und Visualisierungsparameter (Größe, Farbe, Transparenz) sind festzulegen. Die letzte Stufe, das Rendering, erzeugt aus den Graphikprimitiven das Ergebnisbild.



Abbildung 2-6: Visualisierungspipeline

Die 3D-Visualisierung gewinnt heutzutage an immer größerer Bedeutung. In der Medizin findet sie Anwendung in vielen Bereichen. So kann die 3D-Visualisierung in der Lehre und dem Training von Ärzten hilfreich sein. Beitragen kann sie zur Diagnosenfindung, so z.B. bei Erkrankungen von Gefäßen oder bei der Erkennung von Tumoren. Medizinische Volumenvisualisierung kann aber auch in der Therapieplanung, im Bereich der Bestrahlungs- und der Operationsplanung, eingesetzt werden.

In den nächsten Abschnitten werden drei Methoden zur Darstellung von Volumendaten vorgestellt, die Schichtdatendarstellung, das indirekte und das direkte Volumenrendering.

2.2.1 Schichtdatendarstellung

Eine einfache Art der Darstellung tomographischer Bilddaten ist die Schichtdatendarstellung. Die schichtweise Abtastung eines Patienten erzeugt einzelne 2D-Bilder und der Arzt stellt anhand der Auswertung der einzelnen 2D-Schichten die Diagnose und Art der Behandlung. Ärzte sind diese Art der Darstellung gewohnt und sie haben einen „guten Blick“ dafür bekommen, einzelne Strukturen zu erkennen und Krankheiten zu diagnostizieren. Anhand der Schichtdatendarstellung ist jedes einzelne Voxel eindeutig erkennbar, d.h. die Präzision dieser Darstellung ist groß. Daher ist es sinnvoll, auch am PC die Schichtdatendarstellung zur detaillierten Information anzubieten. Hierbei können Einzelheiten, wie z.B. kleine Tumore, gut gefunden werden.

Am PC ist es zusätzlich möglich, die Ansichten zu ändern. Der Arzt kann sich die Daten axial wie gewohnt anschauen und zusätzlich sagittal und koronal. Abbildung 2-7 zeigt Aufnahmen des Kopfes in den drei üblichen Ansichten.

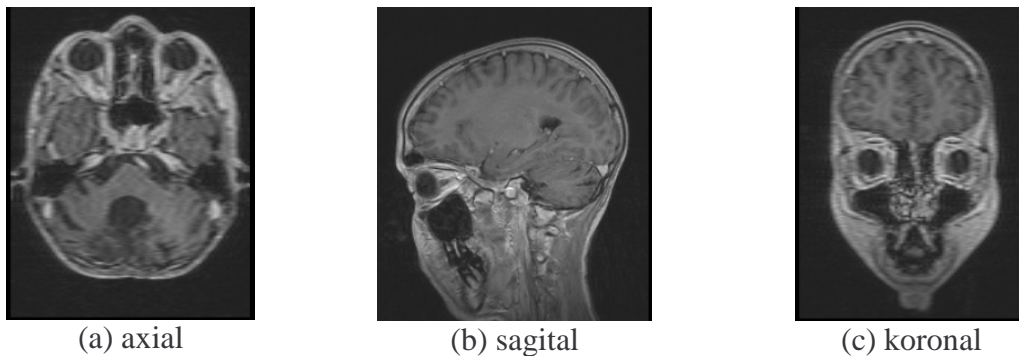


Abbildung 2-7: Schichtdarstellung eines MRT-Kopf-Datensatzes

Um diese Ansichten zu visualisieren, müssen die Intensitätswerte (Anzahl der HU-Werte = 4096) auf wenige Grauwerte abgebildet werden, zum einen da der Mensch nur zwischen ca. 100 Grauwertstufen unterscheiden kann und zum anderen kann ein Ausgabegerät (beispielsweise ein Monitor) nur wenige Grauwerte darstellen. Daher erfolgt die Abbildung der Intensitätswerte auf Grauwerte mittels einer flexibel einstellbaren TF. Insbesondere wird dafür die Fensterung, eine spezielle und einfache TF, benutzt, die in Kapitel 3 Abschnitt 3.1 genauer beschrieben wird.

Ist die TF gut eingestellt, kann der Arzt die für ihn relevanten Kontraste im Schichtbild wahrnehmen.

2.2.2 Indirektes Volumenrendering

Beim indirekten Volumenrendering (kurz VR) werden mittels verschiedener Algorithmen, z.B. Marching Cube, entwickelt von Lorensen et al. [LC87], Oberflächen berechnet und gerendert. Bei diesem Verfahren muss ein geeigneter Schwellwert (Isowert) gefunden werden, mit Hilfe dessen die Bestimmung von Isooberflächen erfolgt. Wird der Schwellwert ungünstig gesetzt, so repräsentieren die erzeugten Flächen die Gewebestrukturen nur schlecht. Einen geeigneten Schwellwert zu finden, ist meist schwer, daher werden die darzustellenden Oberflächen oft durch eine vorverarbeitende Segmentierung gewonnen. Diese ist jedoch mit viel Arbeitsaufwand verbunden.

Ein Problem dieser Methode liegt darin, dass die Exaktheit der Oberflächen täuschen kann. So ist es gerade in der Medizin wichtig, genaue Ausmaße eines Objektes, z.B. eines Tumors, messen zu können. Ein weiteres Problem besteht darin, dass indirektes VR auf die Oberflächendarstellung beschränkt ist, was bedeutet, dass Informationen im Inneren des Objektes verloren gehen.

Diese Art der 3D-Darstellung wird häufig zur Visualisierung von Gefäßen genutzt und findet hier vor allem Anwendung in der medizinischen Diagnostik von Gefäßerkrankungen. Es können Gefäßdurchmesser abgeschätzt und somit Verengungen oder Erweiterungen von Gefäßen erkannt werden.

Eine weitere Anwendung liegt in der Therapieplanung. Hierbei wird ausgenutzt, dass der Verlauf von Gefäßen mittels Oberflächendarstellung klar erkennbar ist. Anhand der Verläufe kann der Arzt die Versorgung einzelner Gebiete durch die Gefäße erkennen und er kann so festlegen, wo z.B. eine Resektion (operatives Entfernen krankhafter Organ- bzw. Körperteile)

stattfinden kann, z.B. bei Leberlebenspenden, wo es wichtig ist, dass die verbleibende Leber des Spenders ausreichend durch die verbleibenden Gefäße versorgt wird. In Abbildung 2-8 sind die Leber und deren Gefäße als Oberflächenvisualisierung dargestellt.

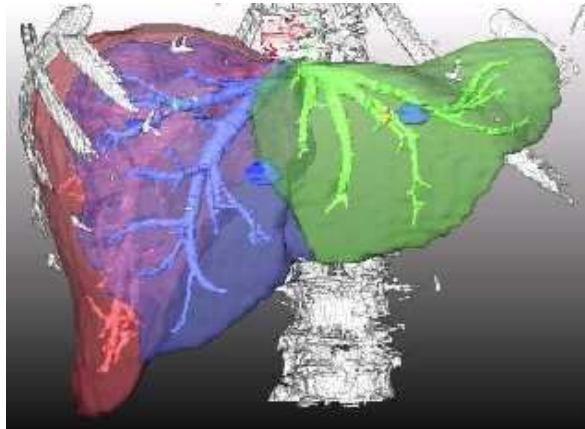


Abbildung 2-8: Oberflächendarstellung einer Leber mit Gefäßen. Quelle: [PP04]

2.2.3 Direktes Volumenrendering

Das direkte VR unterscheidet sich von dem indirekten in der Art, dass statt einen spezifischen Schwellwert zu definieren, für jedes Voxel des skalaren 3D-Feldes eine Transparenz bzw. Opazität und eine Farbe festgelegt werden. Diese werden mittels einer TF den Voxeln zugeordnet. Ziel ist es mit einer geeigneten TF, relevante Bereiche und Strukturen sichtbar zu machen und hervorzuheben. Anders als beim indirekten Volumenrendering werden hierbei die Originaldaten detailgetreu dargestellt und die Daten können ohne vorherige Segmentierung visualisiert werden.

Mit Hilfe des direkten VRs ist die Visualisierung des gesamten Volumens mit der Darstellung der inneren Strukturen bei gleichzeitig erhaltener räumlicher Orientierung möglich. Im nächsten Abschnitt werden die grundlegenden Verfahren des direkten VRs erläutert.

2.3 Direktes Volumenrendering

Ein Überblick über das direkte VR kann in [Rez01] gefunden werden.

Für das Rendering von Volumendaten existieren verschiedene Algorithmen, die unterschieden werden in bildbasiert, objektbasiert und texturbasiert.

2.3.1 Bildbasiertes Volumenrendering

Dem bildbasierten Ansatz liegt folgendes Prinzip zugrunde:

Die Berechnung eines Pixels des Ergebnisbildes ergibt sich aus der Überlagerung semitransparenter Voxel, wofür ein Strahl in die Szene geschickt und verfolgt wird. Dieses Verfahren wird auch Raycasting genannt. Der Wert des Pixels ergibt sich aus den Werten der vom Strahl durchlaufenden Voxel. Die Berechnung der Transparenz und der Farbe des Pixels kann folgendermaßen realisiert werden:

$$\begin{aligned}\lambda_i &= \lambda_{i-1} * (1 - a_i) + a_i C_i \\ \alpha_i &= \alpha_{i-1} * (1 - a_i) + a_i\end{aligned}\tag{2-2}$$

Die Farbe λ_i an der Voxelposition i ergibt sich aus der Farbe des vorherigen Voxels λ_{i-1} gewichtet mit der Differenz aus dem Wert eins und der Transparenz a_i des gerade betrachteten Voxels. Anschließend erfolgt eine Addition mit der Multiplikation der Farbe C_i und Transparenz a_i des aktuell geschnittenen Voxels. Der Transparenzwert wird ähnlich berechnet.

Dieses Verfahren ist ein Front-To-Back-Verfahren, d.h. der Anteil des Voxels am Bild wird von der Sichtebene an (vorn) nach hinten berechnet.

Der Aufwand bei diesem Verfahren liegt in der Neuberechnung der Intensitäten entlang des Sichtstrahls. Immer wenn der Datensatz transformiert wird, muss das Bild neu berechnet und entweder zwischen den 8 umgebenden Voxeln trilinear interpoliert oder der nächstliegende Nachbar berechnet werden. Dies führt zu einer rechenintensiven und zeitaufwendigen Realisierung des VRs.

2.3.2 Objektbasiertes Volumenrendering

Das objektbasierte VR betrachtet alle Voxel des Volumens und berechnet dessen Anteil am Ergebnisbild. Als Objekt wird das einzelne Voxel betrachtet. Ein bekannter Algorithmus ist hierbei der Splatting-Algorithmus [ZPBG0], der davon ausgeht, dass jedes Voxel auf der Bildebene einen Abdruck hinterlässt, d.h. das Voxel nimmt auf einen bestimmten Bildbereich Einfluss. Dieser Abdruck wird mit einem Filter, z.B. Gaußfilter, erzeugt. Es werden alle Voxel von hinten nach vorn (Back-to-Front-Verfahren) auf die Bildebene projiziert und für jedes Voxel der Szene wird dessen Anteil am zu berechnenden Bild ermittelt. Vorteile dieses Verfahrens liegen in einer guten Parallelisierbarkeit, siehe [HSCSM00]. Die Anwendung verschiedener Filter führt zu unterschiedlicher Bildqualität. Eine Beschleunigung dieses Verfahrens kann erreicht werden, wenn räumliche Kohärenzen (Zusammengehörigkeiten) ausgenutzt werden. Dies kann durch den Aufbau hierarchischer Strukturen erfolgen, z.B. Octrees, die die nicht-transparenten Voxel repräsentieren. Liegt eine derartige Repräsentation vor, so muss nur der Anteil der in dem Octree enthaltenen Voxel berechnet werden. Dies beschleunigt das VR und führt dazu, dass es teilweise schneller ist als das bildbasierte VR.

2.3.3 ShearWarp Algorithmus

Dieser Algorithmus bildet eine Mischung aus objektbasiertem und bildbasiertem VR. Levoy und Lacroute stellen in [LL94] einen sehr schnellen auf software-basierenden Shear-Warp-Algorithmus vor.

Das Ziel des Algorithmus ist die Vereinfachung der trilinearen Interpolation, indem die Abtastpunkte so platziert werden, dass eine einfache bilineare Interpolation ausreicht. Die Generierung des Bildes erfolgt hierbei in einem mehrstufigen Prozess. Zuerst wird das Volumen in achsenparallele Schichten zerlegt. Dann werden die Schichten des Volumens so geschert, dass die Sichtstrahlen senkrecht zu den Schichten stehen. Das vereinfacht das 3D-Interpolationsproblem auf eine einfache 2D-Interpolation. Danach wird der Datensatz auf die Sichtebene projiziert, wobei jedoch ein durch die Scherung verzerrtes Zwischenbild entsteht. Dieses wird mittels 2D-Warping wieder entzerrt. Bei einem starken Zoom treten bei diesem Verfahren Bildartefakte auf, da zwischen den einzelnen Schichten keine Interpolation stattfindet.

det. Außerdem wird das Volumen für jede orthogonale Blickrichtung in den Speicher geladen, d.h. insgesamt dreimal, was zu einem hohen Speicherbedarf führt.

2.3.4 Texturbasiertes Volumenrendering

Bei dieser Methode werden die beschleunigten Interpolationen in der Texture-Unit, der Graphikhardware ausgenutzt, was zu einer Beschleunigung des VRs führt. Das Volumen wird in den Texturspeicher der Graphikkarte geladen und in äquidistante Ebenen geschnitten. Dabei entstehen für jede Schnittebene texturierte Polygone, die von hinten nach vorn gezeichnet und dabei semi-transparent überlagert werden, siehe Abbildung 2-9.

Ist das Volumen zu groß für den Texturspeicher der Graphikkarte, so wird es in Blöcke, sog. Bricks unterteilt. Bei der Zusammensetzung müssen Interpolationsfehler vermieden werden. Es können Inkonsistenzen an den Schnittstellen entstehen, da in den Teilvolumen nicht ausreichend Information vorhanden ist. Dieses Problem lässt sich lösen, indem eine Voxel Ebene in jedem Brick dupliziert wird, siehe auch [Rez01].

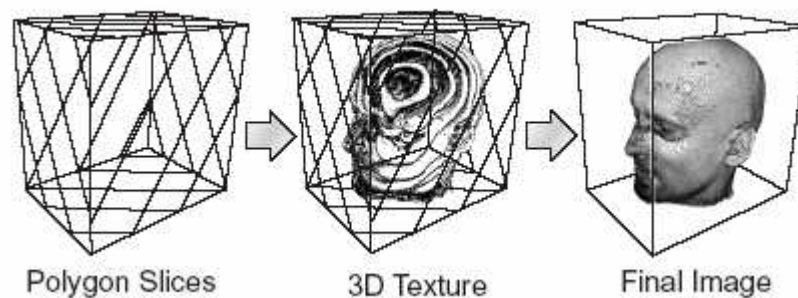


Abbildung 2-9: Unterteilung des Volumens in Polygonschichten. Quelle: [Rez01]

2D-Texturen

Bei 2D-Texturen wird das 2D-Texturemapping der Graphikhardware ausgenutzt. Dazu wird das Volumen dreimal in den Texturspeicher geladen, wobei jeweils eine Kopie für jede orthogonale Richtung erzeugt wird. Die Schichtpolygone werden entlang des Objektes ausgerichtet, siehe Abbildung 2-10 (mitte). Bei einer Rotation des Datensatzes wird jeweils die Kopie genutzt, bei der der Winkel zwischen Blickrichtung und entsprechender orthogonaler Richtung minimal ist.

Probleme:

- Umschalteffekte zwischen den drei Durchlaufrichtungen (Artefakte, Verzögerung)
- hoher Speicherbedarf

3D-Texturen

Dieses texturbasierte VR-Verfahren erfordert die Unterstützung von 3D-Texturen mit trilinear-er Interpolation der Graphikhardware (3D-Texturmapping). Das Volumen wird, wie in Abbildung 2-10 (links) dargestellt, in Schichtpolygone zerlegt, die abhängig von der Blickrichtung immer parallel zur Bildebene verlaufen. Da sich in diesem Szenario die Schichtpolygone bei Änderung der Blickrichtung ebenfalls verändern, müssen die Texturen für jedes Bild

neu interpoliert werden. Da die Hardware trilineare Interpolation unterstützt, ist dies bei interaktiver Bildrate möglich.

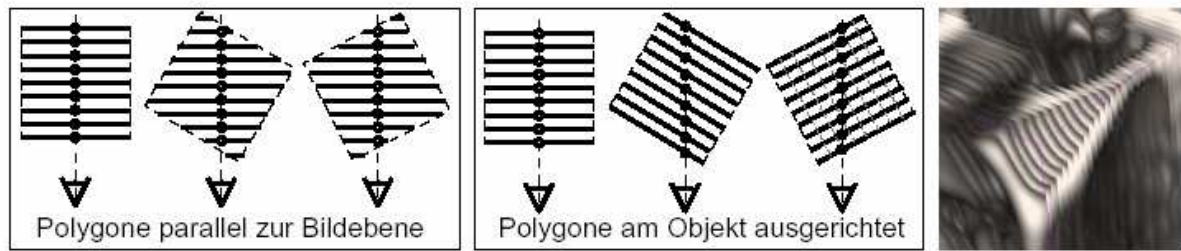


Abbildung 2-10: Ausrichtung der Polygone an der Bildebene (links) und am Volumen (mitte). Bei zu geringer Abtastrate führen beide Verfahren zu visuellen Artefakten (rechts). Quelle: [RS01]

Nachteile dieses Verfahrens liegen in den hohen Anschaffungskosten für spezifische Hardware und darin, dass für jeden einzelnen Kartentyp eine andere Implementierung nötig ist. Die Qualität des Renderings wird jedoch erheblich verbessert.

Graphikkarten, die 3D-Texturvolumenrendering unterstützen, sind beispielsweise die ATI Radeon ab Version 8500 und die NVidia Gforce ab Version 3.

2.3.5 Beleuchtung

Gradienten

An dieser Stelle werden zunächst Gradienten vorgestellt, da sie die Voraussetzungen für die Beleuchtungsberechnung bilden. Mit dem Begriff Gradienten wird der Begriff der Kanten assoziiert. Der Gradient gibt an, in welche Richtung ein Intensitätswert ansteigt bzw. abnimmt. Kanten eines Datensatzes befinden sich an den Stellen, an denen große Gradienten auftreten. Analytisch lässt sich der Gradient folgendermaßen berechnen:

$$\nabla I = (I_x, I_y, I_z) = \left(\frac{\partial}{\partial x} I, \frac{\partial}{\partial y} I, \frac{\partial}{\partial z} I \right) \quad (2-3)$$

Dabei ergibt sich der Gradient an einem Punkt $P(x, y, z)$ im Bild, aus der Ableitung der Intensität dieses Punktes in alle Richtungen.

Numerisch lässt sich dies mittels Differenzgleichungen approximieren. Dafür werden die Differenzen für jede Richtung x , y und z ermittelt. Die Differenz in x -Richtung lässt sich beispielsweise mit zentralen Differenzen folgendermaßen berechnen:

$$I_x(x, y, z) = I(x+1, y, z) - I(x-1, y, z) \quad (2-4)$$

Die Intensität des Punktes in x-Richtung ergibt sich aus der Differenz der in der Nachbarschaft vorliegenden Voxelintensitäten. Die Intensitätsberechnung in y- und z-Richtung erfolgt analog.

Die Realisierung numerischer Differenzen kann in der Bildverarbeitung mit Konvolutionsfiltern erfolgen. Ein häufig verwendeter Gradientenfilter ist der Sobelfilter, der die zentralen Differenzen anwendet und zusätzlich eine Glättung auf dem Gradientenbild durchführt. Dadurch spiegelt sich das Rauschen des Originalbildes weniger stark in dem Gradientenbild wieder. Es gibt noch weitere Gradientenfilter u.a. den Roberts-Gradient, ein einfacher Gradientenfilter. Der Sobelfilter bildet die Grundlage für die in dieser Arbeit erzeugten Kantenbilder.

Mithilfe des Sobelfilters lassen sich die Gradientenrichtungen berechnen, jedoch basiert die TF-Berechnung häufig auf den Gradientenstärken. Ziel ist es, möglichst stark abgrenzende Kanten hervorzuheben. Die Stärke einer Kante ergibt sich aus dem Betrag des Gradientenvektors.

$$\|\nabla I\| = \sqrt{I_x^2 + I_y^2 + I_z^2} \quad (2-5)$$

In Abbildung 2-11 ist ein CT-Datensatz und dessen zugehöriges Gradientenstärkebild dargestellt. Starke Gradienten sind weiß visualisiert und niedrige schwarz. Der Übergang vom menschlichen Körper zu Luft hat einen starken Gradienten, da der Intensitätsunterschied von Haut zu Luft hoch ist.

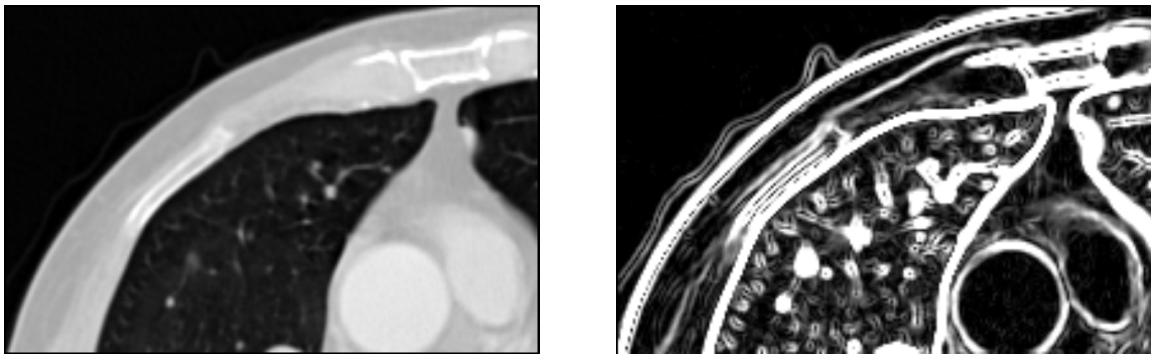


Abbildung 2-11: Intensitätsbild und zugehöriges Gradientenstärkebild

Beleuchtung

Beleuchtung ist in der Computergraphik von hoher Bedeutung für die räumliche Wahrnehmung. Für einfache Darstellungen ist schon die Definition einer ambienten Lichtquelle (Lichtquelle, die alle Oberflächen mit derselben Intensität beleuchtet) ausreichend, jedoch können mittels komplexer Beleuchtungsmodelle Kontextinformationen wie Schatten besser wahrgenommen werden. Außerdem verbessert Beleuchtung die Tiefenwahrnehmung und die Wahrnehmung von Oberflächeneigenschaften. Eine ausführliche Beschreibung zu Beleuchtungsmodellen ist in [Fol90] beschrieben, hierin wird auch das weit verbreitete Beleuchtungsmodell von Phong vorgestellt, welches neben dem ambienten Licht, noch das diffuse (wird von der Oberfläche in alle Richtungen reflektiert) und das spekulare (erzeugt Glanzlichter auf spiegelnden Oberflächen) beinhaltet. Abbildung 2-12 zeigt die Auswirkungen des

Lichtes auf das Ergebnisbild. Während im linken Bild wegen fehlender Beleuchtung nur die Form des Kopfes zu erkennen ist, können in dem mittleren Bild schon Strukturen wahrgenommen werden. In dem rechten Bild kann der Betrachter durch die Reflektionen den Bezug der Lichtquelle zum Objekt errahnen und somit den räumlichen Bezug des Objektes besser wahrnehmen.

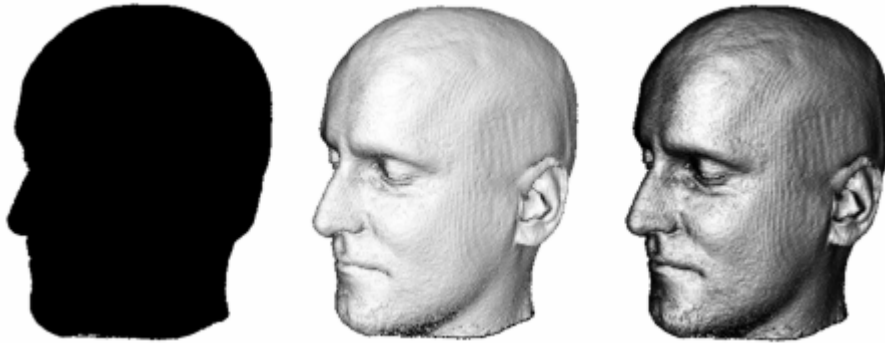


Abbildung 2-12: Isooberflächendarstellung unterschiedlich beleuchtet ohne Beleuchtung (links), mit diffuser Beleuchtung (Mitte), mit spekulativer Beleuchtung (rechts). Quelle: [Rez01]

Das Phongsche Beleuchtungsmodell ist jedoch nur ein lokales Beleuchtungsmodell, d.h. es wird nur betrachtet, wie viel Licht von einer Lichtquelle auf ein Objekt fällt. Lichtreflektionen zwischen den Objekten werden nicht berücksichtigt.

Um das Phongsche Beleuchtungsmodell beim VR anwenden zu können, müssen Normalen berechnet werden. Normalen sind Vektoren, die angeben, in welche Richtung eine Fläche ausgerichtet ist. Diese Normalen werden für das VR durch Gradientenberechnung bestimmt.

Probleme entstehen bei stark verrauschten bzw. bei hoch frequenten Daten, da das Rauschen über die Gradientenberechnung Einfluss auf die Beleuchtung nimmt. Eine Lösung für dieses Problem wird in [KPHSM03] beschrieben, in der Kniss et al. ein Modell zur Berechnung globaler Beleuchtung vorstellen und dabei speziell auf stark frequente Volumendaten eingehen.

2.3.6 Von der TF zur Lookup-Tabelle

TFs können die Intensitätswerte des Datensatzes auf Grau- und auch auf Farbwerte abbilden. Da der Mensch nur wenige Grauwerte unterscheiden kann, bietet es sich an, die Intensitätswerte auf Farbe abzubilden. Dadurch können unterschiedliche Gewebe leichter voneinander differenziert werden.

Damit die Anwendung einer TF effizient erfolgen kann, wird diese diskretisiert und auf eine Lookuptabelle (kurz LUT) abgebildet. Die einzelnen Kanäle der LUT (bei Farben vier Kanäle = RGBA) werden z.B. mit jeweils 8 Bit (256 Stufen) aufgelöst. Das Volumenrendering wird auf diese Art und Weise wesentlich beschleunigt, da die Transparenz- und Farbwerte schnell aus der LUT ausgelesen werden können, anstatt diese mittels der TF für jedes Voxel auszurechnen. Die LUT kann im Graphikkartenspeicher abgelegt werden. Das Auslesen der LUT wird von der Graphikkarte unterstützt, so dass Berechnungen dieser Art direkt auf der Hardware erfolgen können. Probleme liegen in der Beschränkung der vom Graphikkartenspeicher

unterstützten LUT-Größe, worunter die Genauigkeit des VRs leiden kann. Ältere Graphikkarten sind auf 8 Bit Volumendatensätze und 8 Bit LUTs beschränkt.

2.3.7 Klassifikation der TF

In Abschnitt 2.2 wurde die Visualisierungspipeline vorgestellt. Erst werden Rohdaten gefiltert, dann werden Visualisierungsparameter gesetzt und zuletzt soll das Bild gerendert werden. Um das Ergebnisbild zu berechnen, kann ein Strahl in die Szene geschickt werden, der mehrere Voxel durchläuft. Mithilfe einer TF werden Visualisierungsparameter gesetzt, d.h. den Voxeln werden Transparenz und Farbe zugeordnet. Der Wert des Strahls an einer Position im Datensatz wird z.B. durch trilineare Interpolation der 8 umgebenden Voxel berechnet.

Die Anwendung einer TF kann zu zwei verschiedenen Zeitpunkten während des Renderingprozesses erfolgen, hier wird zwischen pre- und post-Klassifikation unterschieden. Dabei wird die TF entweder vor der Interpolation angewendet, was die Interpolation von Grau/Farb- und Transparenzwerten nach sich zieht oder die Interpolation erfolgt erst auf den Intensitätswerten und dann werden die Farben mittels der TF zugeordnet.

Nachteile des pre-Verfahrens liegen darin, dass weil auf Farbwerten interpoliert wird, ein größerer Speicherbedarf nötig wird. Farbinterpolationen sind problematisch, da lineare Farbänderungen nicht linear wahrgenommen. Außerdem gehen durch die Diskretisierung der TFs auf eine LUT und die Anwendung dieser auf den Datensatz bereits vor der Interpolation Informationen verloren. Dies kann u.U. zu einem schlechteren Visualisierungsergebnis führen, siehe Abbildung 2-13 (links). Die pre-Klassifikation wird jedoch häufig von Graphikkarten unterstützt.

Sollen gradientenbasierte TFs verwendet werden, so ist es besser, wenn die post-Klassifikation Anwendung findet, da hier die Kantenschärfe, die für gradientenbasierte TFs die Grundlage bildet, besser dargestellt wird, siehe Abbildung 2-13 (rechts). Die post-Klassifikation wird inzwischen von neueren Graphikkarten unterstützt.

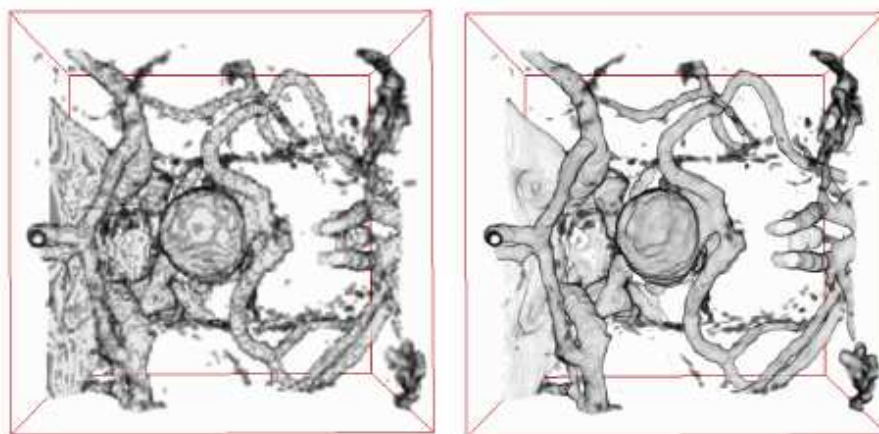


Abbildung 2-13: Klassifikation der TF

Angewendet auf einen hochfrequenten Datensatz mittels pre-Klassifikation (links) und mittels post-Klassifikation (rechts), die Schärfe der Kanten wird in der Post-Klassifikation bei verrauschten Daten besser dargestellt. Quelle [Rez01]

2.3.8 Der GigaVoxelRenderer

Der für diese Arbeit verwendete Volumenrenderer, der GigaVoxelRenderer (GVR) von Me-Vis wird an dieser Stelle vorgestellt.

Für den Entwurf zweidimensionaler gradientenabhängiger TFs ist folgendes nötig:

- Die Verwendung von 2D-LUTs muss möglich sein.
- Die Berechnung von Gradientenstärken oder die Anbindung extern berechneter Gradientendatensätze muss vom Volumenrenderer unterstützt werden.
- Der Volumenrenderer muss in der Lage sein, die Postklassifikation der TF durchzuführen, um qualitativ gute Bilder zu erzeugen.
- Am günstigsten ist es, wenn der Volumenrenderer zusätzlich noch den Texturspeicher der Hardware ausnutzt, um ein schnelles VR zu ermöglichen
- Bei gradientenabhängigen TFs ist außerdem eine gute Beleuchtungssimulation von Vorteil.

Der GVR hat zwei verschiedene Modi des VRs. Zum einen den Modus des direkten VRs, in dem der Benutzer eine TF für Grauwerte und Transparenzen einstellen kann. Zum anderen gibt es noch den Surface-Modus. Dieser dient dazu, Oberflächen im Volumen darzustellen und unterstützt dabei 2D-LUTs. In diesem Modus können verschiedene zweidimensionale LUTs angeschlossen werden, wichtig dabei ist es, den entsprechenden Datensatz (TagData genannt) an den GVR anzuschließen. In diesem TagData können die Gradientenstärken oder andere Informationen (z.B. Distanzinformationen) stehen, aus denen die Zusatzdaten für ein Voxel ausgelesen werden. In diesem Modus wird u.a. die pre- als auch die post-Klassifikation unterstützt und es wird eine Beleuchtung simuliert. Außerdem nutzt der GVR das 3D-Texturmapping der Graphikkarte aus, was ein schnelles und effizientes VR ermöglicht.

Der Modus des Surface-Renderings unterstützt alle benötigten Funktionen, sodass dem Entwurf von kantenbasierenden TFs nichts mehr im Wege steht.

Die für den GVR erzeugten LUTs werden als 2D-Bilder repräsentiert. Dabei müssen die LUTs quadratisch sein. Bei älteren Graphikkarten ist die Auflösung auf 256^2 begrenzt. Dies kann möglicherweise zu Problemen führen, da ein großer Intensitätsbereich, d.h. 4096 Intensitätswerte (CT), auf einen kleinen Farbbereich (256 Grau- oder Farbwerte) abgebildet wird.

Die in dieser Arbeit gerenderten Volumendarstellungen wurden auf einer NVidia Quadro4 980 XGL erzeugt. Diese unterstützt LUTs mit einer Bittiefe von 8Bit, d.h. pro Farbkanal können 256 Farbstufen gespeichert werden, und eine LUT-Größe von 256^2 .

2.4 Zusammenfassung

In diesem Kapitel wurden zuerst tomographische Verfahren und anschließend Visualisierungsmethoden vorgestellt. Das direkte VR ist eine Visualisierungsmethode, die einen guten Überblick über das gesamte Volumen ermöglicht. Mithilfe des indirekten VRs können nur Oberflächen dargestellt werden. Es wird ein Großteil der im Bild enthaltenen Informationen verworfen.

Während das indirekte Volumenrendering von geeigneten Schwellwerten bzw. einer zeitintensiven Segmentierung abhängig ist, muss beim direkten Volumenrendering eine geeignete TF gefunden werden. Das nächste Kapitel wird einen Überblick über TFs geben.

Die Schichtdatendarstellung darf nicht außer Acht gelassen werden. Auch sie ist eine mächtige Visualisierungsmethode, da hier Einzelheiten wahrgenommen werden können, die in einer komplexen Volumenvisualisierung verloren gehen oder durch Überdeckung von anderen Strukturen nicht sichtbar sind. Deshalb sollte dem Arzt stets die Möglichkeit der Schichtdatendarstellung gegeben werden, da der Arzt diese Art der Darstellung gewohnt ist und kleinste Unregelmäßigkeiten im Gewebe sichtbar werden.

3 Definition von Transferfunktionen

Die Stufe der Abbildung der Daten, siehe Visualisierungspipeline in Abschnitt 2.2, wird im direkten VR mittels Definition von Transferfunktionen gelöst.

Mithilfe einer fest definierten Transferfunktion werden Sichtbarkeit und Erkennbarkeit von Strukturen bestimmt. Eine Transferfunktion τ ist definiert als eine Abbildung von einem kartesischen Skalarprodukt F auf ein kartesisches Skalarprodukt mit Darstellungsparametern O :

$$\tau : F_1 \times F_2 \times \dots \times F_n \rightarrow O_1 \times O_2 \dots \times O_m \quad (3-1)$$

Normalerweise werden Intensitätswerte ($n=1$) auf Grau-/Farb- und Transparenzwerte ($m=2$) abgebildet. Hier wird von 1D-TFs gesprochen. Bei 2D-TFs können zusätzlich zu den Intensitätswerten andere Bildeigenschaften wie beispielsweise Gradientenstärken betrachtet werden ($n=2$) oder bei 3D-TFs als dritte Information die Ableitung der Gradientenstärke in Gradientenrichtung hinzugezogen werden ($n=3$). [KG01]

In den folgenden Abschnitten werden bereits existierende Verfahren zur Entwicklung von TFs vorgestellt und miteinander verglichen. Dabei wird auf 1D- und mehrdimensionale TFs eingegangen.

3.1 Eindimensionale Transferfunktionen

1D-TFs erzeugen Abbildungen in Abhängigkeit einer Eigenschaft der Datensätze. Bei dem VR von tomographischen Datensätzen werden dabei deren Intensitätswerte betrachtet. Die Generierung 1D-TFs kann mittels verschiedener Techniken erfolgen.



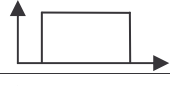

3.1.1 Interaktiv/Trial-and-Error

Bei Trial-and-Error-Methoden werden TFs interaktiv durch den Benutzer festgelegt, indem Parameter einer Repräsentationsform eingestellt werden. Eine mögliche Repräsentationsform sind elementare stetig lineare Funktionen. Diese werden im Folgenden erläutert. Anschließend wird die sog. Fensterung vorgestellt, die eine Anwendung dieser Funktionen ist.

Elementare stetig lineare Funktionen

Elementare stetig lineare Funktionen lassen sich mithilfe weniger Parameter definieren. In Tabelle 3-1 sind die am weitesten verbreiteten Parametrisierungsmöglichkeiten aufgelistet.

Tabelle 3-1: Verbreitete Konzepte für die Definition elementarer linearer Funktionen

Funktion	Parameter	Funktion
Rampe	Zentrum c und Breite w	
Dach	Zentrum c und Breite w	
Box	Zentrum c und Breite w	
Trapezoid	Zentrum c und zwei Breiten w, s	

Die Rampe definiert eine stetig ansteigende oder abfallende Funktion. Das Dach definiert eine zunächst ansteigende Funktion, die danach mit dem reziproken Anstieg wieder abnimmt. Die Box legt für einen festgelegten Bereich einen konstanten Wert fest. Eine etwas raffinierte Funktion ist das Trapez, das zunächst stetig ansteigt, in einem bestimmten Bereich konstant bleibt und anschließend wieder abfällt.

Alle Funktionen können aus der für das Trapez definierten Funktion abgeleitet werden:

$$f(t) = \begin{cases} 0 & 0 \leq t \leq c - \frac{w}{2} \\ \frac{t - c - \frac{w}{2}}{s} & c - \frac{w}{2} < t \leq c - \frac{w}{2} + s \\ 1 & c - \frac{w}{2} + s < t \leq c + \frac{w}{2} - s \\ 1 - \frac{t - c - \frac{w}{2} + s}{s} & c + \frac{w}{2} - s < t \leq c + \frac{w}{2} \\ 0 & c + \frac{w}{2} < t \leq 1 \end{cases} \quad (3-2).$$

Quelle [KG01]

Vorteile dieser Funktionen liegen in der schnellen Berechnung, d.h. es wird linear interpoliert, und vom Benutzer müssen nur wenige Parameter (maximal drei) eingestellt werden. Nachteilig ist die geringe Flexibilität, da nur wenige Funktionsgrundtypen zur Verfügung stehen.

Fensterung

Die Fensterung wird bei der Schichtdatendarstellung, siehe Abschnitt 2.2.1, verwendet. Hier kommt in der Regel die Rampe zum Einsatz. Ein Tomograph bildet die Intensitätswerte in einer sehr großen Bandbreite ab (bei CT 4096). Der Mensch kann aber nur ca. 100 Grauwerte unterscheiden und auch der Monitor kann nur ein paar Graustufen darstellen. Die Fensterung dient dazu, einen Ausschnitt der HU-Skala auf eine differenzierbare Grauwertskala abzubilden. Die Fensterbreite repräsentiert den HU-Bereich, über den die Grauwerte verteilt werden. Gewebe mit einem HU-Wert oberhalb des Fensters (Intensitätsintervall) werden weiss und unterhalb schwarz abgebildet. Ein kleines Fenster wird für Weichteile gewählt, mit einem Zentrum bei ~40 HU und einer Breite von ~250 HU. Ein breiteres Fenster wird bei Knochen und Lunge angewendet. Die Fenstertechnik ermöglicht so eine detaillierte Kontrastauflösung für einen bestimmten Intensitätsbereich. Abbildung 3-1 verdeutlicht das Prinzip der Fensterung.

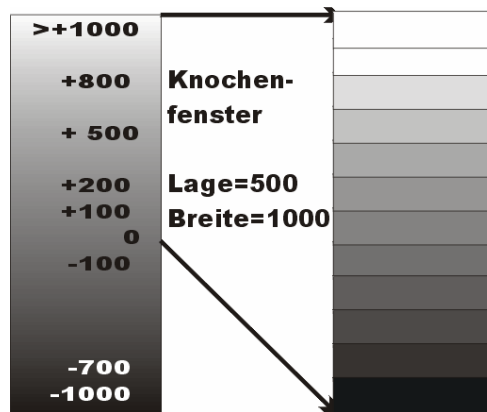


Abbildung 3-1: Prinzip der Fensterung

Abbildung 3-2 zeigt die Funktion für das Knochenfenster und einen Datensatz, der so gefenstert wurde, dass die Wirbelsäule erkennbar ist.

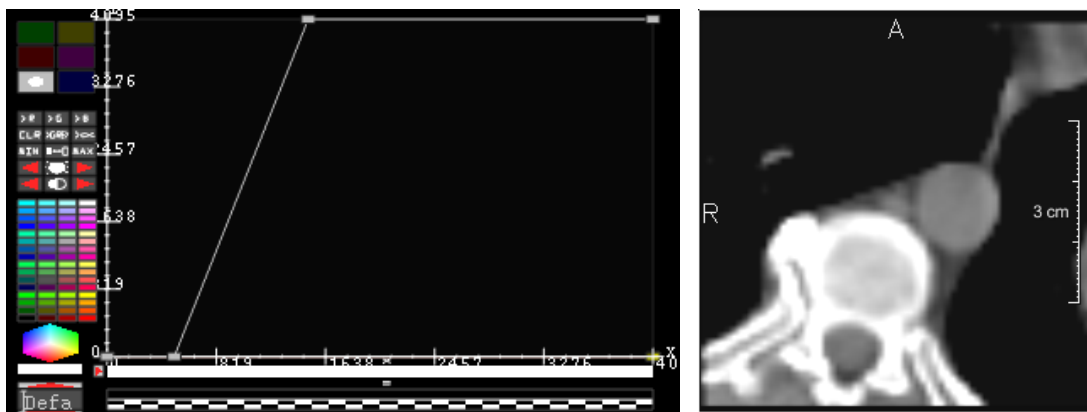


Abbildung 3-2: Fensterung für den Knochenbereich
 Links: die Transferfunktion in Form einer ansteigenden Rampe
 Rechts: das dazugehörige gefensterte Bild einer Wirbelsäule

Histogramme

Für die Erzeugung von TFs werden häufig Informationen aus Histogrammen ausgewertet. An dieser Stelle wird ein Überblick über Histogramme gegeben, die unter anderem auch die Basis für die in Abschnitt 3.2 vorgestellten mehrdimensionalen TFs bilden.

Definition: ein Histogramm eines Bildes/Datensatzes ist eine Funktion $H(\text{Intensitätswert})$, die die Häufigkeit der im Bild/Datensatz auftretenden Intensitäten repräsentiert.

Histogramme charakterisieren die Verteilungen von Werten bzw. Eigenschaften des Datensatzes. Anhand der Verteilungen können die Daten interpretiert und Visualisierungsparameter festgelegt werden. So können beispielsweise Kontraste im Bild gefunden werden. Für diese Bereiche lassen sich dann TFs definieren. Ein Vorteil von Histogrammen ist die Überschaubarkeit großer Datenmengen.

Histogramme können dem Benutzer als Kontextinformationen zur Verfügung gestellt werden, um die interaktive Definition von TFs zu unterstützen. In Abbildung 3-3 ist dies anhand zweier Beispiele verdeutlicht. Im linken Histogramm wird an der Stelle, an der viele Intensitäts-

werte vorkommen, eine Rampe definiert. Im rechten Histogramm interessieren die Bereiche, an denen eine zusätzliche Häufigkeit erkennbar ist.

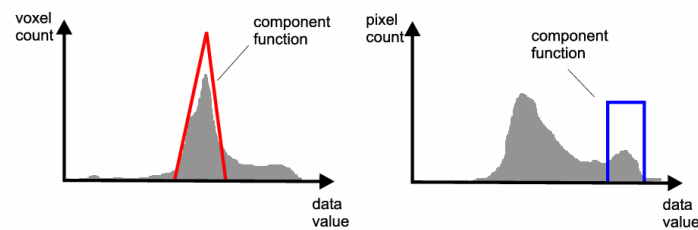


Abbildung 3-3: Histogramm hinterlegt als Kontextinformation. Quelle: [CKLG98]

Eine Erweiterung einfacher Histogramme sind mehrdimensionale Histogramme, wobei jede weitere Dimension eine andere Eigenschaft des Datensatzes beinhaltet. Kindlmann stellt in [Kin98] 3D-Histogramme vor, wobei eine Dimension die Intensitäten abbildet, die zweite die Gradientenstärke (Betrag der 1. Ableitung) und die dritte ist die 2. Ableitung in Richtung des Gradienten. Mithilfe der 2. Ableitung werden Nulldurchgänge sichtbar, die auf die Positionen von Kanten hinweisen. Anhand der Auswertung dieses mehrdimensionalen Histogramms werden Definitionen für Transferfunktionen abgeleitet.

Mehrdimensionale Histogramme veranschaulichen ähnlich wie die eindimensionalen, Verteilungen im Bild, wobei deutlich wird, in welchem Zusammenhang die einzelnen Dimensionen zueinander stehen. Ein zweidimensionales Histogramm kann zum Beispiel die Intensitäten und deren Gradientenstärke betrachten. Darin wird dann sichtbar, bei welchen Intensitätswerten starke bzw. schwache Kanten liegen.

3.1.2 Bildbasiert

Beim bildbasierten Entwurf von TFs werden Eigenschaften des Datensatzes analysiert und daraufhin TFs festgelegt und angewendet, wobei die berechneten Visualisierungsergebnisse anschließend in kleinen Vorschaubildern dargestellt werden.

Design Galleries

Marks et al. präsentiert die Design Galleries. Dem Benutzer werden kleine Bilder zur Voransicht (Presets) gerendert, aus denen er eines auswählen kann, siehe Abbildung 3-4. Die Prozesse zur Bestimmung der Parameter und zur Berechnung der TF laufen im Hintergrund ab, z.B. durch die Berechnung eines Histogramms und der Definition einer TF anhand der Auswertung der Werte des Histogramms. Dieses Verfahren ist benutzerfreundlich, da die mitunter komplexe Einstellung von Parametern auf die Auswahl von Vorschaubildern reduziert wird. Der Benutzer wählt das Ergebnisbild anhand der Vorschaubilder und das entsprechende Bild wird gerendert. Dies ist gerade für den unerfahrenen Benutzer von Vorteil, da dieser mitunter nicht weiß, wie die Parameter für eine TF eingestellt werden und er erst nach vielen Versuchen zu einem vernünftigen Visualisierungsergebnis gelangt.

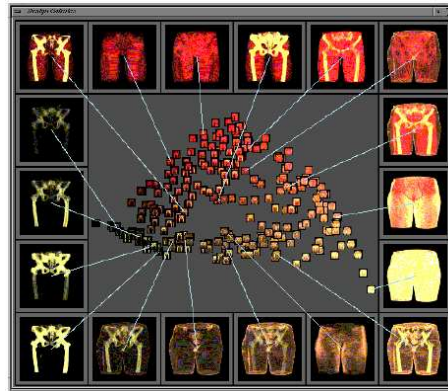


Abbildung 3-4: Design-Galerie. Quelle: [Mar97]

Stochastische Suche

Eine weitere bildbasierte Methode wurde von He et al. vorgestellt, in der die TFs-Generierung mittels stochastischer Suche realisiert wurde. Zuerst wird eine geeignete Parametrisierung gewählt. Basierend auf diesen Parametern wird eine Menge von TF berechnet und für jede TF ein kleines Vorschaubild gerendert, Abbildung 3-5 zeigt verschiedene automatisch erzeugte Vorschaubilder. Diese Menge von TFs dient als Initialisierungsmenge. Ein Such-Algorithmus wird entweder durch manuelle Eingabe des Benutzers oder durch Wahl geeigneter Metriken wie Entropie, Varianz oder Kantenenergie gesteuert. Ein genetischer Algorithmus sucht nach der besten TF.

Genetische Algorithmen ahmen Vorgänge der Natur nach. Prozesse der Natur werden beobachtet und anschließend kopiert. Das Ziel ist es, anhand der kopierten Prozesse zu einer guten Optimierungstechnik zu gelangen. Es wird auf einer Menge von individuellen Lösungen gearbeitet. Diese werden durch mehrere Prozessstufen hindurch mittels Mutation, Reproduktion oder Überkreuzungen optimiert und verbessert bis das beste Ergebnis erreicht ist.

Der von He et al. vorgestellte Algorithmus berechnet für jede TF aus der vorinitialisierten Menge mehrere neue Bilder und prüft, welche davon am geeignetsten sind. Aus der neu berechneten Menge von Bildern werden durch Überkreuzungen und Mutationen neue Initialisierungsbilder generiert. Dies wird mehrmals wiederholt bis ein zufrieden stellendes Resultat erreicht ist. [HHKP96]

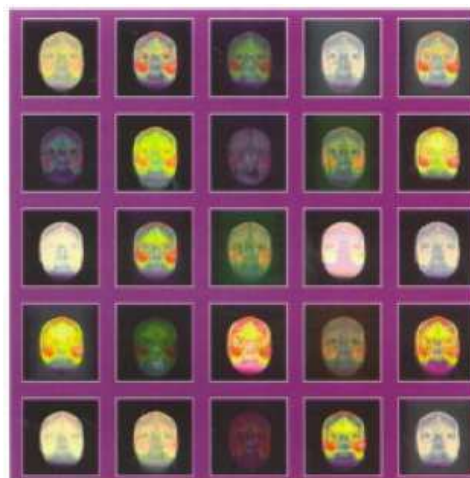


Abbildung 3-5: Verschiedene stochastisch erzeugte Ergebnisbilder für einen MRT-Kopf. Quelle: [HHKP96]

Wie bei den Design Galleries schon erwähnt, bietet das Auswählen von Beispielbildern unerfahrenen Anwendern eine gute Methode, um TFs zu definieren. Jedoch wird der erfahrene Benutzer in seinen Möglichkeiten stark eingeschränkt. Er kann nur ein paar allgemeine Visualisierungsziele bzw. ein paar spezifische Parameter angeben, jedoch gelangt er nicht an die Parameter der TF. Die Dauer der Erzeugung der Vorschaubilder ist hier ein zusätzlicher kritischer Punkt. Außerdem ist nicht zu erkennen, wann die beste Visualisierung erreicht ist. Dies ist ein allgemeines Problem bei genetischen Algorithmen. Es gibt keine Garantie, dass das beste Ergebnis erreicht wird. Außerdem ist die Rechenzeit dieser Algorithmen hoch, da kein Zeitpunkt für das Ende des Algorithmus festgelegt ist.

3.1.3 Weitere eindimensionale Transferfunktionen

Weitere Arbeiten zur Generierung von TFs beschäftigen sich mit ausgefeilteren Methoden. So können u.a. lokale TFs generiert werden. Dabei wird das Volumen in Teilvolumen zerlegt, wobei hier auf Segmentierungsalgorithmen zurückgegriffen wird. Für jedes Teilvolumen wird eine spezifische TF definiert. In [TSH98] wird ein Ansatz für die Definition lokaler TFs beschrieben. Darin wird insbesondere erläutert wie das Verfahren des VRs verbessert werden kann, um die einzelnen Segmente exakter und detaillierter zu rendern. Hierbei wird das bildbasierte VR verwendet. Die Berechnung des Intensitätswertes erfolgt mittels trilinearere Interpolation. In der $2*2*2$ -Nachbarschaft wird nach dem beteiligten Volumen gesucht, d.h. das lokale Volumen wird ermittelt.

Lokale TFs liefern bei guter Segmentierung qualitativ gute Visualisierungen, jedoch ist hierbei der Aufwand (Segmentierung, trilineare Interpolation und Nachbarschaftssuche) groß.

Neben den lokalen TFs werden häufig stetig lineare TFs verwendet. Es werden mehrere Kontrollpunkte zwischen denen linear interpoliert wird, definiert, siehe [CKLG98]. Abbildung 3-6 verdeutlicht dies an einem Beispiel. Mithilfe dieser TFs kann ein Intensitätsbereich dargestellt und ein Objekt eines anderen Bereichs hervorgehoben werden.

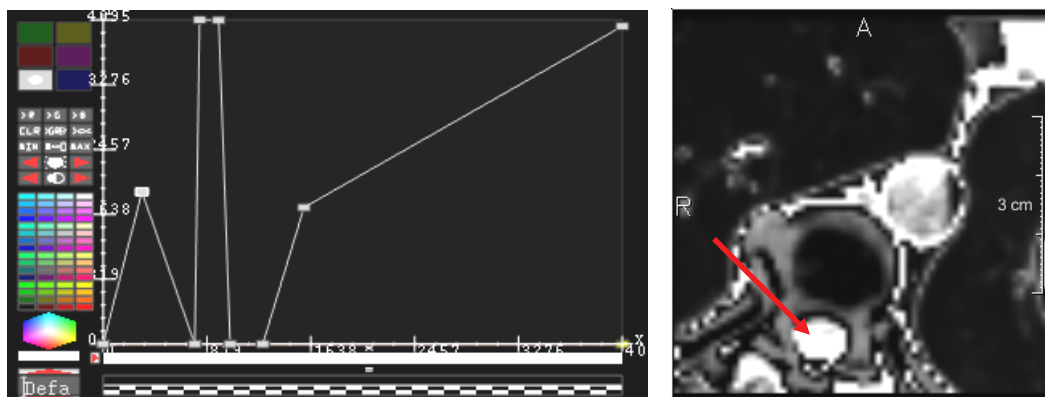


Abbildung 3-6: Stetig lineare TF und der dazugehörige Datensatz mit Hervorhebung des Rückenmarks.

3.1.4 Zusammenfassung und Vergleich

Die Einfachheit der elementaren stetig linearen Funktionen und die hohe Kontrolle über den Verlauf der Funktion ermöglichen ein interaktives Interface, welches für den Benutzer einfach

zu bedienen ist. Diese Art der TF-Einstellung ist in vielen medizinischen Workstations, (Rechnerplätze der Radiologen) umgesetzt und vielen Radiologen bereits bekannt. Die Rampen können über wenig Parameter festgelegt werden. Die Implementierung ist einfach und die Berechnung schnell.

Bildbasierte TFs erfordern eine komplexe Implementierung. Außerdem ist die Berechnung einzelner Vorschaubilder zeitaufwendig. Werden genetische Algorithmen verwendet, ist die Variabilität der erzeugten Bilder problematisch. Das bedeutet, wenn ein Datensatz untersucht wird und es werden für diesen Datensatz optimale Anfangsbedingungen gefunden, so können diese Einstellungen nicht auf einen ähnlichen Datensatz übertragen werden. Der Verlauf des Algorithmus' ist nicht vorhersehbar bzw. nicht rekonstruierbar. Der Benutzer kann darauf wenig Einfluss nehmen. Er kann sein Visualisierungsziel nur über die Auswahl einer der angezeigten Möglichkeiten erreichen. Die bildbasierten TFs sind jedoch für unerfahrene Nutzer von Vorteil, da mit dem VR experimentiert werden kann, ohne dass großes Vorwissen nötig ist.

Für die Entwicklung von 2D-TFs sind die elementaren linear stetigen TFs interessant. 2D-TFs werden durch das Hinzufügen einer zusätzlichen Dimension komplexer. Werden elementare lineare Funktionen für jeweils eine Dimension verwendet, wird dem Benutzer bereits Bekanntes zur Verfügung gestellt, wodurch das Erlernen der neuen Definition erleichtert werden kann.

3.2 Mehrdimensionale Transferfunktionen

Die Generierung mehrdimensionaler TFs erfordert die Betrachtung zusätzlicher, aus den Datensätzen abgeleiteter, Informationen.

Bei medizinischen Daten können Distanzen eine bedeutende Information sein. So kann es für den Arzt wichtig sein, Distanzen von Gefäßen zu Organen abschätzen zu können, um operative Eingriffe zu planen.

Eine wesentliche Zusatzinformation sind die Kanten bzw. die Gradienten. Mithilfe von kantendetektierenden TFs lassen sich Objekte wie Organe oder Blutgefäße hervorheben und die Position bzw. der Verlauf kann verdeutlicht und visualisiert werden. Marc Levoy war einer der ersten Wissenschaftler, der sich mit kantendetektierenden TFs beschäftigte [Lev88]. Er bestimmte Isooberflächen für das indirekte VR mit Hilfe der Kanteninformationen.

In den folgenden Abschnitten werden bereits existierende Verfahren zur Entwicklung von gradientendetektierenden TFs vorgestellt.

3.2.1 Das Kontur-Spektrum

Das Konturspektrum gehört eigentlich nicht in den Bereich der mehrdimensionalen TFs, soll jedoch trotzdem an dieser Stelle erwähnt werden, da in diesem Verfahren zusätzliche Informationen des Datensatzes betrachtet werden u.a. auch die Gradienten.

Bajaj et al. [BPS97] präsentieren eine Methode, die es dem Benutzer erleichtern soll, geeignete Isooberflächen (Isokonturen) für das indirekte Volumenrendering zu erzeugen. Dabei werden zu jeder Isokontur bzw. jedem Isokonturwert zusätzliche stochastische Parameter wie die Größe der entstehenden Oberfläche, des entstehenden Volumens oder die mittlere Gradientenstärke berechnet. Dem Anwender wird zum Auswählen einer Isokontur eine Benutzeroberfläche zur Verfügung gestellt, in der die Wahl eines geeigneten Isowertes durch Anzeige von Zusatzinformationen unterstützt wird. Er kann z.B. einstellen, dass nur die Isooberfläche mit

dem größten Gradienten visualisiert werden soll. Die berechneten Zusatzinformationen des Datensatzes werden als 2D-Funktionsgraph dargestellt.

Mit dieser Repräsentation erhält der Benutzer einen Überblick über das gesamte Volumen, was ein wesentlicher Vorteil dieser Methode ist. Von Vorteil ist außerdem, die Erweiterungsmöglichkeit dieses Verfahrens um neue Parameter.

Hierbei auffällig ist, dass zwar die Information des Gradienten betrachtet wird, die eigentliche TF jedoch nur eindimensional ist. Es wird der Isowert dargestellt, bei dem die durchschnittlichen Gradienten einen bestimmten Wert haben. Hier findet sich die Gradienteninformation nicht in der TF wieder.

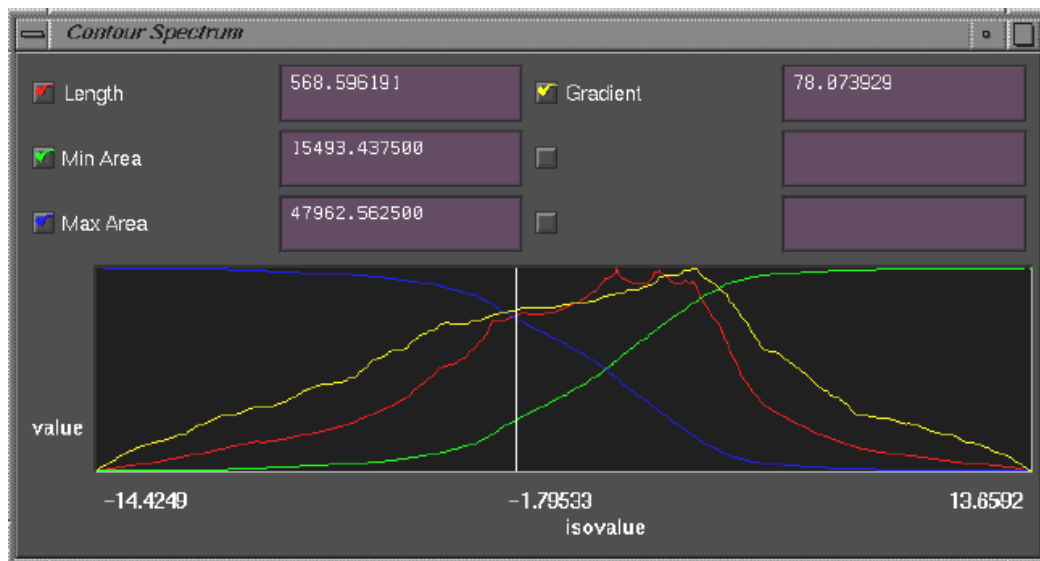


Abbildung 3-7: Parameter einer Isokontur dargestellt als 2D-Funktionsgraph
Die Zusatzinformationen sind als 2D-Funktionsgraphen dargestellt. Die weiße senkrecht stehende Linie, die verschoben werden kann, repräsentiert einen aktuell ausgewählten Isowert. In diesem Beispiel ist der Isowert -1.79553. Im oberen Teil des Fensters können die Zusatzinformationen, die bei diesem Isowert vorliegen, abgelesen werden. Anhand des Verlaufes der Zusatzinformationsfunktionen kann der Benutzer erkennen, welchen Isowert er wählen muss, um z.B. einen hohen Gradienten zu visualisieren. Quelle: [BPS97]

3.2.2 Datenbasierte Transferfunktionen

In [Kin98] wird eine TF basierend auf dem folgendem Kantenmodell entwickelt. Davon ausgehend, dass eine Kante durch starke physikalische Änderungen definiert ist, wird angenommen, dass es durch Abtastung und Rekonstruktion zu einer Verschmierung der Kante durch eine Fehlerfunktion (Pointsread-Funktion) kommt. Dabei wird davon ausgegangen, dass die Fehlerfunktion einer Gaußfunktion entspricht.

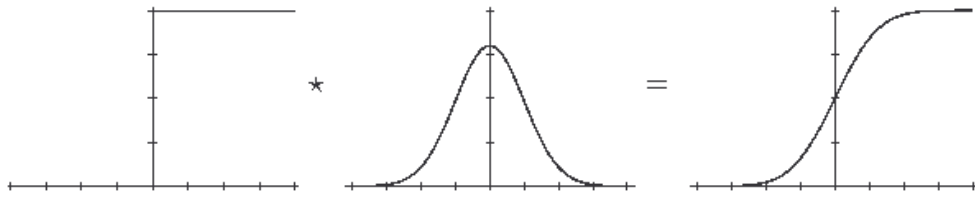
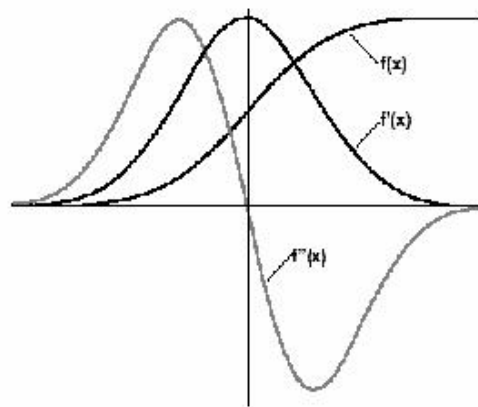


Abbildung 3-8: Modell einer Kante. Quelle: [Kin98]

In Abbildung 3-8 ist die ideale und die reale Kante in einem Modell dargestellt. Eine starke Kante wird mit einer Gaußfunktion gefiltert. Es entsteht eine geglättete Kante. Mithilfe dieses Modells entwarfen Kindlmann und Durkin eine 3D-TF, die zur Hervorhebung von Kanten gut geeignet ist. Sie entwickelten TFs unter der Betrachtung von Intensitäten, von Gradienten und der Ableitung der Gradientenstärke in Gradientenrichtung des Datensatzes.

Um die Hervorhebung von starken Kanten zu erreichen, müssen die Voxel betrachtet werden, deren Gradientenstärke groß und deren 2. Ableitung in der Nähe von 0 liegt. Siehe Abbildung 3-9, in der eine Kante $f(x)$ und deren zugehörige Ableitungen $f'(x)$ und $f''(x)$ dargestellt sind.

Abbildung 3-9: Intensitätsverlauf $f(x)$ und die zugehörige Ableitungen $f'(x)$ und $f''(x)$. Quelle: [Kin98]

Um TFs zu definieren, werden bei Kindlmann et al. jedoch nicht die einzelnen Ableitungen benutzt, sondern es erfolgt die Betrachtung der Ableitungen $f'(x)$ und $f''(x)$ in Bezug zu ihren Intensitätswerten $f(x)=v$. Dabei ist von Bedeutung: die 1. Ableitung in Bezug zu den Intensitätswerten (ist als $g(v)$ definiert) und die 2. Ableitungen ebenfalls in Bezug zu den Intensitätswerten (definiert als $h(v)$).

In der Praxis werden für jedes Voxel die 1. und 2. Ableitung bestimmt. Dann wird für jede Intensitäts-Gradientenkombination und für jede Intensitäts- 2.Ableitung-Kombination gezählt wie oft diese im Datensatz auftreten. Intern wird dies bei Kindlmann in einem 3D-Histogrammvolumen gelöst, siehe Abbildung 3-10. Dieses Histogrammvolumen ist aus Performancegründen in seiner Größe begrenzt. Gute Ergebnisse wurden mit einer Auflösung von $80^3 - 256^3$ erzielt. Aus diesem Histogrammvolumen werden $g(v)$, die 1. Ableitung in Bezug zu ihren Intensitätswerten und $h(v)$, die 2. Ableitung in Bezug zu ihren Intensitätswerten, ermittelt.

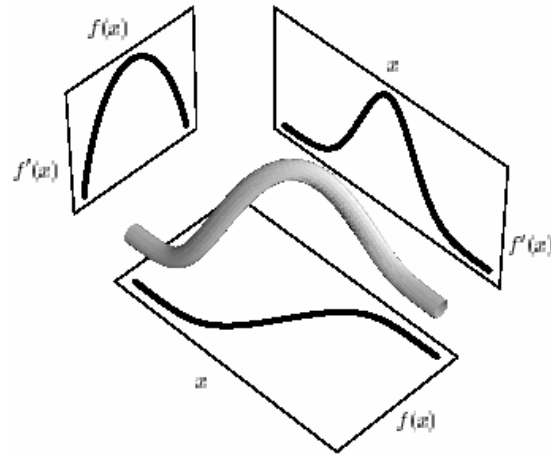
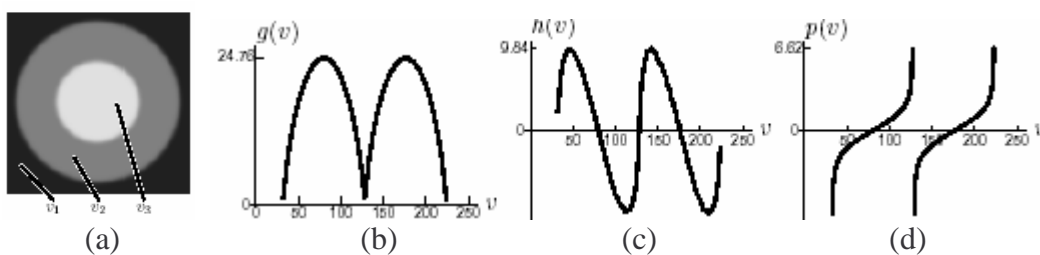


Abbildung 3-10: Histogrammvolumen. Quelle: [Kin98]

Die Kombination aus diesen beiden Funktionen, bezeichnet als Positionsfunktion, kann interpretiert werden als der durchschnittliche Abstand eines Intensitätswertes v zur nächsten Kante:

$$p(v) = \frac{-h(v)}{g(v)} \quad (3-3)$$

Mithilfe der Positionsfunktion werden die TFs erstellt. Der Benutzer kann nun mittels einer sog. Betonungsfunktion (engl. Emphasis) festlegen, wie die Kante dargestellt werden soll (soll die Kante stark oder eher schwach visualisiert werden). In Abbildung 3-11 (a) ist ein Objekt mit drei unterschiedlichen Materialien (v_1 , v_2 und v_3) dargestellt. In Abbildung 3-11 (b, c und d) sind die aus dem Histogrammvolumen ermittelten Funktionen dargestellt.

Abbildung 3-11: Kugelförmiges Objekt und dessen Funktionen $g(v)$, $h(v)$, $p(v)$. Quelle: [Kin99]

Dieses Objekt wird mithilfe von Gewichtsfunktionen, siehe Abbildung 3-12 (oben) zum einen mit weichen Kanten und zum anderen mit starken Kanten dargestellt.

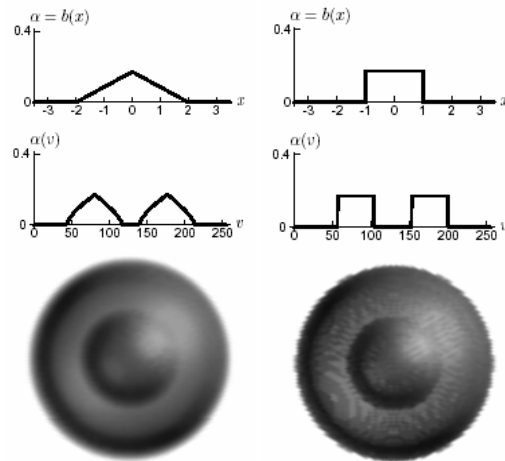


Abbildung 3-12: Objekt mit verschiedenen Gewichtsfunktionen
Weiche Kanten (links) und starke Kanten (rechts).
Quelle: [Kin99]

Benutzerinteraktion auf einem datenbasierten Verfahren

Beruhend auf den Arbeiten von Kindlmann und Durkin wurde in der gleichen Arbeitsgruppe von Kniss et al. ein Editor für die TF-Generierung entwickelt, der es dem Benutzer ermöglichen soll, schnell und einfach TFs zu entwerfen. Dabei stellt Kniss dem Benutzer ein zweidimensionales Histogramm zur Verfügung, in dem Intensitätswerte und Gradientenstärken gegeneinander abgetragen sind. Mithilfe eines Applikators (Abbildung 3-13) kann der 3D-Datensatz erkundet werden und der Benutzer erhält im Histogramm die Rückmeldung, wo die TF für diese Daten angelegt werden kann. Dies ist für den Benutzer nützlich, da er auf diese Weise einen guten Überblick über das Histogramm erhält. Er muss nicht einzig auf Basis des Histogramms mittels Trial-and-Error versuchen, TFs zu finden, sondern gelangt mit Hilfe des Applikators schnell zu einer Lösung seines Problems [KK01].

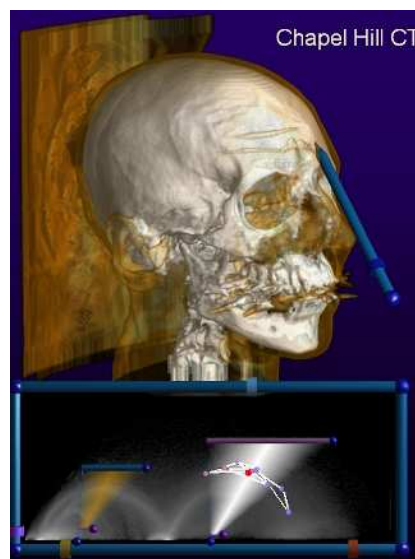


Abbildung 3-13: Applikator und die korrespondierenden Punkte im Histogramm weiß dargestellt.
Quelle: [KK01]

Der Benutzer kann zwischen verschiedenen TFs wählen, siehe Abbildung 3-14. Zur Auswahl stehen Vierecke, Dreiecke und Ellipsen, denen jeweils eine Farbverlauf und ein Transparenzverlauf zugeordnet werden. Es existiert zusätzlich noch eine TF, die für einen bestimmten Bereich eine Farbskala (z.B. Regenbogenskala) anlegt.

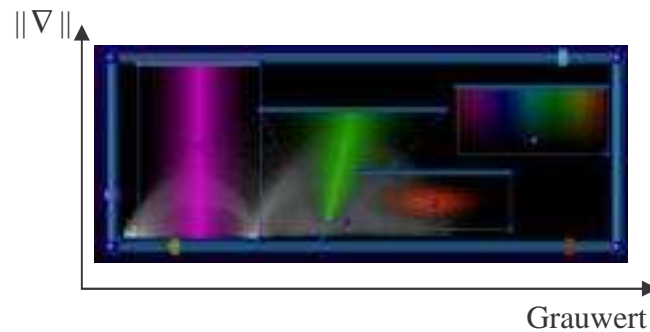


Abbildung 3-14: Auf einem Histogramm basierender TF-Editor. Quelle: [KK01]

Ein weiterer datenbasierter Ansatz

Benjamin King [Kin03] stellt in seiner Arbeit einen weiteren Ansatz zum Entwurf datenbasierter TFs vor. Basierend auf den Arbeiten von Kindlmann und Kniss erzeugt er ein zweidimensionales gradientenabhängiges Histogramm, welches er dem Benutzer zur Datenexploration zur Verfügung stellt. Auf diesem Histogramm kann der Benutzer Vierecke einzeichnen, wobei die korrespondierenden Voxel in einer Schichtdatendarstellung sichtbar werden. Dadurch kann der Benutzer erkennen, welche Bereiche im Histogramm von großer Bedeutung sind. Anhand der Interpretation der Histogramm Daten legt der Benutzer zwei 1D-TFs (stetig lineare Funktionen) fest, eine für die Transparenzen, die von den Gradientenstärken abhängig ist, und eine für die Farbzurordnung, abhängig von den Intensitäten.

Der Benutzer erhält hierbei einen guten Überblick über das Volumen, jedoch ist die Reduktion von einer 2D-Repräsentation des Histogramms auf eine letzten Endes aus zwei 1D-TFs bestehende Definition nachteilig. Dabei können Informationen verloren gehen.

3.2.3 Zusammenfassung und Vergleich

In diesem Abschnitt wurden mehrdimensionale TFs vorgestellt. Das Kontur-Spektrum erwies sich hierbei als ein erster Ansatz zur Definition von mehrdimensionalen TFs. Jedoch erfolgt die Definition der TF bzw. die Bestimmung des Isowertes nicht anhand zweier Dimensionen, sondern mittels Zusatzinformationen wird ein Isowert bestimmt. Es werden Isokonturen berechnet und zu jeder Isokontur Eigenschaften ermittelt. Anhand statistischer Auswertungen wird die beste Isokontur gewählt. Dieses Verfahren erweitert 1D-TFs, ist jedoch nicht zum Entwurf von 2D-TFs geeignet.

Kindlmann entwickelt gradientenabhängige mehrdimensionale TFs. Die Frage ist, ob ein ähnlicher Ansatz gefunden werden kann, der sich für 2D-TFs anwenden lässt. Die Idee ist, den Ansatz von Kindlmann abzuwandeln, so dass nur die Betrachtung der Intensitätswerte und Gradientenstärken erforderlich ist und dies zur Entwicklung gradientenabhängiger 2D-TFs zu benutzen. Joe Kniss und auch Benjamin King stellen dem Benutzer ein 2D-Histogramm zur

Verfügung. Dieses Histogramm enthält die Informationen Intensität und Gradientenstärke. Im Kapitel 4 wird untersucht, wie auf diesem Histogramm geeignete 2D-TFs generiert werden können.

4 Entwurf gradientenabhängiger 2D-TFs

Die Idee zur Generierung von 2D-gradientenabhängigen TFs beruht auf der Arbeit von Gordon Kindlmann. Dazu werden die Intensitäten im Datensatz in Beziehung zu ihren Gradientenstärken betrachtet. Diese lassen sich in einem 2D-Histogramm gut visualisieren. Die in Kapitel 2 Abschnitt 2.2 vorgestellte Visualisierungspipeline muss hierfür abgewandelt werden, siehe Abbildung 4-1. Nachdem die Daten gefiltert worden sind, wird ein Kantenbild erzeugt. In der Stufe der LUT-Erzeugung wird die LUT mit Hilfe der Auswertung eines 2D-Histogramms erzeugt. Die LUT, die bearbeiteten Daten sowie die Gradientendaten werden dem Volumenrenderer übergeben, der in der Stufe des Renderings das Bild erzeugt.

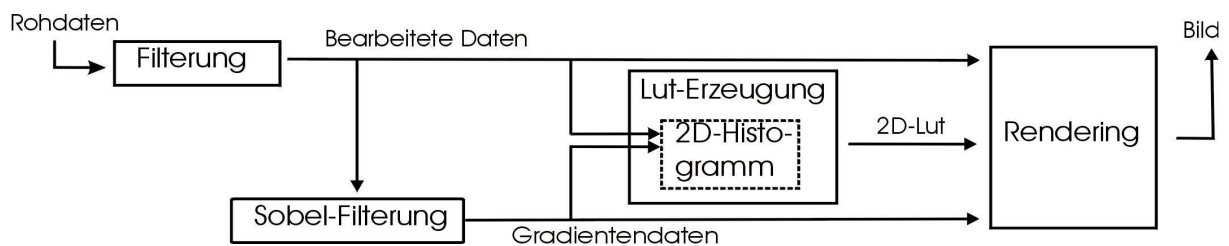


Abbildung 4-1: Visualisierungspipeline für 2D-TFs

In diesem Kapitel wird zunächst erläutert, wie der Sobelfilter im Detail aufgebaut ist und daraufhin erklärt, wie das Intensitäts-Gradienten-Histogramm generiert wird. Anschließend werden die Beziehungen, die in diesem Histogramm erkennbar sind, ausgewertet. Anhand dieser Auswertungen werden im darauf folgenden Abschnitt TFs entwickelt. Danach werden Interaktionskonzepte diskutiert. Zusätzlich wird ein Ansatz zur automatischen Generierung von 2D-TFs vorgestellt und im Anschluss daran wird erläutert wie Presets für gradientenabhängige TFs definiert werden können.

4.1 Bereitstellen von Gradienteninformationen

Der Sobel-Kantenfilter wird sehr häufig eingesetzt. Sein Einsatz ist ein Kompromiss zwischen Rauschempfindlichkeit und Sensibilität gegenüber Gradienten bei einer gleichzeitig hohen numerischen Effektivität. Bei medizinischen Datensätzen findet ein dreidimensionaler Sobelfilter Anwendung. Die Differenzenberechnung in x-Richtung wird mithilfe der in Abbildung 4-2 (a) dargestellten Kernelmaske gelöst. Dabei wird ein Voxel des Gradientendatensatzes in x-Richtung aus den 26 umgebenden Voxeln berechnet. Analog erfolgt die Berechnung in y-Richtung, siehe Abbildung 4-2 (b) und in z-Richtung, siehe Abbildung 4-2 (c).

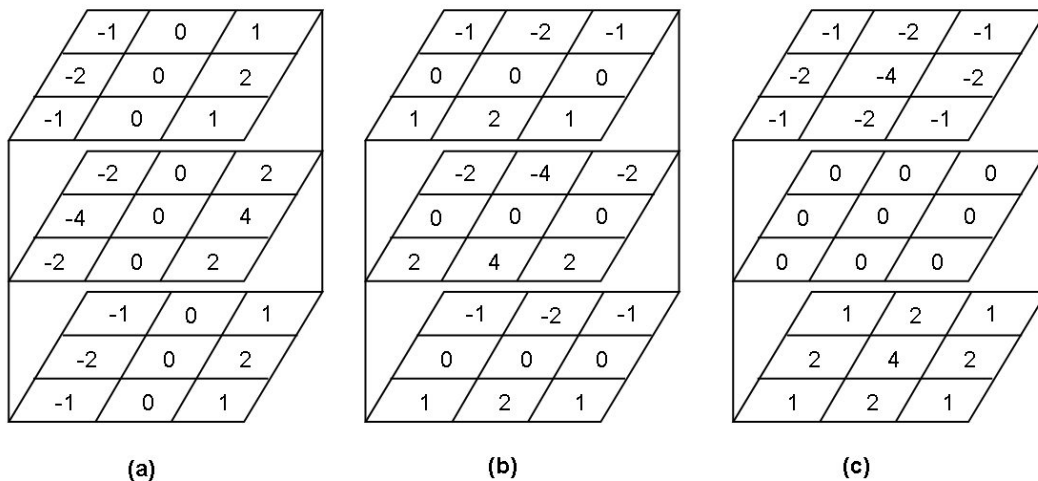


Abbildung 4-2: Kernelmasken des dreidimensionalen Sobelfilters für x, y und z

Der Sobelfilter berechnet einen Gradientenvektor. Von diesem wird der Betrag berechnet, was zu einem Gradientenstärkebild führt. Die einzelnen Gradientenrichtungen werden mit der Summe der Kernelemente (=16, Skalierung = 1/16) skaliert, um normierte Differenzen zu erhalten. Außerdem wird zusätzlich noch durch die Voxelgröße dividiert, um möglichst dieselben Gradienten bei unterschiedlicher Voxelgröße zu erhalten. Dies kann bei der Entwicklung von Transferfunktionen eine wesentliche Rolle spielen, da einheitliche Gradienten besser interpretiert werden können und die Definition von Presets (Standardeinstellungen für TFs) erleichtert werden kann.

4.2 Generierung eines Intensitäts-Gradienten-Histogramms

Um ein gradientenabhängiges 2D-Histogramm zu generieren, wird zuerst dessen Größe festgelegt. Das Histogramm wird als ein rectilineares Gitter repräsentiert, wobei ein Element dieses Gitters als Bin bezeichnet wird. Eine Auflösung in der Größenordnung von 500*500 Bins hat sich dabei als geeignet erwiesen. Dies kann schnell berechnet werden und Strukturen sind bei dieser Auflösung gut erkennbar. Im nächsten Schritt wird die Größe der Bins (binSize) für beide Dimensionen des Histogramms ermittelt. Diese ergibt sich aus der jeweiligen Größe der Dimension des Histogramms in Beziehung zu der Größe des Intensitätswertebereichs bzw. der Größe des Gradientenwertebereichs.

$$\begin{aligned} binSizeX &= DimHistInX / SizeOfRange(I) \\ binSizeY &= DimHistInY / SizeOfRange(\|f'\|) \end{aligned} \quad (4-1)$$

Im darauf folgenden Schritt wird für jedes Voxel des Datensatzes der zugehörige Intensitäts- und Gradientenwert im Histogramm abgetragen. Dies bedeutet, dass sich der Wert des entsprechenden Bins um 1 erhöht. Dadurch werden im Histogramm die Anzahl aller Intensitäts-Gradienten-Kombinationen dargestellt. Abbildung 4-3 verdeutlicht die Histogrammerzeugung an einem Beispiel. Darin wird die Intensitäts-Gradientenkombination (1124,1.99) abgebildet auf das Bin (8,0), welches dann um 1 erhöht wird.

$\text{Max}(I) = 2549$, $\text{Max}(\|f'\|) = 933$, Histogrammauflösung = 20×14

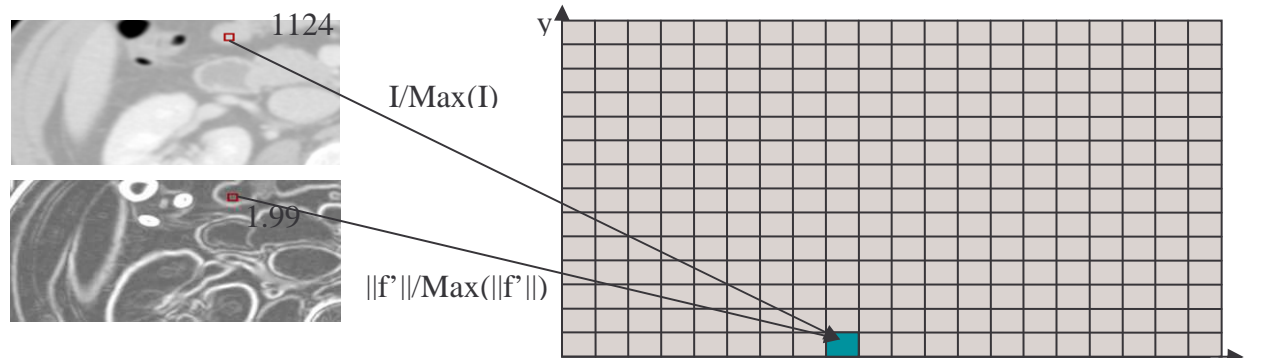


Abbildung 4-3: 2D-Histogrammgenerierung

Im Datensatz treten einige Kombinationen sehr oft auf, was zu einem hohen Wert des entsprechenden Bins führt. Wird das Histogramm als Höhenbild interpretiert, so kommen in diesem Bild Bereiche vor, in denen nur geringe Höhen vorkommen und einige Bereiche enthalten hohe Ausreißer. Diese Ausreißer führen dazu, dass kleine Höhenunterschiede verloren gehen. Daher werden die Werte des gesamten Histogramms mit dem natürlichen Logarithmus geglättet. Dabei werden die hohen Peaks stark skaliert, während kleine Einträge des Histogramms erhalten bleiben. Im Folgenden wird auf logarithmierten Histogrammen gearbeitet.

4.3 Analyse von Intensitäts-Gradienten-Histogrammen

Nach der Erstellung des Histogramms kann mit der Interpretation der Daten begonnen werden. Die Entwicklung einer TF kann basierend auf dem Histogramm erfolgen.

Um den Zusammenhang einzelner Strukturen zu verstehen, wird zuerst eine Analyse des Histogramms durchgeführt. Das Histogramm verdeutlicht Zusammenhänge zwischen Intensitätswerten, abgetragen auf der x-Achse, und Gradientenstärken, abgetragen auf der y-Achse. Dieser Zusammenhang spiegelt sich in erkennbaren Strukturen wieder, die als Bögen mehr oder weniger klar hervortreten, siehe Abbildung 4-4, in der ein auf einem Zahndatensatz beruhendes Histogramm dargestellt ist. Dieser wurde mit einem High-Resolution-CT-Gerät mit hoher Abtastrate erstellt. Der Zahn ist ein Datensatz aus dem Projekt: „The Transfer Function Bake-Off“, in dem verschiedene TFs getestet wurden [PLB01]. Er entspricht nicht realen medizinischen Datensätzen, wird hier jedoch benutzt, um die einzelnen Zusammenhänge der Strukturen im Histogramm besser zu verdeutlichen.

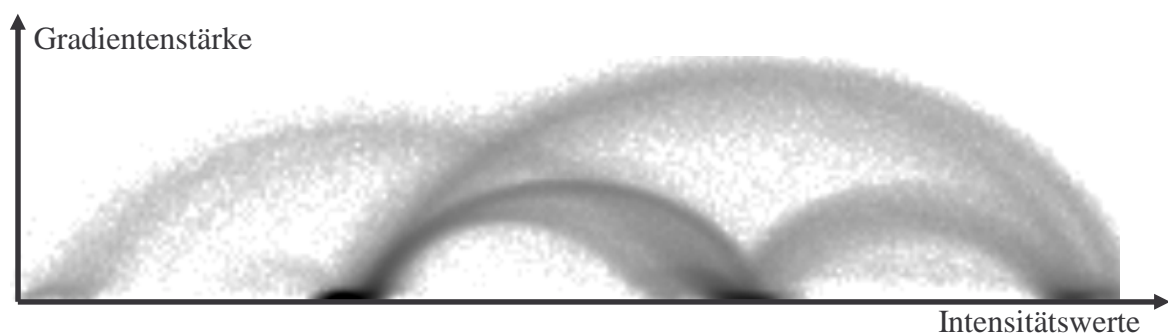


Abbildung 4-4: 2D-Histogramm

Das Zustandekommen dieser Bögen lässt sich anhand der Partialvolumeneffekte bzw. an dem Kantenmodell (Abschnitt 2.1.3) erklären. Dies bedeutet, dass die Bögen eine Kante zwischen zwei Geweben repräsentieren.

Der linke Fuß des Bogens repräsentiert den beginnenden Übergang zweier Gewebe, der am rechten Fußpunkt endet. In der Mitte, also an der Spitze des Bogens, treten beide Gewebe mit den gleichen Intensitäts-Gradienten-Kombinationen mit 50%iger Wahrscheinlichkeit auf. Dort ist der höchste Gradient und somit die stärkste Kante. In den ausgewerteten Histogrammen ist eine Symmetrie der Bögen erkennbar. Diese lässt sich anhand des Partialvolumeneffektes erklären, der sich auf beiden Seiten der Kante gleich stark auswirkt.

Abbildung 4-5 verdeutlicht die Strukturbildung im Histogramm. Dabei repräsentieren die Fußpunkte den Beginn bzw. das Ende eines Gewebes, welche durch Ellipsen gekennzeichnet sind. Gewebeübergänge (Kanten im Datensatz) zeichnen sich als Bögen ab.

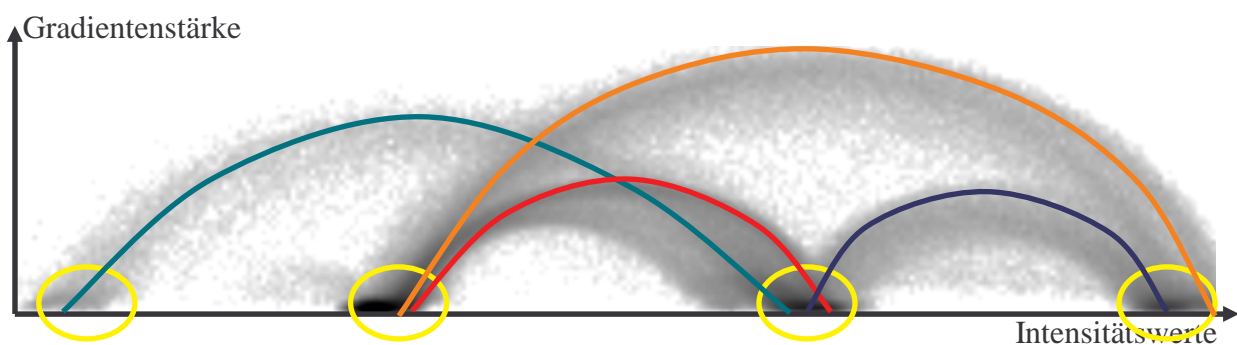


Abbildung 4-5: Bedeutung der Strukturen

In Abbildung 4-6 ist der Zusammenhang zwischen Intensitätswerten und Gradientenstärken an einem Beispiel verdeutlicht. Ausgangspunkt für dieses Beispiel ist ein Datensatz mit zwei Geweben, eines mit einer hohen Intensität (weiß) und eines mit niedriger Intensität (schwarz). In der Realität befindet sich zwischen diesen beiden Geweben eine scharfe Kante, d.h. es existiert ein harter Übergang von niedriger zu hoher Intensität (Abbildung 4-6 (a)). Resultierend aus den Abbildungseigenschaften der Tomographen (Pointspread-Funktion) und der Rekonstruktion der Volumendaten liegt eine geglättete Kante vor, d.h. zwischen Geweben niedriger Intensität und Geweben hoher Intensität bildet sich ein Intensitätsübergang (Abbildung 4-6 (b)).

In dem korrespondierenden Kantenbild (Abbildung 4-6 (c)) spiegelt sich die geglättete Kante wieder. Der Gradient steigt langsam an. In der Mitte erreicht er den Höhepunkt während er anschließend wieder abnimmt. Bei der Betrachtung des 2D-Histogramms (Abbildung 4-6 (d)) werden die Eigenschaften des Intensitäts- und des Kantenbildes wieder gefunden. Der Gradient steigt zu Beginn des Gewebsübergangs an. In der Mitte (Mischung aus beiden Geweben) ist er maximal. Danach nimmt er ab, und der Intensitätswert des Gewebes hoher Intensität wird angestrebt.

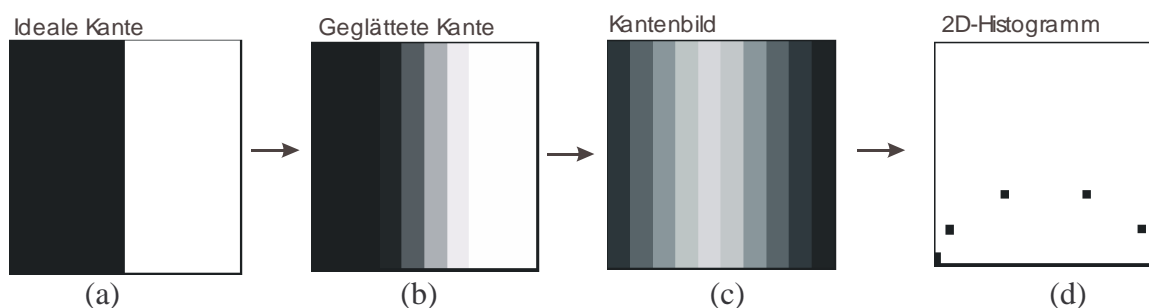


Abbildung 4-6: Zusammenhang zwischen Intensitäten und Gradientenstärken an einer Kante

Das bisher vorgestellte Histogramm entstammt einem idealen Datensatz, in dem jedes Gewebe jeweils an ein weiteres angrenzt. Die Histogramme realer medizinischer Datensätze sehen dagegen etwas anders aus. In diesem Abschnitt wird zuerst analysiert, welche Kanten in medizinischen Datensätzen auftreten und anschließend wird gezeigt, wie diese Kanten sich in den Histogrammen widerspiegeln. Die Differenzierung dieser Kanten ist weniger leicht. Diese Histogramme weisen ein starkes Rauschen auf und je nach den Aufnahmebedingungen und den verwendeten Rekonstruktionsfiltern entstehen unterschiedliche Bögen.

4.3.1 Strukturen in Histogrammen medizinischer Datensätze

In Histogrammen medizinischer Datensätze treten verschiedene Bögen auf. Ein markanter Bogen ist der, der den Übergang zwischen Luft und Lungengefäßen repräsentiert. Hier treten starke Kanten auf. Der umfassende Intensitätsbereich ist sehr groß. Dies ist der größte erkennbare Bogen und er fängt bei den HU-Werten -1024 an und endet bei -50 bis $+50$.

Ein weiterer Bogen, der für die Volumenvisualisierung interessant sein kann, ist der Bogen, der die Kante zwischen Luft und Bronchien repräsentiert. Bei der Darstellung der diesem Bogen zugeordneten Voxel werden beispielsweise in einem Lungendatensatz die Bronchien sichtbar.

Von Interesse ist außerdem der Übergang zwischen Weichteilen und Knochen. Dabei werden Knochen und evt. vorhandene Frakturen sichtbar. Außerdem sind alle Bögen wichtig, die Übergänge von kontrastmittelhaltigen Gefäßen zu anderen repräsentieren. Dies ist besonders bei der Visualisierung der Leber, der Nieren und der Blutgefäße von Bedeutung. In Tabelle 4-1 sind mögliche Bögen zusammenfassend aufgelistet. Zwei Gewebe, die durch ein „×“ gekennzeichnet sind, können in medizinischen Datensätzen benachbart sein, d.h. zwischen diesen beiden Geweben wird eine Kante sichtbar. Nicht alle in dieser Tabelle gekennzeichneten Gewebepaare werden im Histogramm auch wirklich sichtbar. Da beispielsweise zwischen Muskel und Fett der Intensitätsunterschied und damit die Gradienten winzig klein sind, zeichnet sich der entsprechende Bogen nur schwach im Histogramm ab. Dieser kann sogar in Histogrammen mit mittlerer bis hoher Binngröße verloren gehen.

Knochen wiederum grenzen an verschiedene Gewebe, was zu einem verschwommenen Bogen führen kann.

Tabelle 4-1: Nachbarschaftliche Beziehungen zwischen einzelnen Geweben

Gewebe	Luft	Bronchien	Muskel	Fett	Blut	Knochen
Luft	-	×	×	×	×	-
Bronchien	×	-	-	-	×	-
Muskel	×	-	-	×	×	×
Fett	×	-	×	×	×	×
Blut	×	×	×	×	-	-
Knochen	-	-	×	×	-	-

Anhand eines Beispiels werden die medizinisch relevanten Bögen verdeutlicht. Das Histogramm, das in Abbildung 4-7 verwendet wird, beruht auf einem weich-rekonstruierten Abdomendatensatz. Diese Abbildung verdeutlicht die sichtbaren Bögen des Datensatzes.

Der Bereich, der Muskel, Blut, Bronchien und Organe enthält, wird hierbei als Mischgewebe zusammengefasst.

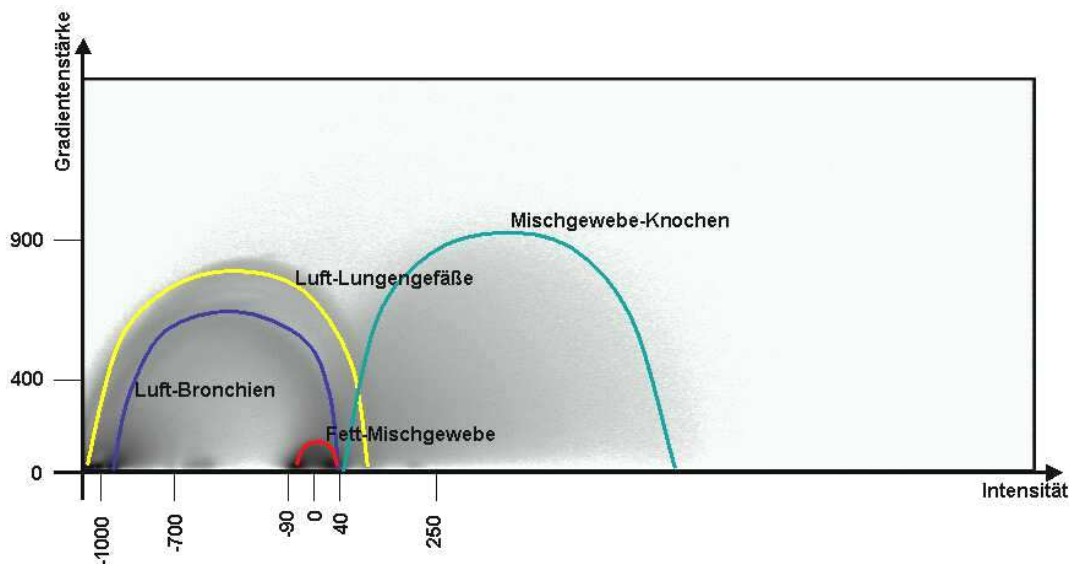


Abbildung 4-7: Bögen in einem nicht-kontrastiertem Abdomenhistogramm

Aufgrund der Standardisierung der CT-Intensitätswerte in HU-Werte können die einzelnen Gewebe leicht zugeordnet werden. Die Intensitätsskala ist bei allen Datensätzen gleich. Einzig die Gradientenstärke ändert sich in Abhängigkeit von dem verwendeten Rekonstruktionsfilter (harte, mittlere, weiche Rekonstruktion). Diese Abhängigkeiten werden im nächsten Abschnitt näher erläutert.

4.3.2 Auswirkungen unterschiedlicher Rekonstruktionsfilter in CT-Daten

Der Radiologe stellt zur Rekonstruktion des CT-Datensatzes verschiedene Rekonstruktionsfilter ein. Er kann zwischen weich (B20f/B30f), mittel (B40f) und hart (B50f/B60f) wählen. (Bf ist ein Code, der bei Siemens-CT-Geräten verwendet wird. Es ist nicht bekannt wie die Kernel im Detail aussehen. Andere CT-Geräte verwenden andere Code-Bezeichnungen). Diese Rekonstruktionskernel beeinflussen die Stärke, mit der sich Bögen abzeichnen.

In dem Histogramm werden die Kanten in Abhängigkeit von ihren Intensitäten betrachtet. Dabei werden jedoch nur die weichen Kanten gut sichtbar. Sind nur harte Kanten im Bild, können keine TFs erzeugt werden, da kein Übergang zwischen den Geweben vorhanden ist. Es kann entweder das eine Gewebe oder das andere gefärbt werden

In Abbildung 4-8 ist eine scharfe Kante abgebildet, es liegt keinerlei Glättung vor. Die Intensitätswerte liegen bei 0 und 255. Im Histogramm liegen viele Werte bei dem Gradienten 0 kombiniert mit den zwei Gewebeintensitäten. Bei einem höherem Gradienten treten wiederum bei jeder der beiden Gewebeintensitäten Häufigkeiten auf, jedoch sind diese geringer.

Für dieses Beispiel kann die TF nur schwer definiert werden. Der Benutzer muss genau wissen, wie er hier vorzugehen hat. Er kann eine TF bei den hohen Gradienten definieren, denn dort befindet sich die Kante. Die Kante liegt zum einen Teil in dem weißen Gewebe und zum anderen in dem schwarzem. Im Histogramm kann die TF auf den oberen beiden Punkten angelegt werden. Dann wird genau der Kantenbereich getroffen, der in diesem Fall jedoch sehr schmal ist und teilweise nicht im VR dargestellt werden kann. Werden die beiden unteren Punkte mit eingeschlossen, so entstehen keine Kanten im VR. Dann werden beide Gewebeararten komplett gefärbt. Dies sollte vermieden werden, denn in realen medizinischen Datensätzen, kann im VR auf diese Art und Weise nichts erkannt werden. Außerdem sind kantendetektierende TFs das Ziel.

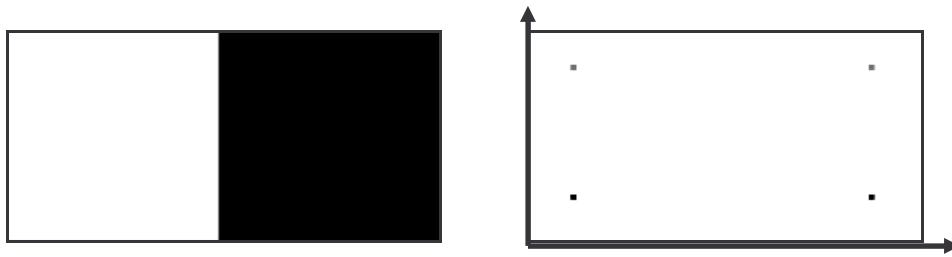


Abbildung 4-8: Scharfe Kante (links) und das entsprechende Histogramm (rechts)

Ein weiterer Punkt, weshalb hart rekonstruierte Datensätze ungeeignet sind, ist das bei der Rekonstruktion erzeugte Rauschen. Dieses Rauschen spiegelt sich in den Gradienten wieder, d.h. es entstehen Kanten, wo keine sind. Daraus folgt, dass dieses Rauschen auch im Histogramm entsteht. Die Bögen sind verschwommen, was dazu führt, dass sie nur schlecht voneinander differenziert werden können. Abbildung 4-9 und Abbildung 4-10 verdeutlichen diesen Zusammenhang an einem Beispiel. Beide Histogramme wurden auf demselben Datensatz erzeugt, wobei dieser unterschiedlich rekonstruiert wurde. In diesem Beispiel ist zu sehen, dass sich deutliche Bögen bei weich-rekonstruierten Datensätzen abzeichnen, in hart-rekonstruierten verschmelzen die Bögen miteinander.

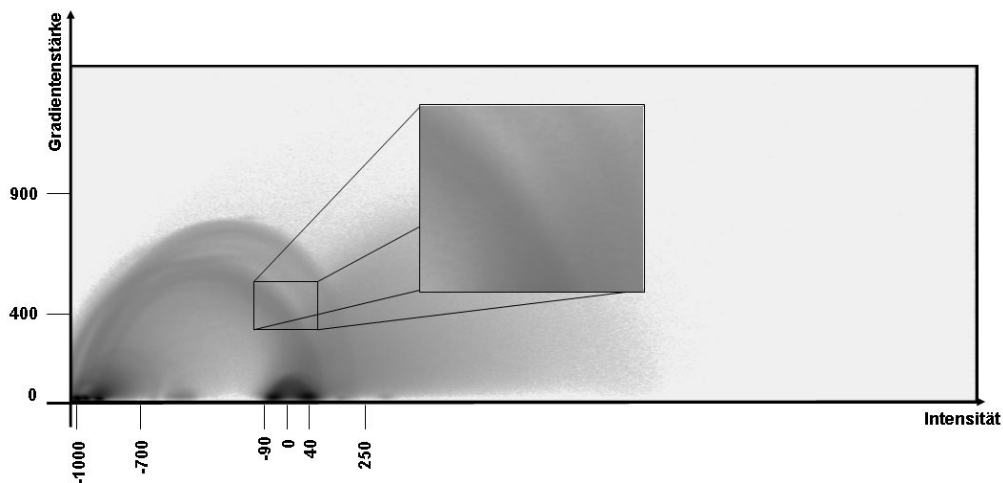


Abbildung 4-9: Histogramm eines CT-Abdomendatensatzes
Größe = $512 \times 512 \times 299$, Rekonstruktionskern = B30f
(weich), Auflösung = $0.57 \times 0.57 \times 1.0$ mm

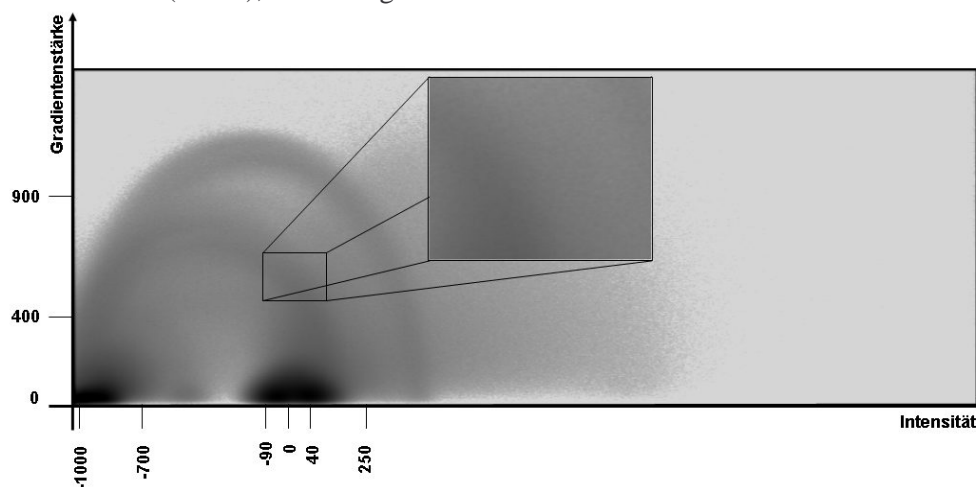


Abbildung 4-10: Histogramm eines CT-Abdomendatensatzes
Größe = $512 \times 512 \times 299$, Rekonstruktionskern = B50f
(hart), Auflösung = $0.57 \times 0.57 \times 1.0$ mm

4.3.3 CT-Datensätze mit harten Rekonstruktionskernen

Dadurch, dass sich nur weiche Kanten gut in den Bögen widerspiegeln, ist es problematisch, in hart rekonstruierten Datensätzen geeignete Bögen zu finden. Durch Glättung der Originaldaten wird der Versuch unternommen, ähnlich gute Bögen wie bei weichen Rekonstruktionsfiltern zu erreichen.

Die harten Rekonstruktionsfilter erzeugen ein verrauschtes Bild. Dieses Rauschen kann durch einen Medianfilter entfernt werden. Anschließend kann das Bild mittels eines Glättungsfilters, beispielsweise eines Gaußfilters, geglättet werden. Dies kann unter Umständen zu einem Bild führen, welches einem weich-rekonstruierten ähnelt. In Abbildung 4-11 sind die Histogramme eines hart-rekonstruierten Datensatzes dargestellt. Der Datensatz wurde zuerst mediangefiltert mit einer Kernelmaske von $3 \times 3 \times 3$, wodurch das Rauschen im Histogramm etwas abnimmt. Im nächsten Schritt wurde der Datensatz mit einer $3 \times 3 \times 3$ -Maske gaußgefiltert, was dazu führt, dass sich Bögen im Histogramm stärker herausbilden und besser erkannt werden können.

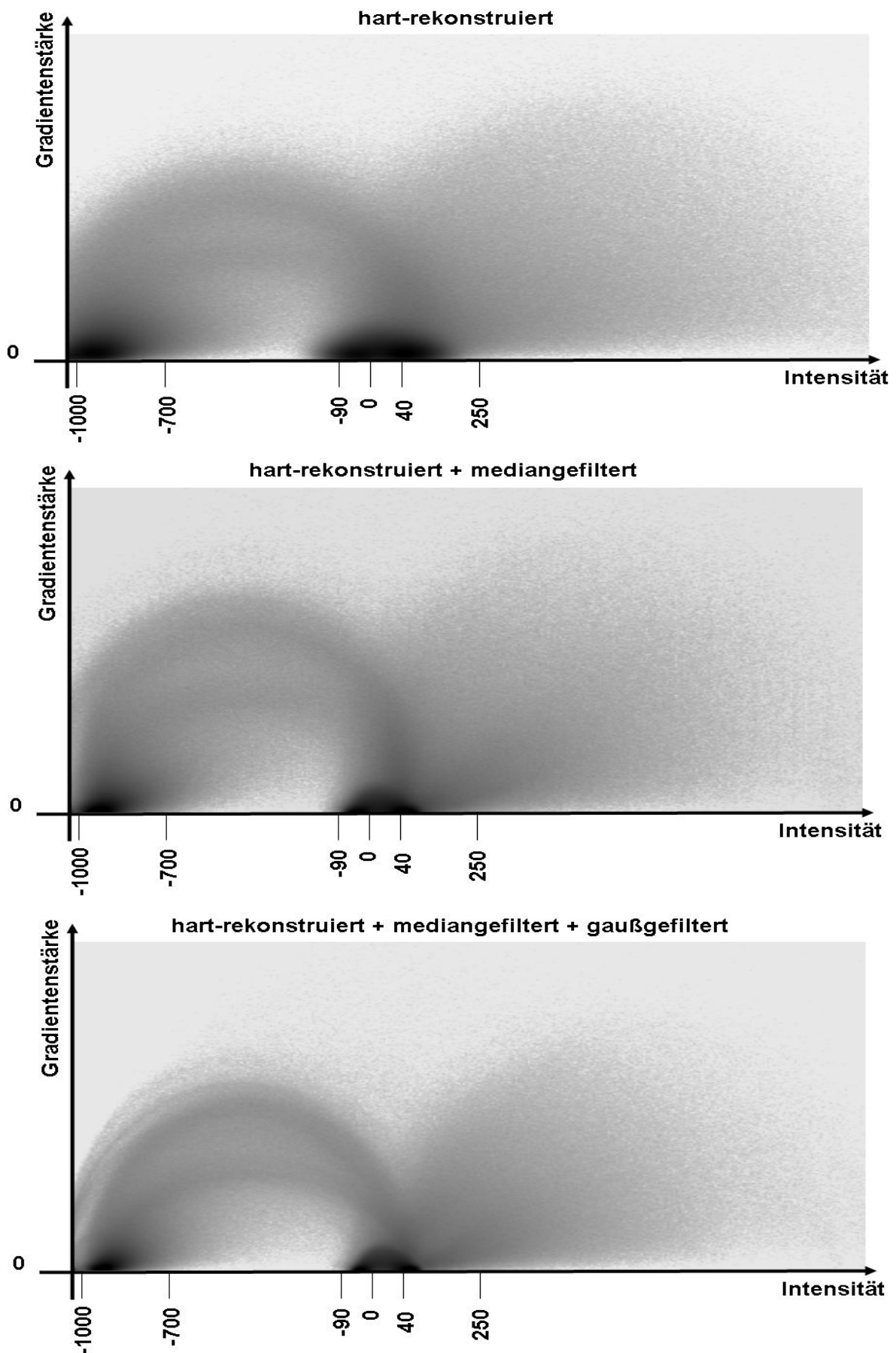


Abbildung 4-11: Auswirkungen von Filterungen eines CT-Datensatzes auf das Histogramm

4.3.4 Auswertung von MRT-Daten

Bei MRT-Daten sind die Intensitätswerte nicht standardisiert. Es kann deshalb keine allgemeine Zuordnung von Intensitätswerten zu Gewebetypen getroffen werden. Dies erschwert die Definition von TFs auf Grundlage der Histogramme erheblich.

Die Auswertung der Histogramme wird dadurch erschwert, dass in realen medizinischen MRT-Datensätzen kaum differenzierbare Strukturen zu erkennen sind. Die Histogramme können nicht so interpretiert werden wie die von CT-Datensätzen. TFs können nicht bzw. nur schwer auf der Grundlage der Histogramme definiert werden.

In realen medizinischen Daten zeichnen sich keine Bögen im Histogramm ab. Alle Histogramme weisen ein zu starkes Rauschen auf. Dabei spielt es keine Rolle, ob ein T1- oder T2-gewichteter MRT-Datensatz vorliegt, siehe Abbildung 4-12 und Abbildung 4-13. Für diese Daten ist es schwer TFs zu definieren, sodass nur schlechte Visualisierungen möglich sind.

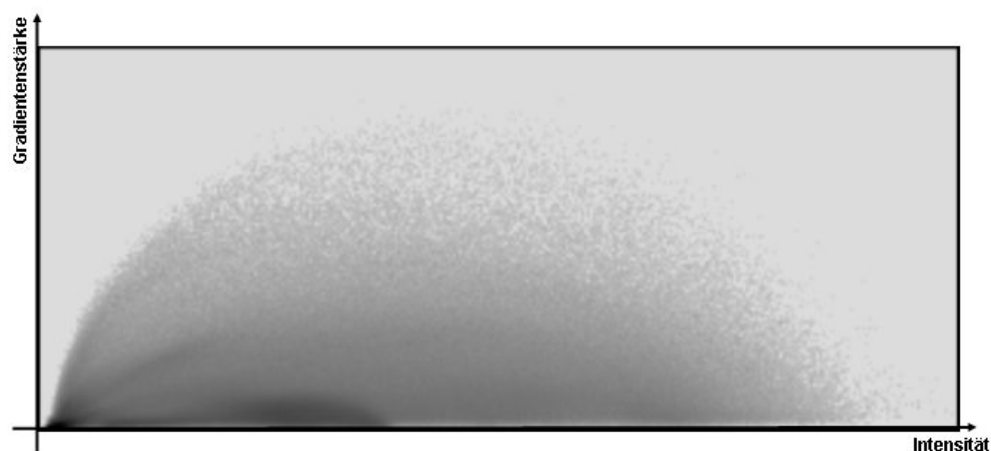


Abbildung 4-12: Histogramm basierend auf einem realen T1-MRT-Kopfdatensatz
Größe = $256 \times 256 \times 64$, Auflösung = $0.97 \times 0.97 \times 2.5$, Gewichtung = T1

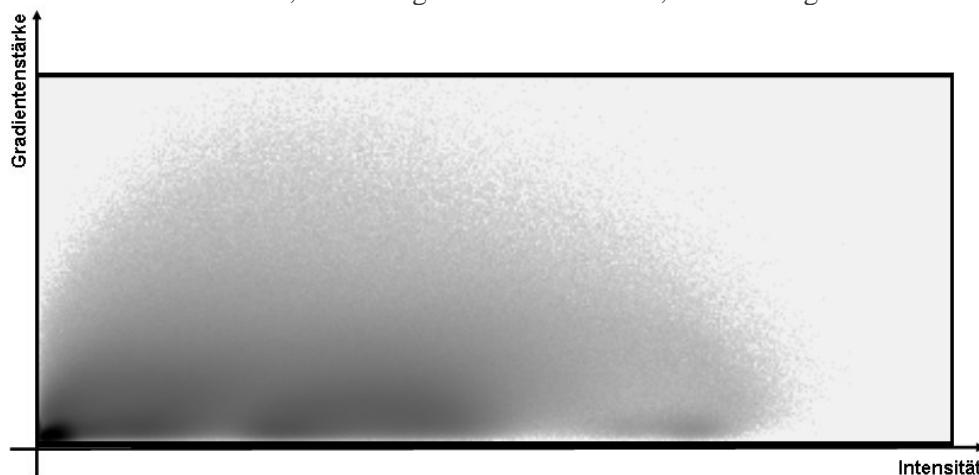


Abbildung 4-13: Histogramm basierend auf einem realen T2-MRT-Kopfdatensatz:
Größe = $512 \times 512 \times 80$, Auflösung = $0.5 \times 0.5 \times 2$, Gewichtung = T2

Vorhandenes Rauschen dieser Datensätze kann mittels eines Medianfilters entfernt werden. Dadurch können sich die Kanten in den Datensätzen besser abzeichnen, was dazu führt, dass die Bögen in den Histogrammen klarer werden. Anschließend kann der Datensatz mit einem

Gaußfilter geglättet werden, wodurch weiche Übergängen zwischen den Kanten des Datensatzes entstehen.



Abbildung 4-14: Histogramm basierend auf einem realen geglätteten T1-MRT-Kopfdatensatz:
Größe = $256 \times 256 \times 64$, Auflösung = $0.97 \times 0.97 \times 2.5$, Gewichtung = T1

In Abbildung 4-4 wird deutlich, dass auch in einem Histogramm basierend auf einem median- und gaußgefiltertem Datensatz keine deutlichen Strukturen hervortreten und somit die TFs-Definition basierend auf einem MRT-Histogramm schwer realisierbar ist.

In Abbildung 4-15 ist ein Histogramm basierend auf einem Phantom-Kopf-MRT-Datensatz dargestellt. Der Phantomdatensatz ist ein Datensatz der BrainWeb-Datenbank, die simulierte MRT-Datensätze bereitstellt, siehe [CZK98]. Dieser Datensatz enthält sehr wenig Rauschen, deshalb treten in dem Histogramm, wie bei den auf CT-Datensätzen basierenden, Bögen hervor. Der diesem Histogramm zugrunde liegende Datensatz wurde mit kleiner Abtastrate erzeugt.

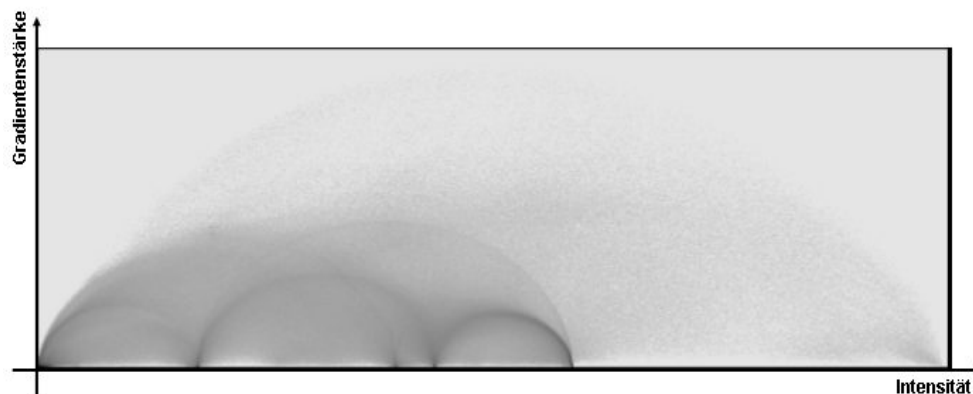


Abbildung 4-15: Histogramm basierend auf einem Phantom MRT-Kopf-Datensatz:
Größe = $181 \times 217 \times 181$, Auflösung = $1.0 \times 1.0 \times 1.0$, Gewichtung = T1

Wird auf diesen Phantomdatensatz eine simulierte Inhomogenität, beispielsweise mit einem Intensitätsverlauf von 230 bis 256HU, multipliziert, dann werden die Bögen undeutlicher und können nur noch schlecht wahrgenommen werden, siehe Abbildung 4-16. In medizinischen MRT-Datensätzen tritt diese Inhomogenität im gesamten Datensatz auf. Dadurch ändern sich die Intensitäten der einzelnen Materialien zu stark, sodass sich dies auf das Histogramm auswirkt und keine erkennbaren Strukturen sichtbar werden.

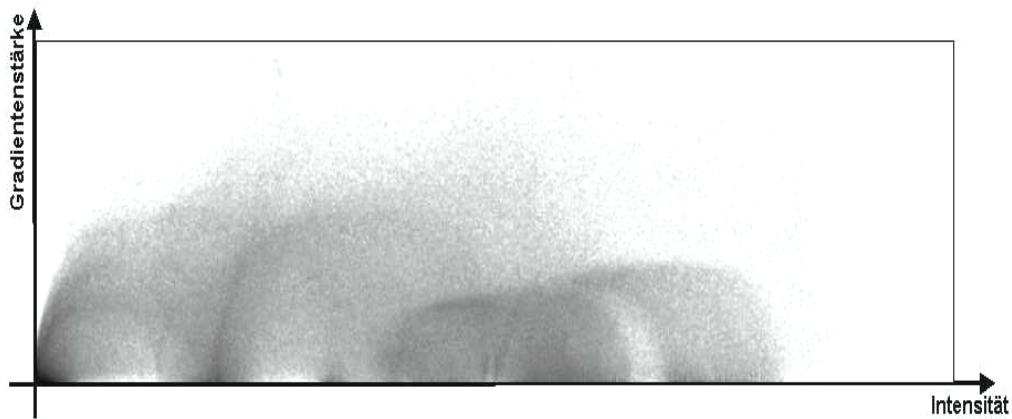


Abbildung 4-16: Phantomdatensatz mit Inhomogenität
Aufgrund der Inhomogenität geht die Bogenform
verloren.

4.4 Approximation der Bogenverläufe durch Funktionen

Anhand der Analyse der erkennbaren Strukturen im Histogramm werden Repräsentationsformen von TFs entwickelt. Interessant sind die Spitzen der Bögen, da dort die stärkste Kante liegt. Daher ist es möglich, eine zweidimensionale Repräsentation einzig an der Spitze der Bögen anzusetzen. Es besteht die Möglichkeit, wie in Kniss et al. bereits praktiziert, auf den Bögen einfache trigonometrische 2D-Objekte wie Vierecke, Ellipsen, Dreiecke oder Kreise als Repräsentation festzulegen, siehe Abbildung 4-17. Durch die Skalierung und Verschiebung dieser Objekte können unterschiedliche TFs definiert werden. So können bei einer Skalierung des Rechtecks sowohl starke und schwache Gradienten als auch einzig die starken Gradienten betrachtet werden. Auch der Grauwertbereich kann interaktiv beschränkt werden.

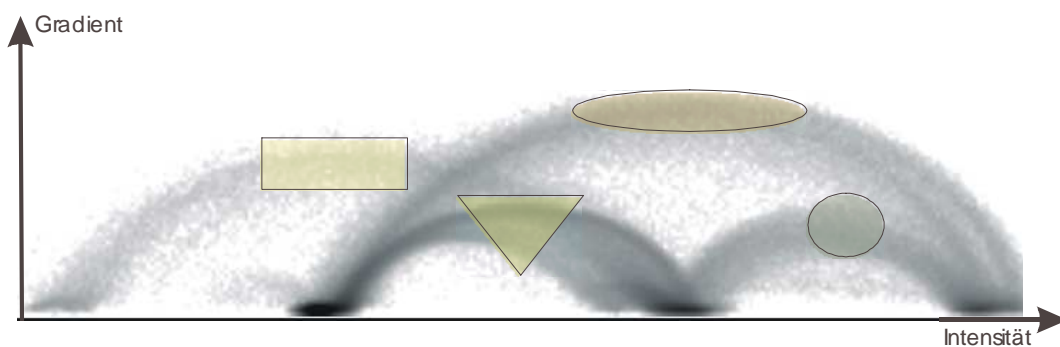


Abbildung 4-17: einfache 2D-Repräsentation

Mithilfe dieser einfachen Objekte kann die erkennbare Struktur jedoch nicht genau approximiert werden, d.h. die Repräsentation läuft immer über den Rand der Bögen hinaus und die bogenförmige Struktur kann nicht genau erfasst werden. Daher wird nach Lösungen gesucht, die Bögen möglichst exakt zu approximieren wie in Abbildung 4-18 verdeutlicht.

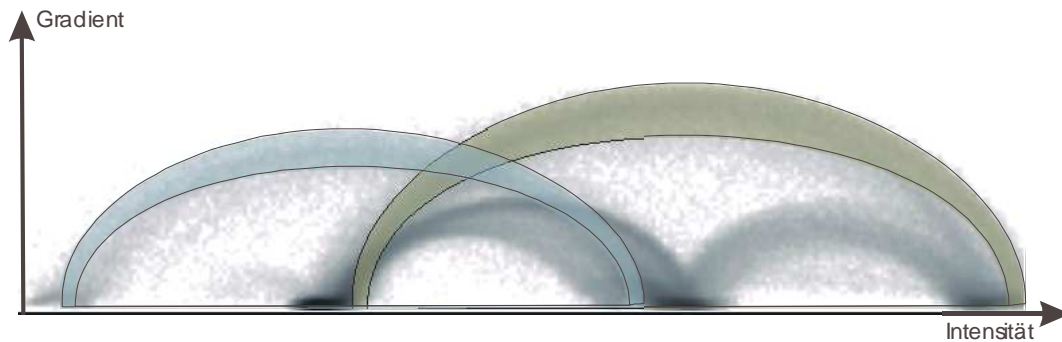


Abbildung 4-18: komplexe Repräsentation

4.4.1 Parametrische Kurven

Um gekrümmte geometrische Objekte oder Punktmenge zu approximieren, werden in der Computergraphik häufig parametrische Kurven verwendet. Eine Kurve wird dabei mittels einzelner Kurvensegmente zusammengesetzt. Im folgendem werden die drei gängigsten Kurven vorgestellt und hinsichtlich ihrer Parameter und Eigenschaften untersucht. Genaueres zu parametrischen Kurven kann in [Wat02] und in [Far90] nachgelesen werden.

Bezier-Kurven

Kubische Bezier-Kurven werden durch vier Kontrollpunkte (einen Startpunkt und ein Punkt auf der Tangente dieses, einen Endpunkt und ein Punkt auf der Tangente des Endpunktes) beschrieben. Die Bernsteinpolynome können die Gewichtsfunktionen der Bezierkurven bilden, siehe Gleichung 4-2.

$$\begin{aligned}
 B_0(u) &= (1-u)^3 \\
 B_1(u) &= 3u * (1-u)^2 \\
 B_2(u) &= 3u^2 * (1-u) \\
 B_3(u) &= u^3
 \end{aligned}
 \tag{4-2}$$

Quelle: [Wat02]

Diese Polynome sind stetige Polynome dritten Grades deren Summe für alle $u \in [0,1] = 1$ ist. Da die Summe 1 ist, liegt die Kurve immer in der konvexen Hülle der vier Kontrollpunkte. Dadurch kann die Ausdehnung der Funktion eingeschätzt werden. Um eine gegebene Punktmenge besser approximieren zu können, können mehrere Bezierkurven-Segmente aneinander gesetzt werden. Dabei müssen für jedes neue Kurvensegment vier weitere Kontrollpunkte angegeben werden. Zwischen den einzelnen Segmenten besteht eine C^0 -Stetigkeit, d.h. der Endpunkt eines Segmentes ist gleich dem Anfangspunkt des nächsten Segmentes. Sie sind in der Position stetig. Unter bestimmten Bedingungen kann C^1 -Stetigkeit erreicht werden, d.h. die Tangenten im Verbindungspunkt der Kurvensegmente haben denselben Anstieg.

Hermite-Kurven

Die kubischen Hermite-Kurven sind den kubischen Bezierkurven ähnlich. Die Kurve wird durch einen Start- und einen Endpunkt und statt der Tangenten den jeweiligen Ableitungen in den Punkten beschrieben. Außerdem werden andere Gewichtsfunktionen verwendet und die Kontrollpunkte müssen nicht innerhalb der konvexen Hülle liegen, weshalb der Kurvenverlauf schlecht eingeschätzt werden kann. Wie bei den Bezier-Kurven kann C^1 -Stetigkeit zwischen den einzelnen Kurvensegmenten erreicht werden.

B-Splines

Die Kurvensegmente werden wie bei den beiden anderen Kurven durch vier Kontrollpunkte definiert. Es gibt verschiedene Formen von B-Spline, hier wird nur die am häufigsten verwendete kubische Kurve vorgestellt. Die Gewichtsfunktionen des Polynoms ähneln denen der Bezier-Kurve.

$$\begin{aligned}
 B_0(u) &= \frac{(1-u)^3}{6} \\
 B_1(u) &= \frac{3u^3 - 6u^2 + 4}{6} \\
 B_2(u) &= \frac{-3u^3 + 3u^2 + 3u + 1}{6} \\
 B_3(u) &= \frac{u^3}{6}
 \end{aligned} \tag{4-3}$$

Quelle: [Wat02]

Der Unterschied zu den beiden anderen Kurven, entsteht bei der Zusammensetzung mehrerer Kurvensegmente. Während Änderungen der Kontrollpunkte bei den Bezier-Kurven die gesamte Kurve in geringen oder auch größeren Ausmaß beeinflussen, wirken sich Änderungen an den Kontrollpunkten der B-Splines nur auf wenige Segmente in der Umgebung der geänderten Kontrollpunkte aus. Ein Kontrollpunkt ist mit vier Abschnitten verbunden. Das Bewegen des Kontrollpunktes beeinflusst nur diese Abschnitte. Außerdem können die einzelnen Kurvensegmente eines B-Spline so aneinandergereiht werden, dass eine C^2 -Steigkeit (Stetigkeit in der 2. Ableitung) gegeben ist.

Vor- und Nachteile

Mithilfe parametrischer Kurven können Funktionen gut approximiert werden. Auch die Kurven bzw. Bögen in dem Histogramm lassen sich damit sinnvoll annähern. Jedoch ist die Einstellung vieler Parameter nötig. Da die Bögen nicht segmentiert sind und die Kontrollpunkte von Hand gesetzt werden müssen, ist eine hohe Interaktivität erforderlich. Außerdem müssen Bogenflächen approximiert werden, was dazu führt, dass kubische Kurvenflächen erzeugt werden müssen, die jedoch die Komplexität und Interaktivität noch weiter erhöhen. Daher wird im nächsten Abschnitt überprüft, ob einfachere Funktionen wie Parabeln 2. und 4. Ordnung oder Halbellipsen (kurz HEs) geeignet sind, die Bögen zu approximieren.

4.4.2 Untersuchung einfacherer Funktionen

In den Histogrammen fällt auf, dass in den Bögen immer eine Symmetrie vorhanden ist. Daher soll die Eignung einfacher symmetrischer Funktionen untersucht werden. Dazu wurden drei Funktionen näher in Betracht gezogen.

In Tabelle 4-2 ist die mathematische Beschreibung dieser zusammengefasst.

Tabelle 4-2: Mathematische Beschreibung einfacher Funktionen

Parabeln 2. Ordnung:	Parabeln 4. Ordnung:	HEs:
$y = a(x - x_0)^2 + b$ (4-4)	$y = a(x - x_0)^4 + b$ (4-5)	$\frac{(x - x_0)^2}{a} + \frac{(y - y_0)}{b} = 1$ (4-6) wobei gilt : $x, y \in \mathfrak{R}, y \geq y_0$
x_0 = Scheitelpunkt a = Skalierungsfaktor b = Verschiebung	x_0 = Scheitelpunkt a = Skalierungsfaktor b = Verschiebung	x_0, y_0 = Mittelpunkt a und b = Radien

Um zu überprüfen, welche dieser Funktionen am besten geeignet ist, wurden die Histogramme mehrerer Datensätze betrachtet und die drei Funktionen darauf angewendet. Visuell wurde abgeschätzt, welche Funktion den Bogen am besten approximiert. Hier soll die Eignung der Funktionen an drei Beispielhistogrammen verdeutlicht werden.

Für die bessere Visualisierung des Histogramms wurden hier Falschfarbdarstellungen benutzt, um die Bogenform besser hervorzuheben. Niedrige Intensitäten sind grün, mittlere sind pink/rot und hohe sind blau visualisiert.

Das 1. Beispiel: Segmentierung des Bogens, der den Übergang zwischen Luft und Lungengefäßen repräsentiert. Bei dem hier vorliegenden Datensatz handelt es sich um eine weich-rekonstruierte Abdomen-CT-Aufnahme mit einer Auflösung von $365 \times 310 \times 154$ und einer Voxelgröße von $0.652 \times 0.652 \times 1.0$. In Abbildung 4-19 ist das Histogramm dieses Datensatzes dargestellt, wobei in (a) eine HE über den Bogen aufgespannt wurde, in (b) eine Parabel 2. Ordnung und in (c) eine Parabel 4. Ordnung. Anhand der Abbildungen kann erkannt werden, dass die HE in diesem Beispiel den Bogen am besten approximiert.

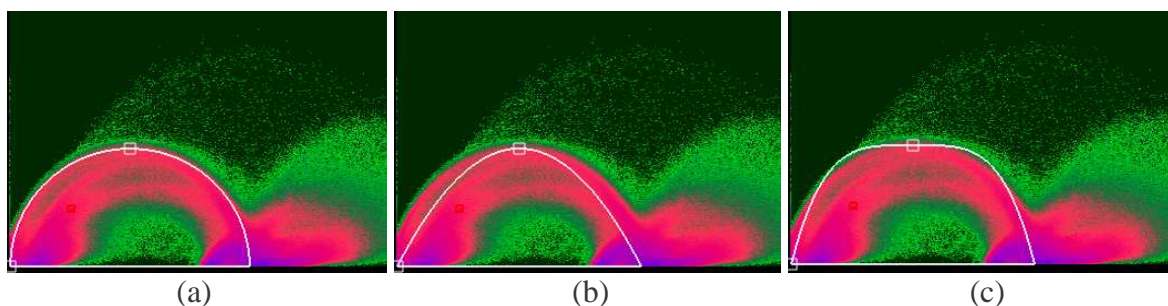


Abbildung 4-19: Segmentierung des Luft-Lungengefäße-repräsentierenden Bogens

Das 2. Beispiel: Segmentierung des Bogens, der den Übergang zwischen Fett und Knochen repräsentiert. Bei diesem Datensatz handelt es sich um eine weich-rekonstruierte Lungen-CT-Aufnahme mit einer Auflösung von $512 \times 512 \times 209$ und einer Voxelgröße von $0.818 \times 0.818 \times 2.0$. Abbildung 4-20 (a) zeigt den Bogen approximiert mit einer HE, (b) mit einer Parabel 2.

Ordnung und (c) mit einer Parabel 4. Ordnung. Auch in diesem Beispiel ist zu erkennen, dass die HE vor der Parabel 2. Ordnung am geeignetsten zu sein scheint.

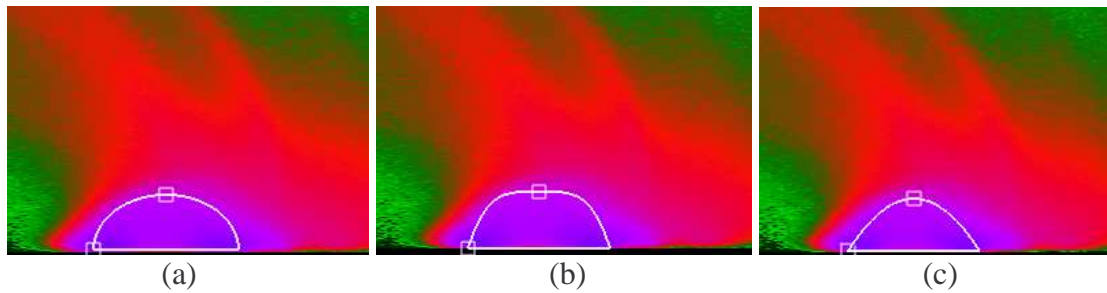


Abbildung 4-20: Segmentierung des Fett-Knochen-repräsentierenden Bogens

Das 3. Beispiel: Dieses Beispiel beruht auf dem bereits erwähnten Zahndatensatz. In diesem treten die Bögen klar hervor, daher soll an diesem Beispiel noch einmal getestet werden, welche Funktion am besten geeignet ist. In Abbildung 4-21 (a) wurde wieder die HE über einen der Bögen aufgespannt, in (b) die Parabel 2. Ordnung und in (c) die Parabel 4. Ordnung. Auch in diesem Beispiel kann erkannt werden, dass die HE die Form des Bogens am besten approximiert. Auch die Parabel 2. Ordnung nähert sich dem Bogen an. Die Parabel 4. Ordnung weicht stark vom Bogenverlauf ab.

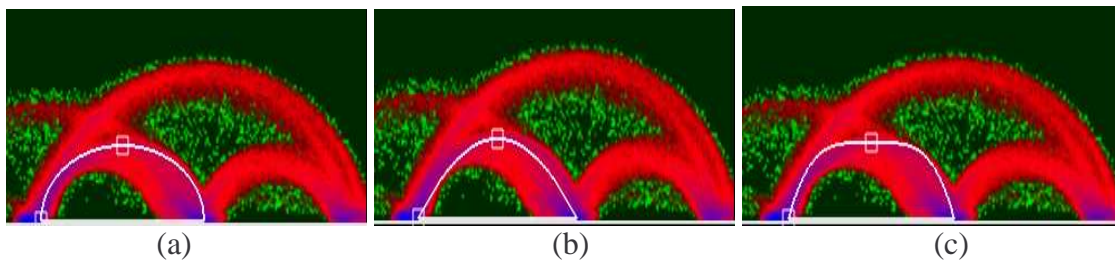


Abbildung 4-21: Segmentierung des die untere äußere Zahnwurzel repräsentierenden Bogens

Die Eignung der Funktionen wurde noch an weiteren Histogrammen getestet. Dabei zeigte es sich, dass die HE die Bögen am besten approximiert.

Außerdem wurden an einem Testdatensatz statistische Auswertungen vorgenommen, um besser einschätzen zu können, inwieweit die HE geeignet ist. Dazu wurde eine einfache Kante erzeugt, diese anschließend geglättet, so dass sie den Bedingungen eines realen medizinischen Datensatzes entspricht. Anschließend wurden die Gradientenstärken berechnet und daraufhin die Werte des sich abzeichnenden Bogens aus dem Histogramm ausgelesen und ausgewertet. In Abbildung 4-6 Abschnitt 4.3 wurde das hier verwendete Beispiel bereits vorgestellt.

Für die Bestimmung einer geeigneten Funktion wird der euklidische Abstand zu den aus dem Histogramm abgelesenen Punkten berechnet. Die Funktion, deren Summe der quadratischen Abstände am kleinsten ist, ist die am besten geeignete.

Die Auswertung der berechneten Daten ergab für die Parabel 2. Ordnung in der Summe der quadratischen Abstände den Wert 1,958, für die Parabel 4. Ordnung einen Wert von 5,79 und für die Ellipse einen Wert von 1,7008. Die Ellipse hat den kleinsten Abstand zum Testbogen.

In Abbildung 4-22 sind die drei Funktionen und der Testbogen (rot) graphisch dargestellt. Auch aus der graphischen Darstellung kann entnommen werden, dass die HE gut geeignet ist.

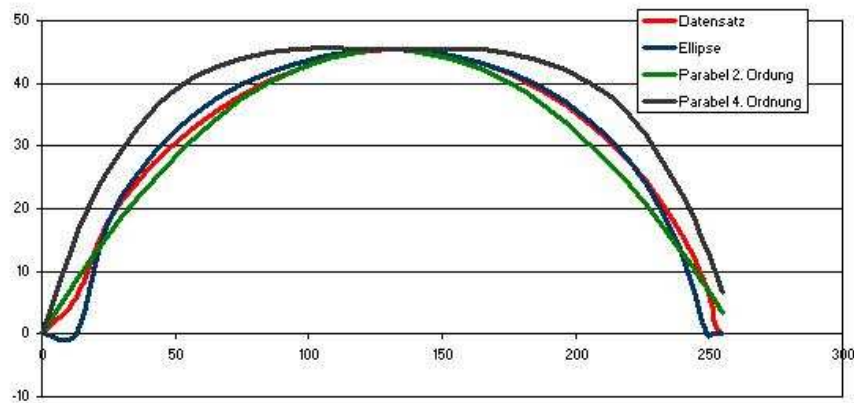


Abbildung 4-22: Ein Datensatz und die Approximation durch einfache Funktionen

Die Vorteile der HEs als auch der Parabeln im Gegensatz zu den kubischen Kurven liegen in der Einfachheit der Funktion und des geringen Interaktionsaufwandes für die Einstellung der Parameter. Durch Angabe von zwei Radien und einer Position oder einem Fußpunkt und einem Extrempunkt kann eine Halbellipse generiert werden.

4.5 Entwurf einer Repräsentationsform kantendetektierender TFs

In diesem Abschnitt wird darauf eingegangen, wie eine kantendetektierende TF, basierend auf 2D-Histogrammen, generiert werden kann.

4.5.1 Repräsentation

Die Repräsentationsform kann mithilfe der HE beschrieben werden. Da dies noch keine 2D-Repräsentationsform ist, wird ein Abstand zu der HE definiert, d.h. die HE approximiert die Form des Bogens und alle Punkte, die in einer bestimmten Distanz zu dieser HE liegen, sind für die Repräsentation relevant.

Seien d_x und d_y Abstände zur Ellipse, dann sieht die Repräsentation folgendermaßen aus:

$$\text{Repräsentation : } \frac{(x - x_0)^2}{a \pm d_x} + \frac{(y - y_0)^2}{b \pm d_y} = 1, \text{ wobei gilt : } x, y \in \mathfrak{R}, y \geq y_0 \quad (4-7)$$

Durch Variationen von d_x und d_y , d.h. verschiedenen Abständen entlang der Ellipse, können unterschiedliche 2D-Formen entstehen. Es genügt das Festlegen zweier verschiedener Abstände, einer an dem Extrempunkt der Ellipse und einer am Fußpunkt. In Abbildung 4-23 werden unterschiedliche Einstellungsmöglichkeiten der Parameter vorgestellt. In Abbildung 4-23 (a) ist eine Ellipsenform mit konstantem Abstand dargestellt, in Abbildung 4-23 (b) ist der Abstand an den Fußpunkten 0 und am Extrempunkt groß. Und in Abbildung 4-23 (c) ist

eine 2D-Form dargestellt, deren Abstand an den Fußpunkten größer ist als an dem Extrempunkt.

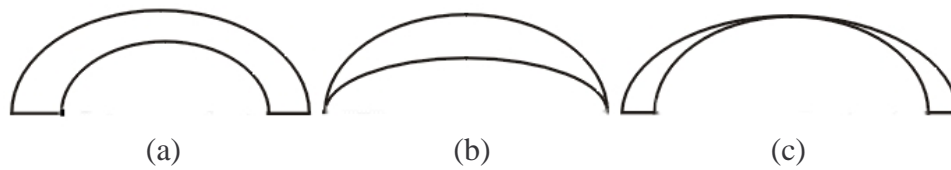


Abbildung 4-23: Halbelliptische Bogenflächen

Die Variabilität der Repräsentationsformen ist ein wesentlicher Vorteil. Durch unterschiedlich starke bzw. schwache Kanten entstehen unterschiedliche Bögen im Histogrammbild. Mithilfe der vorgestellten Repräsentationsvarianten lassen sich alle Bögen erfassen.

Mit einer HE lässt sich jedoch nur eine Repräsentation für einen Bogen anlegen. Die Auswahl mehrerer Bögen n und damit die Visualisierung verschiedener Kanten ergeben sich aus der Summe mehrerer halbelliptischer Bogenflächen (kurz HE).

$$\text{Repräsentation} = \sum_{i=1}^n HE \quad (4-8)$$

4.5.2 Steuerung der Opazitätszuordnung

Nach der Definition der Repräsentationsform werden im nächsten Schritt Opazitäten und Farbwerte bestimmt. Da nur die Bögen selbst von Interesse für das VR sind, werden alle Voxel, deren korrespondierende Bins außerhalb der halbelliptischen Bogenflächen liegen, transparent dargestellt. Dies ist wichtig, weil bei diesem Ansatz nur die Kanten dargestellt werden sollen und die Bögen die Kanten repräsentieren. Alles, was außerhalb der halbelliptischen Bogenflächen liegt, ist irrelevant. Als nächstes müssen Opazitäts- und Grauwert-/Farbfunktionen für die Voxel, deren korrespondierende Bins innerhalb der Ellipsen liegen, definiert werden.

Die einfachste Möglichkeit besteht darin, alle Punkte, die innerhalb der Repräsentationsform liegen, opak zu setzen und ihnen eine Farbe zuzuordnen. Dies ist jedoch nicht ausreichend, da die Kanten dadurch stark hervortreten, in den meisten Fällen sogar auslaufen (zu breit werden) und Strukturen mittels dieser Definition nicht erkennbar sind. Außerdem hängt die Qualität der Darstellung auch stark vom verwendeten Renderer ab, d.h. wird nur ein einfaches Beleuchtungsmodell verwendet, werden Strukturen nur schlecht wahrgenommen. Sie werden erst bei ausreichender Schattierung gut sichtbar.

Um die Kantenstärke beeinflussen zu können, wurde die Opazität so gesetzt, dass sie von dem Maximum des Bogens ausgehend, linear zu den Fußpunkten hin abnimmt. Dadurch werden die hohen Gradienten stark opak gerendert und die niedrigen transparent. Gleichzeitig wurde anfangs der Farbwert ebenfalls so definiert, dass die Intensität zu den Fußpunkten hin linear abnimmt.

In Abbildung 4-24 sind geeignete Opazitätsverteilungen zusammenfassend dargestellt. Die Bögen sind mit einem Schachbrettmuster hinterlegt, um die Transparenzen besser zu visuali-

sieren. Bei allen weiteren Darstellungen von LUTs findet sich dieses Schachbrettmuster wieder, um neben der Farbverteilung zusätzlich die Transparenzen darzustellen.

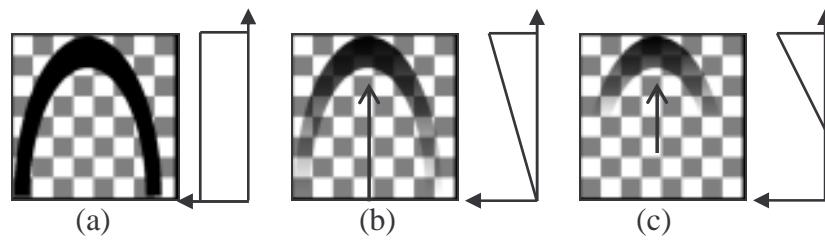


Abbildung 4-24: Verteilung der Transparenzen
 (a) volle Opazität im gesamten Bogen, (b) Opazität gleichmäßig vom Extrempunkt zu den Fußpunkten abnehmend, (c) Abnahme der Opazität in einem definierten Intervall

Anwendungsbereich

Eine weitere wichtige Eigenschaft der Transferfunktion ist der Anwendungsbereich der Opazitätswerte. Von der Spitze des Bogens ausgehend kann ein Bereich definiert werden, in dem die Opazitätsfunktion (ungleich 0) zur Anwendung kommt. Alles, was außerhalb davon liegt, wird auf die Opazität von 0 gesetzt.

Die Bedeutung dieser Option wird klar, wenn die Struktur der Bögen betrachtet wird. An der Spitze befinden sich die hohen Gradienten, d.h. dort ist die stärkste Kante. Zu den Fußpunkten hinnehmen die Gradienten ab, die Kante läuft allmählich aus. Dies bedeutet, dass mit dem Anwendungsbereich die Stärke der Kante beeinflusst werden kann. Wird der komplette Bogen gefüllt, so läuft die Kante aus, wird jedoch nur ein Teil des Bogens genutzt, so wird die Kante scharf dargestellt. Abbildung 4-25 verdeutlicht diesen Zusammenhang am Beispiel des in Abschnitt 4.3 erwähnten Zahndatensatzes.

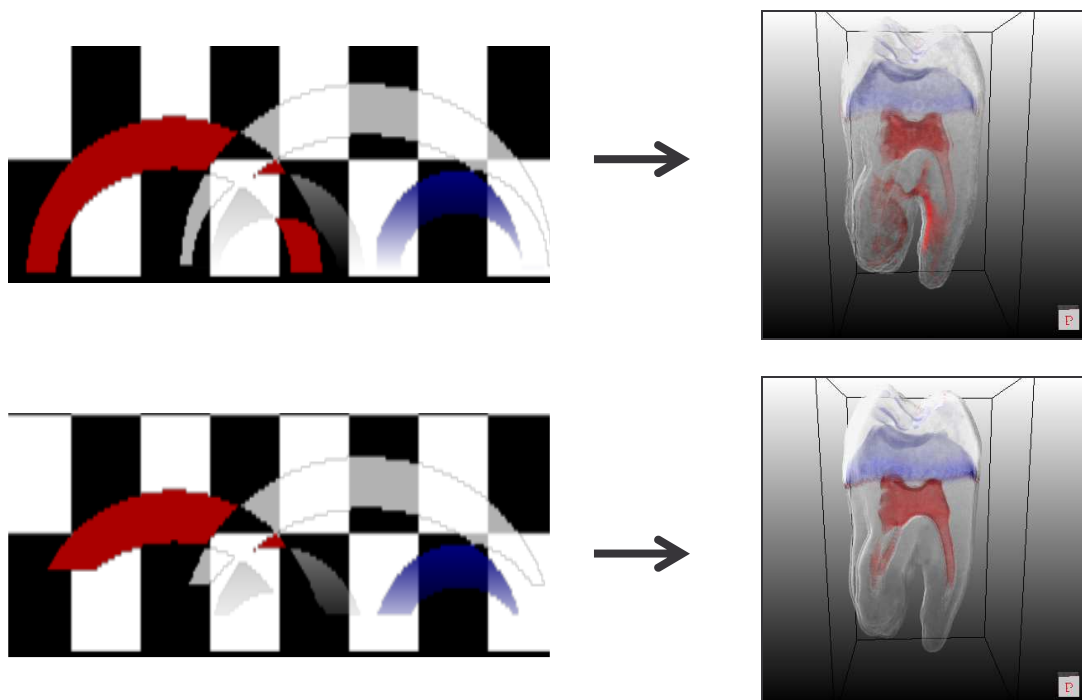



Abbildung 4-25: Auswirkungen verschiedener Anwendungsbereiche am Beispiel eines Zahns

Oben links: der rote Bogen mit einem Anwendungsbereich von 100%. Unten links: der rote Bogen mit einem Anwendungsbereich von 50%. Die Zahnwurzel läuft auf dem rechten oberen Bild aus und im rechten unteren Bild ist sie gut sichtbar.

4.5.3 Steuerung der Grauwert-/Farbzuordnung

Bei der Durchführung von Tests zeigte sich, dass mit den bisher beschriebenen TFs (Grau-/Farbverteilung gekoppelt mit Transparenzverteilung) keine guten Bilder von klinischen Daten erzeugt werden können, da die Kontrolle über die Grauwert-/Farbzuordnung in Bezug zu den Intensitätswerten zu gering ist. Daher wurden TFs entwickelt, die die Grau-/Farbwerte unabhängig von den Transparenzwerten behandeln. In Kapitel 3 Abschnitt 3.1.1 wurden elementare lineare Funktionen vorgestellt. Diese sollen hier zur Anwendung kommen, da sie leicht zu realisieren sind und dem erfahrenen Benutzer aus anderen Applikationen bereits vertraut sind. In Abbildung 4-26 sind alle Funktionen an einem Beispielbogen visualisiert. In diesem Beispiel wird der Grauwert  linear interpoliert.

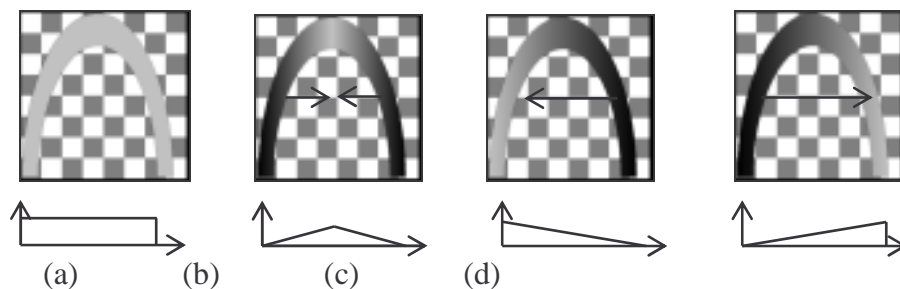


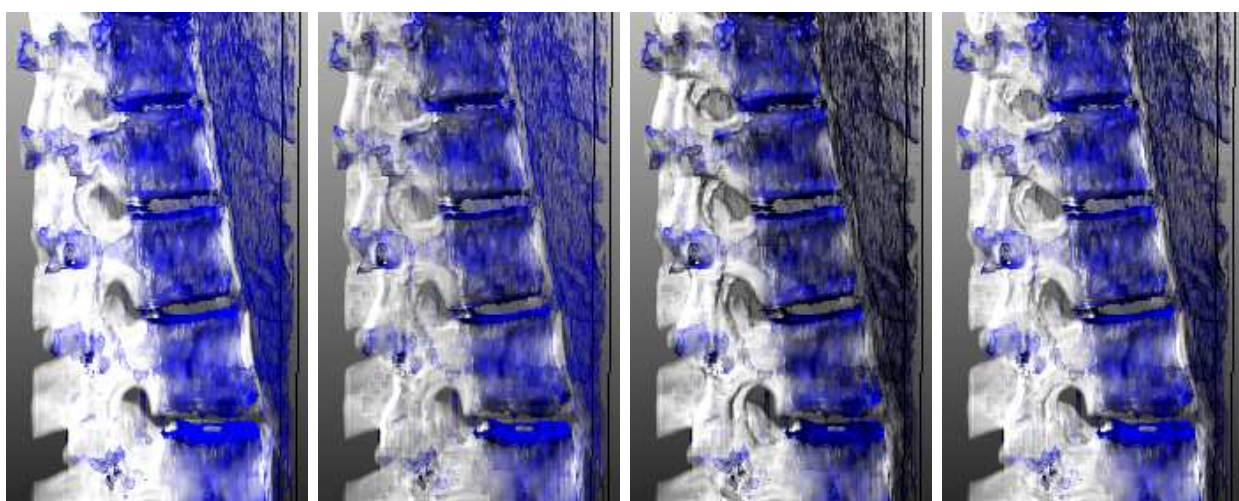
Abbildung 4-26: Farbverteilungen

(a) Box: konstanter Farbwert im gesamten Bogen, (b) Dach: linear ansteigende und abfallende Intensität des Farbwertes, (c) Rampe: linear abfallende Intensität des Farbwertes, (d) Rampe: linear ansteigende Intensität des Farbwertes

Dieses Verfahren lässt sich auch auf Farben anwenden. Dabei wird die Interpolation auf jedem einzelnen Farbkanal ausgeführt. Die Farbe wird dabei immer nach schwarz interpoliert. Beispielsweise wird die Farbe blau in $RGB = [0,0,1]$ linear nach schwarz $[0,0,0]$ interpoliert, indem auf jedem Farbkanal eine Interpolation durchgeführt wird.

Mit diesen stetig linearen Funktionen kann eine intensitätsabhängige Färbung der Kanten realisiert werden.

Auswirkungen der Farbverläufe an einem Beispiel



(a) Box-Farbverteilung (b) linear abfallende Rampe (c) linear ansteigende Rampe (d) Dach

Abbildung 4-27: Farbverteilungen am Beispiel des VRs der Wirbelsäule

In Abbildung 4-26 sind die Farbverteilungen am Beispiel einer Wirbelsäule visualisiert. Mit der Boxverteilung können die Strukturen am schlechtesten dargestellt werden. Mit den linearen Rampen und dem Dach treten intensitätsbasierte Färbungen auf, die einen räumlichen Eindruck entstehen lassen. Die Wirbelsäule wirkt plastischer.

Farbzuordnung einzelner Bögen

Grauwert-/Farbverlaufsfunktionen werden benötigt, da diese das Volumenrendering realistischer bzw. plastischer aussehen lassen. Realistische Darstellungen können zum einen durch Beleuchtung erreicht werden und zum anderen durch eine intensitätsabhängige Färbung. Die intensitätsabhängige Färbung ist besonders dann wichtig, wenn der Volumenrenderer nur eine ambiente Beleuchtung zur Verfügung stellt. Durch unterschiedliche Intensitätswerte können Strukturen des Objektes wahrgenommen werden. Abbildung 4-28 verdeutlicht diesen Zusammenhang. In der Abbildung wird ein Teil der Wirbelsäule dargestellt. Die Voxelgröße beträgt $0.625 \times 0.625 \times 1.0$ und der Datensatz wurde weich rekonstruiert. An diesem Beispiel kann erkannt werden, dass in beleuchteten Volumendatensätzen räumliche Strukturen besser wahrgenommen werden können.

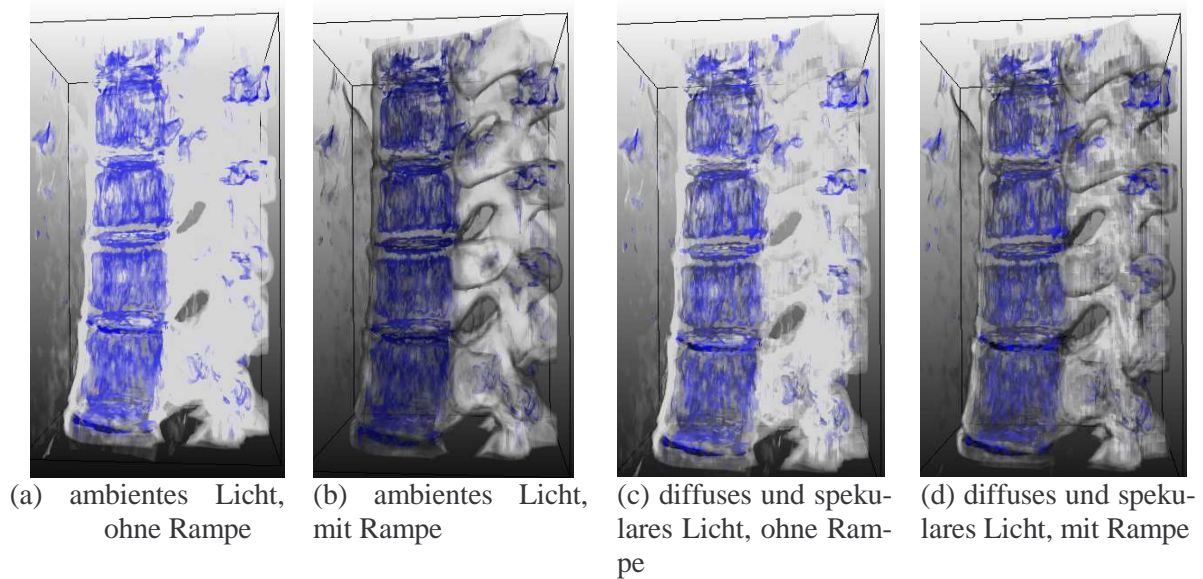


Abbildung 4-28: Beleuchtung von medizinischen Daten

In der Abbildung (a) sind kaum Strukturen wahrzunehmen. Dagegen sind in (b) Intensitätsunterschiede wahrnehmbar, die jedoch mit Beleuchtung (c+d) klarer hervortreten und die Wirbelsäule dreidimensionaler erscheinen lassen.

Jedem Bogen wird genau ein Farbwert (konstanter Wert oder Farbverlauf) zugeordnet. Dadurch erhält jede Kante eine einzige Farbe. Es wurde überlegt, dem Bogen mehrere Farben zuzuordnen. Aber dies führte nur zu einer Verschlechterung des Visualisierungsergebnisses. Im Gegenteil es kann zu einer Verwirrung des Benutzers führen. Getestet wurden u.a. zwei verschiedene Farben in einem Bogen. Dabei treten durch Interpolation der Farben jedoch Mischfarben auf, die zu einem bunten Ergebnisbild führen. Die Kanten können somit keinem spezifischen Materialübergang zugeordnet werden, was jedoch das eigentliche Visualisierungsziel sein soll: die Unterscheidung von verschiedenen Regionen in medizinischen Datensätzen. Aus diesem Grund wird jedem Bogen nur ein einzelner Farbwert zugeordnet.

4.5.4 Behandlung sich überschneidender Bögen

Es kann vorkommen, dass sich die Bögen in einigen Bereichen überschneiden. Bei der Erzeugung von LUTs muss dies Beachtung finden und dementsprechend darauf reagiert werden. Eine Möglichkeit besteht darin, die beiden TFs an dieser Stelle durch Mischung der Farb- und Transparenzwerte miteinander zu kombinieren. Dies ist jedoch aus den im vorherigen Abschnitt erwähnten Gründen ungeeignet.

Eine andere Lösung des Problems liegt darin, nur eine der beiden TFs im Schnittbereich der Bögen anzuwenden. Die Auswahl des Bogens, der im Schnittbereich für die Berechnung der LUT zur Anwendung kommen soll, kann auf unterschiedliche Weise realisiert werden. So kann er anhand des Zeitpunktes der Entstehung gewählt werden, d.h. es wird entweder derjenige Bogen gewählt, der als erstes oder der als letztes gezeichnet wurde. Mit diesen Einstellungen lassen sich gute LUTs erstellen. Jedoch scheint dies nicht ausreichend zu sein, da dadurch nicht genügend Wahlmöglichkeiten zur Verfügung stehen.

Eine andere Möglichkeit besteht darin, die Wahl des Bogens von dem Transparenzwert abhängig zu machen, d.h. es können Einstellungen so gewählt werden, dass entweder der Bogen benutzt wird, der eine hohe Transparenz besitzt oder der, der eine niedrige beinhaltet. Diese Einstellung ermöglicht bereits eine geeignete Auswahl zur Einstellung von gewünschten Visualisierungsparametern. Es besteht die Möglichkeit, Bögen mit gewünschten Transparenzen zu priorisieren.

Darüber hinaus besteht die Möglichkeit, den Bereich, in dem sich zwei Bögen überschneiden außer Acht zu lassen und diesen Bereich der LUT transparent zu zeichnen. Diese Art der Einstellung unterdrückt Informationen des Histogramms und scheint daher nicht gut geeignet zu sein. In mehreren Tests wurde festgestellt, dass dies durchaus wünschenswert sein kann. Denn gerade in dem Bereich in dem zwei Materialien aufeinander treffen, treten im Volumenrendering Artefakte auf. Diese werden in der Art deutlich, dass eine Kante zwei verschiedene Farbwerte aufweist und letzten Endes nicht das gewünschte Ergebnis dargestellt werden kann. Wird der Bereich ausgespart, so wird ein qualitativ gutes VR erzeugt.

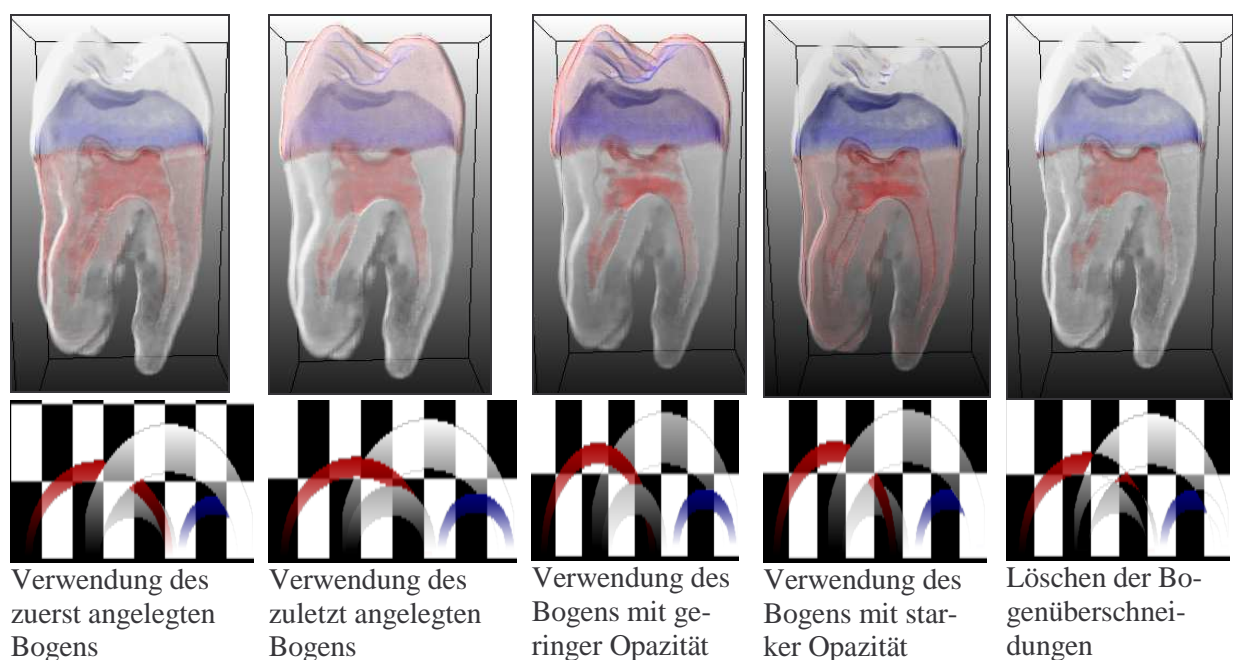


Abbildung 4-29: Mögliche Varianten bei Repräsentationsüberschneidungen

Abbildung 4-29 verdeutlicht die möglichen Varianten. Hier wird wiederum der Zahndatensatz verwendet, da die Unterschiede der einzelnen Varianten an diesem Beispiel stark hervortreten. Es ist erkennbar, dass der rote Bogen einen großen Einfluss auf das Gesamtbild nimmt. Dieser Bogen liegt in dem Bereich von zwei weiteren Bögen. Zum einen beeinflusst er den Bogen, der die äußere Zahnwurzel repräsentiert (kleiner weißer Bogen) und zum anderen den Bogen, der zur Darstellung des Zahnschmelzes (großer weißer Bogen) beiträgt. Diese Beeinflussung spiegelt sich in den ersten vier Bildern von links wieder. Auf eines der beiden Materialien nimmt der rote Bogen immer Einfluss. Erst in der letzten Möglichkeit, dem Löschen der Überschneidungen, wird dieses Problem behoben, und es entsteht die gewünschte Visualisierung - die Darstellung einzelner Kanten.

4.5.5 Zusammenfassung

In diesem Abschnitt wurde ein Konzept zum Entwurf von 2D-TFs vorgestellt. Zusammenfassend sind folgende Einstellungen möglich:

- Einstellen des Transparenzwertes
 - konstanter Transparenzwert
 - linear zunehmender Transparenzwert
 - Transparenzwert in einem festgelegten Intervall
- Einstellen des Farbwertes über verschiedene elementar stetig lineare Funktionen
 - Box: ein konstanter Farbwert für den gesamten Bogen
 - Dach: ein Farbwert, dessen Intensität ansteigt und anschließend wieder abfällt
 - Rampe: eine linear absteigende und eine ansteigende Rampe
- Festlegung eines Anwendungsbereiches
- Einstellen der TF an der Schnittstelle von Bögen

4.6 Interaktionskonzepte

In diesem Abschnitt werden Interaktionskonzepte für die in Abschnitt 4.5 entwickelte Repräsentation vorgestellt. Benutzerfreundlichkeit und Effizienz der Bedienung sollen hierbei im Vordergrund stehen. Das Ziel einer guten Softwareentwicklung liegt darin, dem Benutzer die Möglichkeit zu bieten, die Aufgaben in angemessener Zeit zu erledigen und während der Durchführung möglichst Bedienfehler zu vermeiden.

4.6.1 Parametrierung der Repräsentation

Als erstes wird die Parametrierung der Bögen festgelegt. Dafür soll ein HE-förmiges Objekt angelegt werden. Dies könnte mithilfe von zwei Radien für den Gradienten, zwei Radien für das Intensitätsintervall und einer Position realisiert werden. Der Benutzer müsste dafür das Histogramm auswerten und die einzustellenden Parameter ablesen und anschließend angeben.

Eine andere Möglichkeit besteht darin, für die Grauwerte und den Gradienten jeweils ein Zentrum und eine Breite anzugeben. Aus diesen Parametern könnte dann ein Ellipsenbogen berechnet werden.

Diese beiden Methoden sind zu umständlich für den Benutzer. Daher wurde überlegt, eine direkte Interaktion mit dem Histogramm zu schaffen. Der Benutzer betrachtet das

Histogramm und es soll ihm ermöglicht werden, eine Ellipse direkt darüber zu zeichnen. Die Maus ist für einfache Zeichnungen ein gutes Werkzeug, jedoch ist das Zeichnen einer Ellipse mit der Maus eher weniger gut, da der Benutzer den Bogen nicht rund oder nur sehr schlecht approximiert. Die Zeichnung einer Ellipse lässt sich bereits auf Papier mit einem Stift schwer lösen. Daher wurde nach einer besseren Interaktionsmöglichkeit gesucht.

Eine HE kann durch zwei Parameter berechnet werden. Dafür ist die Positionierung eines Fußpunktes und eines Extrempunktes erforderlich. Mit der Maus kann der Benutzer diese beiden Punkte im Histogramm setzen. Beim Zeichnen der Ellipse wurde das Rubberbanding-Verfahren benutzt. Dieses ist gut geeignet, um Graphikprimitive wie Geraden, Vierecke, Kreise und Ellipsen interaktiv zu erstellen. Dazu spezifiziert der Benutzer durch Drücken der linken Maustaste als erstes den Extrempunkt der HE. Mit gedrückter Maustaste wird die Maus bewegt, wobei veranschaulicht wird, wie die HE beim Loslassen der Maustaste aussehen würde. Wird die Maustaste losgelassen, so wird die HE erstellt, siehe Abbildung 4-30.

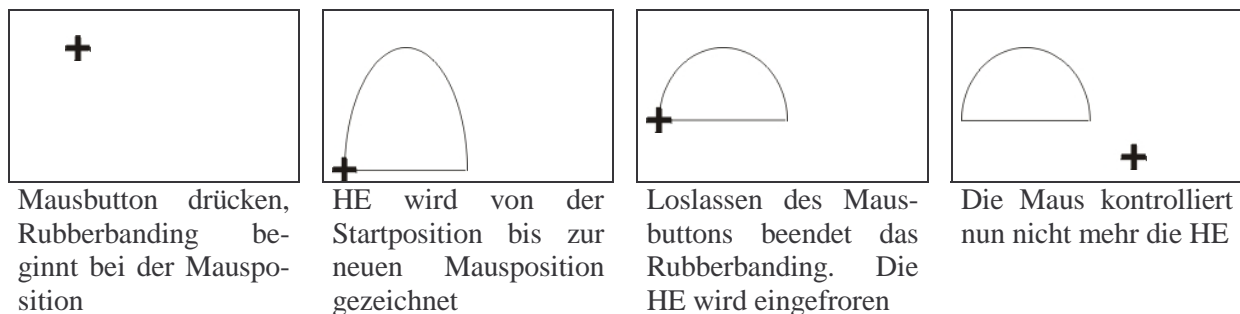


Abbildung 4-30: Prinzip des Rubberbandings für HE

Die Definition der HE kann so festgelegt werden. Danach werden noch zwei Abstände, (um HE-förmige Bogenflächen zu erzeugen, siehe Abschnitt 4.5.1, Abbildung 4-23) angegeben und die Repräsentation der TFs ist realisiert. Um die Abstände festzulegen, muss der Benutzer eine Breite (Abstand an den Fußpunkten) und eine Höhe angeben (Abstand an der Spitze), was über ein editierbares Textfeld oder über das Scrollrad der Maus gelöst werden kann. Diese Abstände können auch relativ über die Höhe und Breite berechnet werden, so dass der Benutzer entweder gar keine Angaben zu diesen Abständen oder nur geringe Änderungen vornehmen muss.

Für die Modifikation der Ellipse werden der Extrem- und der Fußpunkt mit Markern versehen. Diese können mit der Maus selektiert und jederzeit verschoben werden, siehe Abbildung 4-31.

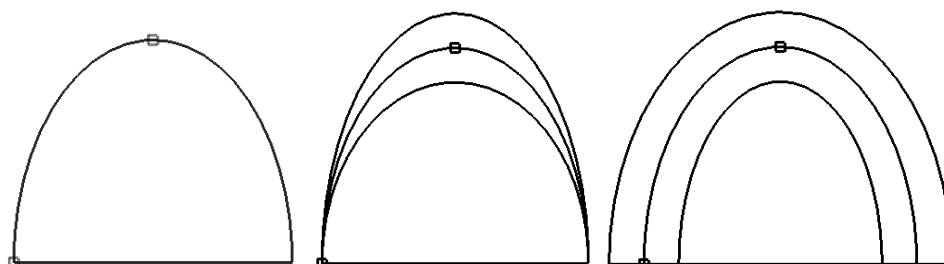


Abbildung 4-31: HE definiert durch zwei Punkte „□“ und zwei Abstände
Links: beide Abstände = 0, Mitte: Abstand am Fußpunkt = 0 und ein Abstand > 0, Rechts: zwei gleich große Abstände > 0

4.6.2 Transparenzen und Farbwerte

Nachdem die Bogenform definiert wurde, müssen im nächsten Schritt jedem Bogen Transparenzen und Farbwerte zugewiesen werden. Wird nur mit Grauwerten gearbeitet, so besteht eine Möglichkeit darin, an der unteren Seite des Histogramms und an der linken Seite eine Linie darzustellen, wobei diese Linien die Rampen für die Intensitäten und die Transparenzen repräsentieren. Durch Kontrollpunkte kann der Benutzer diese Rampen ändern und dadurch die TFs spezifizieren, Abbildung 4-32. In [Kin03] wurde dieses Konzept verwendet, welches bei der Kombination von zwei eindimensionalen TFs sicher gut geeignet ist, jedoch bei dem hier vorgestellten Konzept auf unintuitive Weise mit der vorgestellten Repräsentation verknüpft werden müsste.

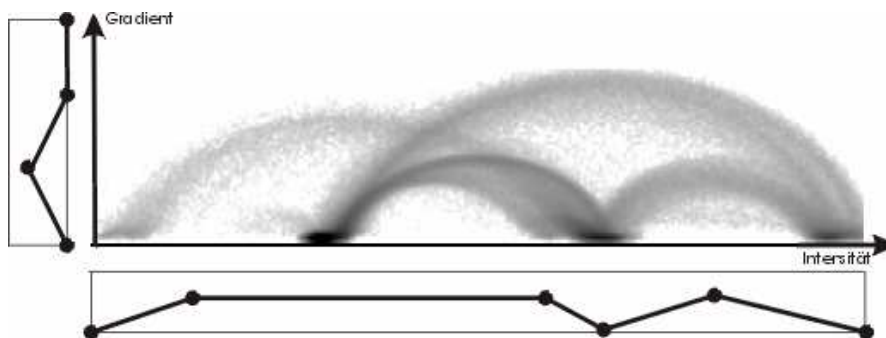


Abbildung 4-32: Interaktion mittels Rampen am Beispiel des Zahndatensatzes

Eine andere Art, die Farben und Opazitäten zu spezifizieren, besteht darin, einen Farb-/Grauwert über eine Palette auszuwählen. Der über dem Histogramm gezeichnete Bogen wird ebenfalls in dem Farb-/Grauwert der TF dargestellt. Die Opazität kann mittels eines editierbaren Textfeldes eingestellt werden. Im nächsten Schritt müssen Farb-/Grauwert- und Opazitätsverteilungen festgelegt werden. Dies erfolgt mittels verschiedener Rampen, die in einem Menü angeboten werden. In diesem Menü kann der Benutzer den gewünschten Farb-/Opazitätsverlauf für einen Bogen auswählen, worauf die Verteilungen automatisch berechnet werden. Die Kontrollpunkte der Rampe ergeben sich aus der Ellipsenrepräsentation. Dabei wird die Rampe für die Transparenz über die Höhe der HE realisiert und die Rampen für die Farb-/Grauwerte werden über die Breite und über die Mitte (nur für die Dachfunktion relevant) festgelegt.

Auf diese Weise können schnell Farb- und Opazitätsverläufe angelegt werden. Auch sind verschiedene Voreinstellungen denkbar, so dass der Benutzer nach dem Anlegen der Repräsentation erste Ergebnisse sehen kann.

4.6.3 Unterstützung der TF-Definition

King stellt in seiner Arbeit ein Modul vor, das dem Benutzer die Histogrammanalyse erleichtern soll. Dabei legt er Boxen im Histogramm an und in der Schichtdatendarstellung werden anschließend, die korrespondierenden Voxel, der von diesen Boxen eingeschlossenen Bins, gefärbt.

Diese Funktion kann leicht für die Bögen übernommen werden, indem die LUT berechnet und anschließend eine Schichtdarstellung des Datensatzes in der Farbe der LUT erzeugt wird.

So kann der Benutzer im Detail erkennen, was von den LUT in welcher Farbe gefärbt wird, siehe Abbildung 4-33.

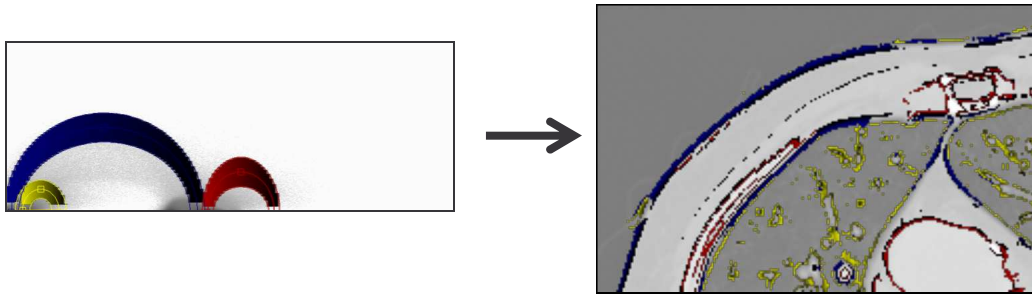


Abbildung 4-33: LUT angewendet auf die Schichtdatendarstellung

Zur Datenexploration ist dies jedoch nicht gut geeignet bzw. nicht ausreichend. Der Benutzer möchte vielmehr die Kanten im Datensatz markieren, die er im VR dargestellt haben möchte.

Kantendetektion einzelner Übergänge

In der Arbeit von Joe Kniss wurde ein DataProbeWidget vorgestellt, wodurch mittels eines Applikators dem Benutzer die Möglichkeit gegeben wird, den 3D-Datensatz zu explorieren. Dies soll auch hier wieder aufgegriffen werden, jedoch ist das Selektieren von Bereichen in einem 3D-Volumen teilweise kompliziert, da die Darstellung des Volumens auf dem Bildschirm nicht überlagerungsfrei erfolgt und die Tiefe des Volumens nur grob abgeschätzt werden kann. Mithilfe der Maus und der Projektion des Volumens auf den Bildschirm lässt sich nur schwer die richtige 3D-Position finden.

Daher wurde dieses Verfahren etwas vereinfacht, indem der Datensatz in der Schichtdatenvisualisierung erkundet werden kann. So kann genau ein einzelnes Voxel selektiert werden und durch entsprechende Markierungen des Bins im Histogramm wird die Intensitäts-Gradienten-Kombination dieses Voxels sichtbar, siehe Abbildung 4-34. Als Zusatzinformation kann die Worldkoordinate des Bins (Intensitäts- und Gradientenwert des Voxels) angezeigt werden.

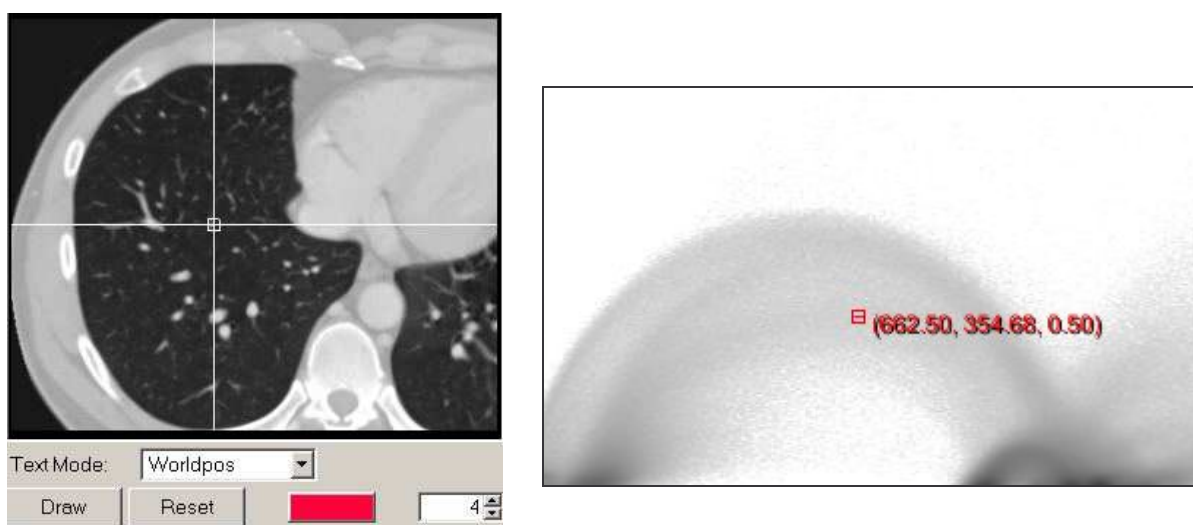


Abbildung 4-34: Selektion eines Voxels (links) und Anzeige des korrespondierenden Bins im Histogramm (rechts). Der Intensitätswert des Voxels, ablesbar an dem markierten Bin, beträgt 662 und der Gradientenwert ist 354

Dieses Interaktionskonzept wurde dahingehend erweitert, dass mehrere Voxel gleichzeitig exploriert werden können, d.h. die Markierungen bleiben im Histogramm solange sichtbar bis der Benutzer festlegt, dass alle Markierungen gelöscht werden sollen. Auf diese Art und Weise lassen sich mehrere Voxel entlang einer Kante, und somit die Kante selbst, erkunden. Der Benutzer untersucht die Voxel entlang einer Kante, d.h. den Übergang von einem Gewebe in ein anderes. Dabei zeichnet sich im Histogramm ein Bogen ab. Für diesen Bogen kann dann eine entsprechende Repräsentation angelegt werden. Im VR wird dann die erkundete Kante sichtbar, siehe Abbildung 4-35.

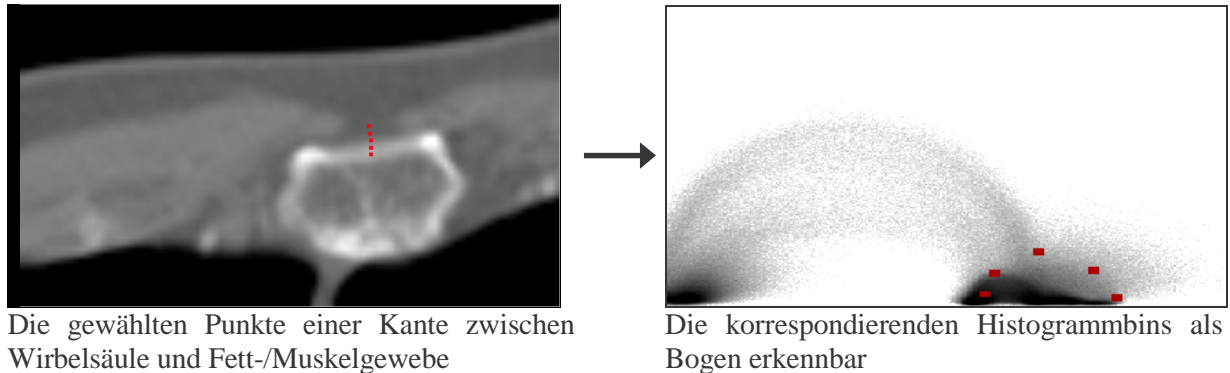


Abbildung 4-35: Unterstützung der Kantensuche

Die Bedeutung dieser Komponente ist besonders für unerfahrene Benutzer von Vorteil. Mitunter ist es schwer, zu einer gewünschten Kante die korrespondierenden Bins zu finden. Zum einen muss der Benutzer wissen, welche beiden Materialien an die Kante angrenzen und zum anderen muss er bei CT-Aufnahmen wissen, welche HU-Werte bei diesen beiden Materialien vorliegen. Den Gradienten kann er nur schwer abschätzen. Daher ist es teilweise recht aufwändig eine TF zu definieren, da viel mittels Trial-and-Error ausprobiert werden muss. Mit der Kantendetektionskomponente kann dieser Aufwand reduziert werden.

Ein Beispiel

Als nützlich hat sich die Kantendetektionskomponente bei der Untersuchung von kontrastierten Leberdatensätzen erwiesen. Hier liegt das Interesse besonders in der Visualisierung der Gefäße. Da der Kontrast zwischen Leber und Gefäßen jedoch gering ist (Kontrast von ca. 100-120 HU), ist der Bogen im Gegensatz zu anderen nur winzig klein und nur schlecht im Histogramm zu finden. Auch liegen die Intensitätswerte wegen der Kontrastmittelgabe von Datensatz zu Datensatz in verschiedenen Intensitätsbereichen.

Die Nützlichkeit dieser Komponente wird an dem folgendem Beispiel verdeutlicht. In diesem Beispiel liegt ein weich-rekonstruierter CT-Datensatz mit einer kontrastierten Leber vor. Mit der Kantendetektionskomponente wurde eine Kante zwischen der Leber und den Gefäßen untersucht. Abbildung 4-36 stellt mögliche Bereiche entlang der Kante zwischen Leber und einem Lebergefäß dar, die für die Erkennung gut geeignet sind. Entlang dieser Bereiche werden Voxel des Datensatzes selektiert. Abbildung 4-37 stellt die mit den selektierten Voxeln korrespondierenden Bins im Histogramm dar. Zu sehen ist, dass die berechneten Bins bogenförmig angeordnet sind. In Abbildung 4-38 ist die Definition der HE-förmigen Bogenfläche dargestellt. In Abbildung 4-39 ist schließlich das Volumenrendering der Lebergefäße visualisiert.

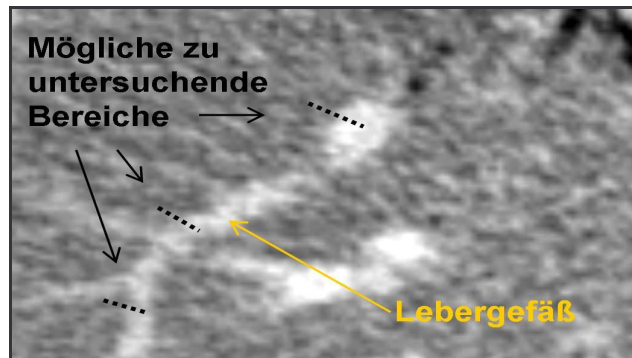


Abbildung 4-36: Mögliche Bereiche zur Untersuchung von Kanten zwischen Leber und Lebergefäßen in einem CT-Datensatz

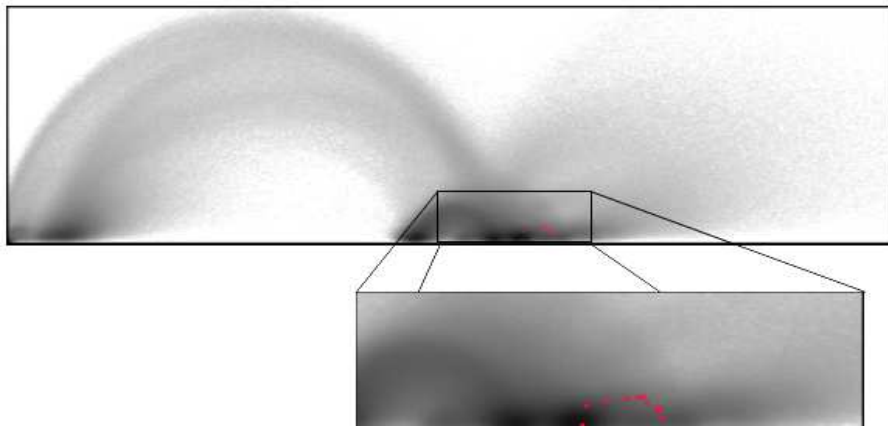


Abbildung 4-37: Histogramm eines Leberdatensatzes und gefundener Gefäßbogen

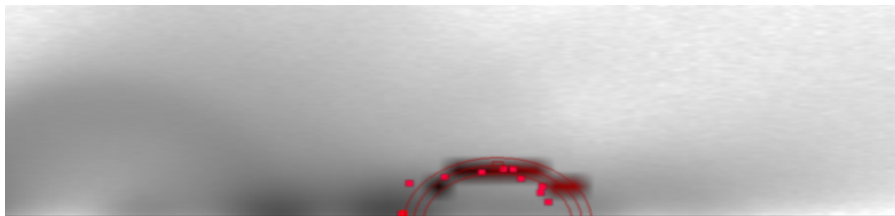


Abbildung 4-38: Repräsentationsform entlang des ermittelten Bogens

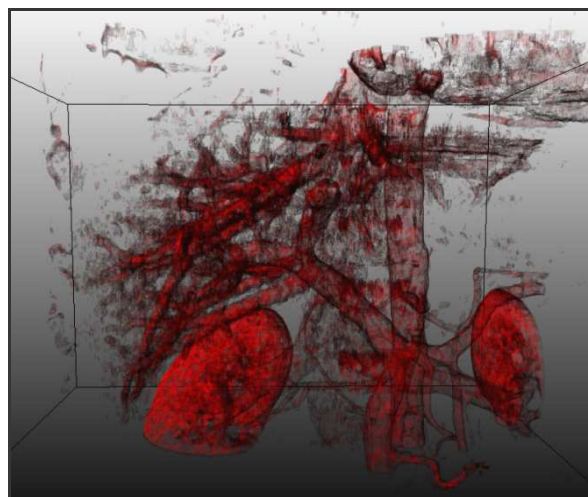


Abbildung 4-39: Gefäßvisualisierung im VR mit einer gradientenabhängigen TF

Kantendetektion mittels einer Segmentierungsmaske

Gerade für medizinische Daten liegen bei MeVis oft segmentierte Daten für die Therapieplanung vor. Das Ziel ist es, die Kanten von segmentierten Objekten im Histogramm darzustellen. Dazu werden die Masken so bearbeitet, dass nur der Rand dieser Masken (sozusagen die Kanten) zur Berechnung genutzt wird. Anhand dieser Kantenmaske werden alle zu der Maske korrespondierenden Histogrammpunkte berechnet und in diesem abgetragen. Der Benutzer kann die zu der Segmentierung gehörenden Intensitäts-Gradienten-Kombinationen erkennen und geeignete TFs definieren. Diese Komponente ist nicht so mächtig wie die zuvor vorgestellte. In dem Histogramm aus Abbildung 4-40 ist zu sehen, dass die Maske viele Grauwert-Gradienten-Kombinationen umfasst. Dies liegt daran, dass der genaue Bereich der Maske nicht festgelegt werden kann. Diese Komponente ist nur zur groben Orientierung geeignet.

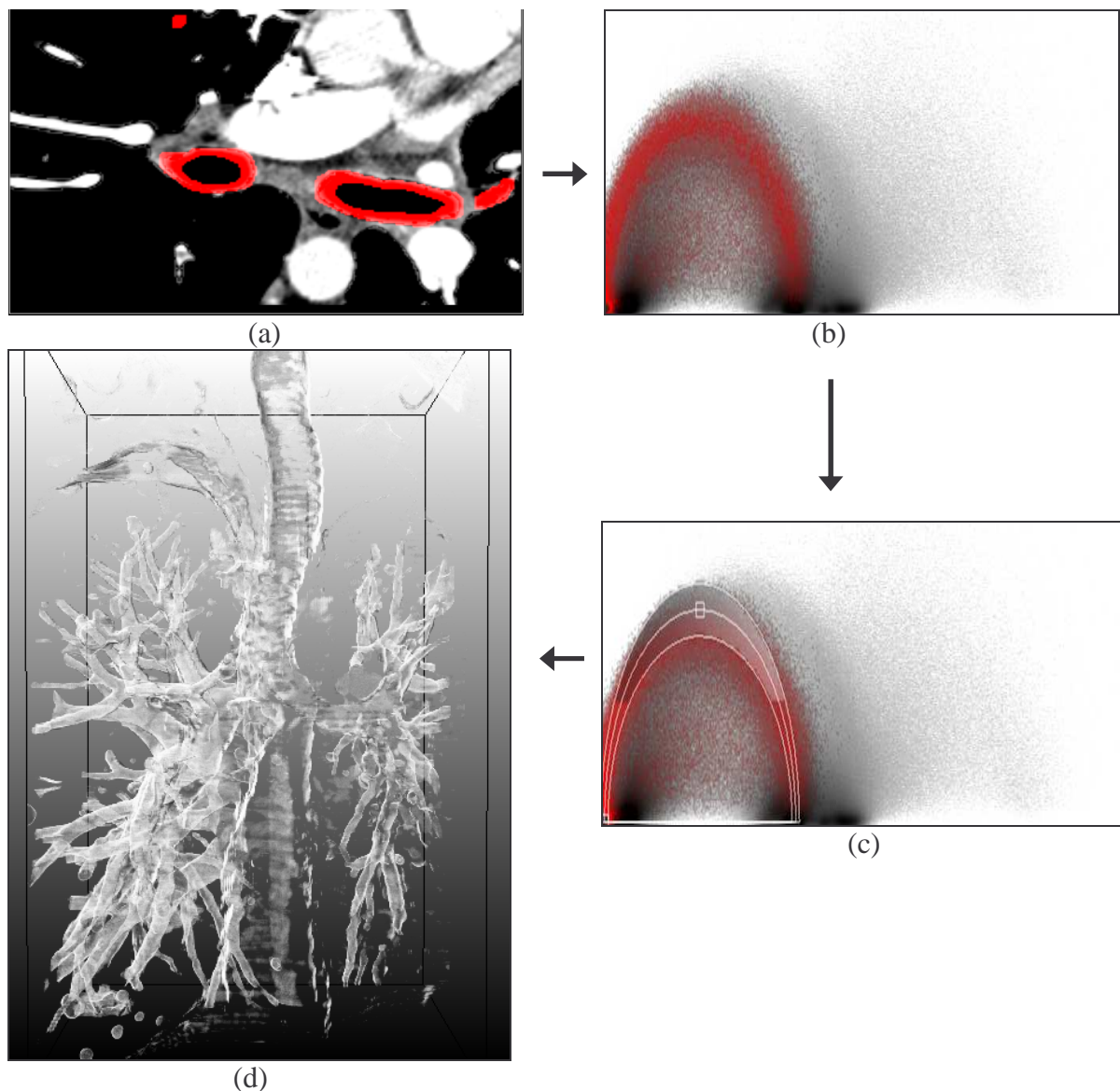


Abbildung 4-40: TFs-Definitionsprozess anhand einer Kantenmaske.

Maske der Bronchien (a) und die dazugehörigen Bins rot gefärbt im Histogramm (b). Auf den am häufigsten auftretenden Voxelkombinationen wird eine TF definiert (c) und das Ergebnis im VR dargestellt (d).

4.7 Automatische Suche/Parametrierung der Bögen

Um dem Benutzer das Einstellen der Repräsentationsparameter abzunehmen, wurden Überlegungen zur automatischen Suche der Bögen vorgenommen. Dabei wurden zuerst Algorithmen zur automatischen Ellipsensegmentierung analysiert. Anschließend wurden in Polarkoordinaten umgewandelte Histogramme daraufhin untersucht, ob damit eine einfachere automatische Suche der Bögen möglich ist. Als letztes wurde ein Algorithmus zur automatischen Suche von Bögen entwickelt. Dabei wurden zuerst alle gegebenen Eigenschaften der Bögen analysiert, um dann bekanntes Vorwissen über den Verlauf der Bögen auszunutzen.

4.7.1 Analyse vorhandener Suchalgorithmen für Ellipsen

Bei der Suche nach bereits vorhandenen Algorithmen zur Ellipsendetektion muss beachtet werden, dass die Algorithmen auf stark verrauschten Bildern arbeiten können und dass die Ellipsen nicht vollständig sind, da nach Habellipsen gesucht wird.

In [Mai02] wird ein genetischer Suchalgorithmus für Ellipsen vorgestellt. Genetische Algorithmen haben den Vorteil, dass sie den Speicher wenig belasten, jedoch sind lange Rechenzeiten wegen hoher Anzahl an Iterationen zu erwarten. Die nichtdeterministische Rechenzeit und die Möglichkeit, dass am Ende nicht das beste Ergebnis erreicht wird, sprechen ebenfalls nicht für diese Art von Algorithmen. Tomás Mainzer testet seinen Algorithmus an verschiedenen Bildern (ohne Rauschen, verrauscht und stark verrauscht), die 256^2 Pixel groß sind und in denen eine einzelne Ellipse vorhanden ist. Leider analysiert er im Ergebnis nur die Anzahl der Durchläufe des Algorithmus'. Wie genau die Ergebnisse sind, wird nicht erklärt, dabei wäre das bei verrauschten Daten wesentlich. Es wird gezeigt, dass der Algorithmus bei verrauschten Daten rechenintensiv ist. Da die Histogramme ein starkes Rauschen aufweisen, wäre der Rechenaufwand dieses Algorithmus' sehr hoch.

Ein auf der Hough-Transformation beruhender Algorithmus wird in [AN95] vorgestellt. Die Hough-Transformation findet oft Verwendung bei der Detektion von Geraden kann aber auch bei der Suche nach Ellipsen eingesetzt werden.

Das Ursprungsbild mit der Ellipse wird in einen Parameterraum transformiert. Liegen Punkte im Ursprungsbild auf einer Ellipse, so treffen sich korrespondierende Kurven im Hough-Raum in einem Punkt. Durch Auslesen der Parameter dieser Kurven können die Parameter der Ellipse bestimmt werden. Diese Hough-Transformation findet bei [AN95] Anwendung. Mithilfe des darin vorgestellten Algorithmus' wurden Ellipsen in synthetischen und realen Bildern gefunden. Zum Rauschen in den Bildern wurden keine Untersuchungen vorgenommen. Probleme die bei der Hough-Transformation auftreten, sind zum einem hoher Speicherbedarf und zum anderen nimmt die Genauigkeit mit der Anzahl der Ellipsen ab.

In [KHK02] wird ein Algorithmus vorgestellt, der schnell und robust in komplizierten (auch verrauschten) Bildern Ellipsen findet. Der Algorithmus besticht durch eine schnelle Berechnung, die erreicht wird durch die Zerlegung der Ellipse in einzelne Segmente. Kleine Segmente können schnell gesucht werden. Die Segmente werden in acht Gruppen eingeteilt, d.h. jeweils in ein Achtel der Ellipse. Sind die Ellipsenparameter der Segmente und der Ellipsentangentenvektor gleich der Segmenttangente an Start- bzw. Endpunkt, so ist eine Ellipse gefunden.

Die von Kim et al. getesteten Bilder zeigen, dass mit diesem Algorithmus gute Ergebnisse erzielt werden können. Dieser Algorithmus kann sogar HEs finden, und es können Störungen im Bild auftauchen, die das Ergebnis nicht beeinflussen.

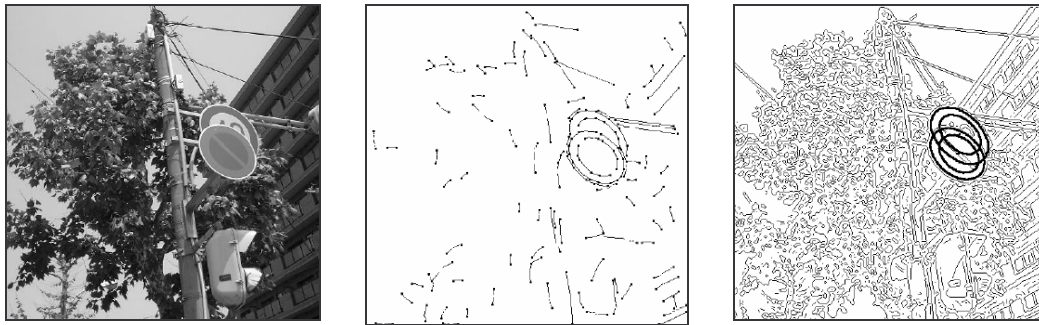


Abbildung 4-41: Ellipsendetektion nach Kim
 Links: Originalbild, Mitte: detektierte Segmente,
 Rechts: gefundene Ellipsen

Der von Kim et al. vorgestellte Algorithmus scheint gut geeignet zu sein. Er ist schnell und findet selbst bei Störungen im Bild Ellipsen. Leider liegen bei den hier verwendeten Histogrammen, Störungen und Rauschen in der Art vor, dass Ellipsen an Stellen gefunden würden, wo sich keine Bögen befinden. Es würden viele Ellipsen gefunden werden, obwohl nur eine begrenzte Anzahl nötig ist. Ein weiteres Problem liegt in der Annahme, dass die Bögen ellipsenförmig sind. In den vorherigen Abschnitten wurde gezeigt, dass die Bögen den Ellipsen sehr ähnlich sind, jedoch ist dies nicht die genaue Form der Bögen. Daher wurden die Algorithmen zur Ellipsenerkennung nicht weiter betrachtet.

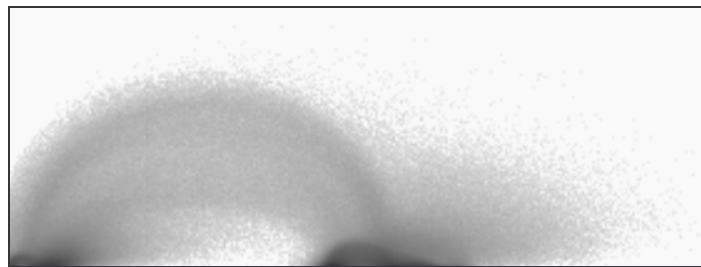


Abbildung 4-42: Ein stark verrauschtes Histogramm

4.7.2 Abtragen der Intensitäts- und Gradientenwerte auf Polarkoordinaten

In Polarkoordinatensystemen wird die Position eines Punktes über einen Abstand zu einem festgelegten Koordinatenursprung und durch einen Winkel angegeben.

Für die hier vorliegenden Datensätze werden die Intensitätswerte auf einen Radius und die Gradienten auf einen Winkel abgebildet. Die Gradienten werden auf ein Winkelintervall abgebildet. Es kann beispielsweise eingestellt werden, dass alle Gradienten auf ein Winkelintervall von $0-180^\circ$ abgetragen werden. Die Idee ist, den Gradienten auf einen Bereichswinkel so abzutragen, dass aus den erkennbaren Bögen des Intensitäts-Gradienten-Histogramms Geraden werden.

An mehreren Datensätzen, u.a. auch Datensätze in denen Bögen klar voneinander abgegrenzt hervortreten, wurde getestet, ob sich diese Geraden herauskristallisieren. Leider musste nach mehreren Testläufen erkannt werden, dass dies nicht der Fall ist. Geraden entstehen, jedoch werden nur einige Bögen auf eine Gerade abgebildet. Es ist nicht möglich, alle Bögen gleichzeitig auf Geraden abzubilden. In Abbildung 4-43 wird dies deutlich.

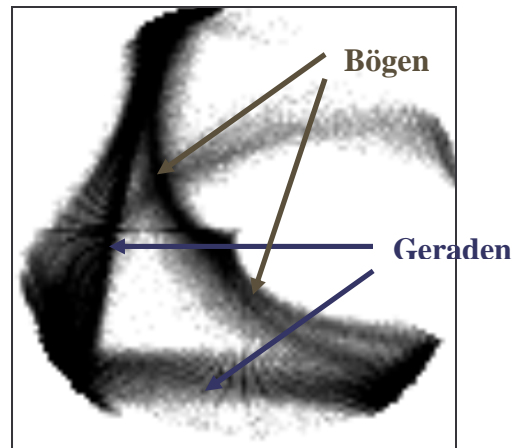


Abbildung 4-43: Gradientenstärke und Grauwerte als Polarkoordinaten abgetragen

4.7.3 Ein auf Apriori-Wissen basierender Algorithmus

Da durch Ellipsensuchalgorithmen und mit Hilfe der Polarkoordinaten kein automatisches Auffinden der Bögen möglich ist, ist die Idee des in diesem Abschnitt behandelten Algorithmus, das vorhandene Wissen über den Verlauf der Bögen auszunutzen. Dafür werden zuerst Eigenschaften der Bögen des Histogrammbildes zusammengefasst.

Ausgehend von den generierten Histogrammen werden folgende Hypothesen aufgestellt:

- Die Bögen verlaufen dort, wo die stärksten Intensitäten im Histogramm auftreten.
- Die Bögen beginnen im unteren Teil des Histogrammbildes.
- In der ersten Zeile gibt es Häufungspunkte.
- Dem Verlauf der x-Achse (zunehmende Intensitäten) folgend steigen alle Bögen kontinuierlich in x-Richtung an. Es kann in einem Winkel von höchstens 90° zu der x-Achse gesucht werden.
- Der Winkel der Bögen nimmt während seines Verlaufs in x-Richtung ab.
- Beim Erreichen des Extrempunktes beträgt der Winkel 0° . Würde der Bogen weiter verfolgt werden, so würde mit einem immer kleiner werdenden negativen Winkel gearbeitet werden müssen.
- Wird der Bogen in mehrere Segmente zerlegt, so bleibt der Winkel von einem Segment zum nächsten entweder gleich groß oder er nimmt ab.
- Es ist ausreichend nur die eine Hälfte des Bogens zu suchen, da der Bogen auf der anderen Seite durch dessen Symmetrie ergänzt werden kann.

Algorithmusskizze:

- Als erstes werden die Startpunkte durch Suchen von Häufungspunkten in den ersten Zeilen ermittelt (a+b+c).
- Im nächsten Schritt werden um die Startpunkte jeweils eine Box, mit einer fest definierten Länge und Breite, gelegt.
- Alle Intensitätswerte der Pixel des Histogrammbildes, die in der Box liegen, werden aufaddiert. Die Summe wird dividiert durch die Anzahl der in der Box enthaltenen Pixel.

4. Die Box wird in einer festgelegten Schrittweite um den Startpunkt bis zu 90° gedreht. Der Winkel, in dem die Summe der Intensitätshäufigkeiten am größten ist, ist der Winkel, in dem das nächste Segment geprüft wird (d).
5. Das nächste Segment des Bogens wird in der Richtung des Winkels mit der höchsten Summe gesucht. Die neue Startposition ergibt sich aus dem Startpunkt und der Addition der Länge der Box.
6. Der ermittelte Winkel wird gespeichert. Von der neuen Startposition aus, wird in dem gleichen Winkel (mit einer kleinen Abweichung (h)) weitergesucht (d).
7. Die Schritte 4 bis 6 werden solange wiederholt, bis der Winkel 0° erreicht hat (f). Dann ist die Spitze des Bogens erreicht und der Bogen kann durch Symmetrie vervollständigt werden (h).

Eine mögliche Erweiterung ist das Suchen von Startpunkten im gesamten Histogramm. Dabei wird das Histogramm in einzelne Boxen zerlegt. In jeder Box wird nach der größten Intensität gesucht. Dies ist der Startpunkt. Auf diese Art und Weise werden von vornherein mehr Punkte auf einem Bogen gefunden. Von diesem Startpunkt aus, werden die Bögen verfolgt, wobei dies in zwei Richtungen erfolgt. Einmal sozusagen nach oben, d.h. entlang der steigenden Intensitäten in einem Winkel von höchstens 90° und einmal nach unten entlang der fallenden Intensitäten in einem Winkel von 180° bis 270° .

Einzustellende Eigenschaften bzw. Parameter:

- Schrittweite des Winkels, d.h. um wie viel Grad soll die Box im nächsten Schritt gedreht werden
- Boxbreite
- Boxlänge

Pseudocode:

```

int boxwidth, boxlength, nextAlpha, alpha=0, alphaStep=2, x, y=2, z=0
float threshold = 3
float sum
box box
vector startPoint[], nextPoint, extremPoint

for x = 0 until x = sizeof(hist(x))
  do
    //suche nach Intensitätshäufigkeiten
    betrachte den Intensitätswert an der Position x,y,z im Histogramm
    if (Intensitätswert >= threshold) then
      speichere Position (x,y,z) in startPoint
    endif
    x = x+1
  endfor
for loop = 0 until loop = sizeof(startPoint)-1
  do
    while nextAlpha != 0
      do int count = 0
        for alpha = 0 until alpha = 90
          do //lege Box um startPoint(loop)
            //box besteht aus 2 Eckpunkten (v1, v2)
            box.v1 = (startPoint.x, startPoint.y-boxwidth/2,z)
            box.v2 = (startPoint.x+boxlength, startPoint.y+boxwidth/2,z)
            sum[count] = Summe der Intensitäten aller in der Box enthaltenen
                          Punkte/(Anzahl der getroffenen Punkte)
            alpha = alpha + alphaStep
          enddo
        endfor
      enddo
    nextAlpha = nextAlpha - alphaStep
  enddo

```

```

    count = count + 1
  endwhile
  int index = index an der Stelle, an der sum = maximal
  nextAlpha = index*alphaStep
  extremPoint ergibt sich aus dem Vektor der Länge der Box in Richtung
  von nextAlpha
  startPoint = extremPoint
endwhile
speichere extremPoint /*ein vollständiger Bogen*/
loop = loop+1
endfor

```

Ergebnisse

Der Algorithmus wurde implementiert und getestet. Die Einstellung der Parameter ist abhängig von der Auflösung des Histogramms. Je nachdem wie die Größe des Histogramms eingestellt wurde, mussten entsprechende Änderungen an der Größe der Box und an der Schrittweite des Winkels vorgenommen werden. Bei synthetischen Volumendatensätzen und geeigneter Parametrierung des Algorithmus' konnten die Bögen automatisch gefunden werden. Die Resultate waren verwertbar und führten zu einer geeigneten Visualisierung.

Ein Beispiel: Datensatz eines Zahnes mit klar abgegrenzten Bögen

Das Histogramm wurde mit einer kleinen BinSize generiert, hier wurde BinSize = 1 gewählt. Daraus ergibt sich, dass ein Intensitätswert und ein Gradientenwert in einem Bin abgetragen werden. Der Algorithmus sollte vier Bögen finden. Die gefundenen vier Bögen liegen fast auf den erkennbaren Bögen wie in Abbildung 4-44 rechts zu sehen ist. Aus dieser gefundenen Repräsentation wurde eine LUT erzeugt (siehe Abbildung 4-44 links), die ein gutes Ergebnis lieferte.

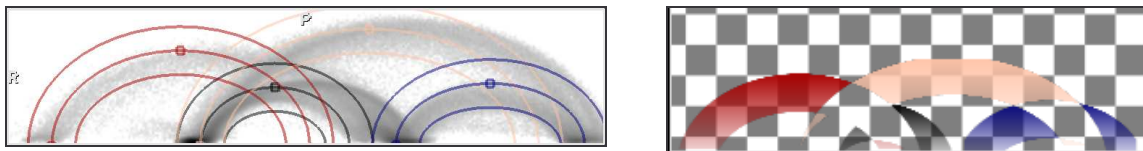


Abbildung 4-44: Gefundene Repräsentation und erzeugte LUT



Abbildung 4-45: Automatisch generierte TFs für einen Zahn

Hier ist zu sehen, dass der Zahn sehr gut visualisiert wurde. Dieses Ergebnis lässt sich jedoch nur auf synthetisch generierten Datensätzen erzielen, da hier klare Objektgrenzen im Datensatz und damit Bögen im Histogramm vorhanden sind. In echten medizinischen Datensätzen ist viel Rauschen, so dass ähnliche Ergebnisse mit diesem Algorithmus nicht erreicht werden können.

Probleme:

Der Algorithmus wurde darüber hinaus auf medizinischen Datensätzen getestet. Aufgrund der verrauschten Histogramme konnten jedoch keine guten Ergebnisse erzielt werden. Die Hervorhebung der Bögen im Histogramm ist nur gering. Das menschliche Auge kann kleine Differenzen wahrnehmen, jedoch ist die Entwicklung eines Algorithmus' für diese feinen Unterschiede schwer. In dem Histogramm ist zu viel Rauschen, wobei die visuelle Wahrnehmung des Menschen in der Lage ist, dieses Rauschen zu unterdrücken. Die menschliche Wahrnehmung ist zusätzlich darauf ausgelegt, Strukturen zu suchen und zu erkennen. Sind in den Bögen Brüche werden sie vom menschlichen Auge nicht als Unterbrechungen wahrgenommen, sondern der Mensch nimmt den Bogen als ganzes wahr. Die Gesamtheit der Struktur wird erkannt. Der Algorithmus müsste somit um diese Eigenschaft, auch nach Unterbrechungen weiter zu suchen, erweitert werden.

Außerdem ist eine geeignete Suche zu stark von der Histogrammerzeugung abhängig. Je nachdem, mit welcher Größe das Histogramm aufgelöst ist, müssen die Parameter verschieden eingestellt werden. Es muss beachtet werden, dass unterschiedlich große Histogramme in der Qualität voneinander abweichen. Je nachdem, wie die Binsize in x- und in y-Richtung eingestellt ist, entstehen unterschiedliche Histogramme. Wird die Binsize sehr groß gewählt, werden viele Intensitäts-Gradientenkombinationen auf ein und dasselbe Bin abgebildet. Wird sie sehr klein gewählt, so entsprechen nur wenige Kombinationen einem Bin. An dem aus dem Zahndatensatz erzeugten Histogramm kann dies verdeutlicht werden. In Abbildung 4-46 oben ist zu erkennen, dass bei großer Binsize die Bögen eine geschlossene Form annehmen, jedoch sehen die Bögen auch teilweise „verschwommen“ aus. In Abbildung 4-46 mitte sind die Bögen etwas klarer. In dem unteren Bild ist zu erkennen, dass die geschlossene Form der Bögen verloren geht, jedoch wird bei kleiner Binsize die Verteilung der Intensitäts-Gradienten-Kombinationen am genauesten charakterisiert.

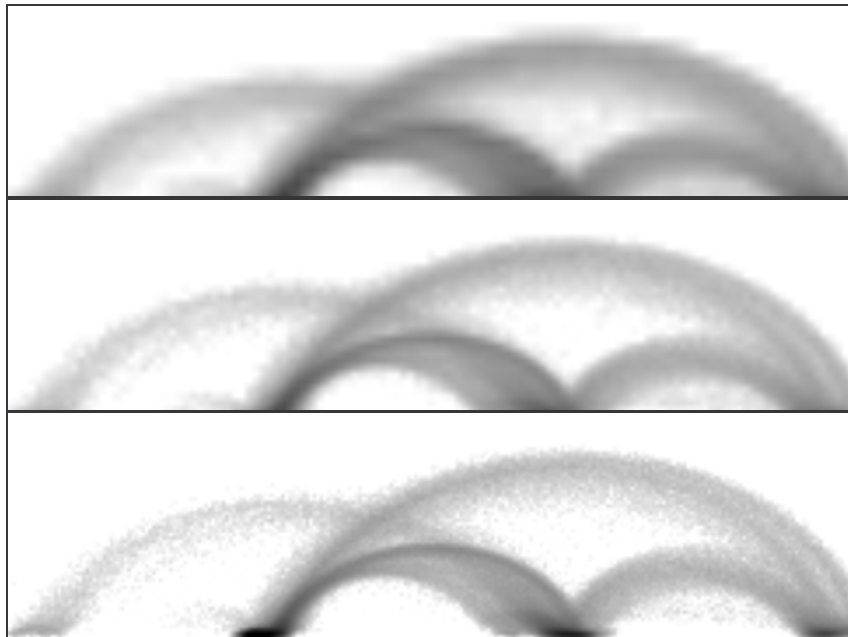


Abbildung 4-46: Histogramme mit unterschiedlicher Binsize
Oben: große Binsize (=20), Mitte: mittlere Binsize (=10), Unten: kleine Binsize (=1)

Beispiel für einen medizinischen Datensatz

Hierfür wurde ein Abdomendatensatz mit einer kontrastierten Leber betrachtet. Die Größe des Datensatzes beträgt $365 \times 310 \times 154$ mit einer Voxelgröße von $0.625 \times 0.625 \times 1.0$. Das Histogramm wurde mit einer kleinen Binsize generiert, um auf eine möglichst genaue Verteilung der Dateninformationen zurückzugreifen. Ein Bogen, der mittels dieses Algorithmus' gut gefunden wurde, ist in Abbildung 4-47 (a) dargestellt. Die anderen relevanten Bögen konnten nicht gefunden werden, da beispielsweise in dem großen Bogen zu viele Unterbrechungen sind. Auch ist es schwierig, besonders kleine Bögen zu finden, da diese sich im Histogramm kaum als Bögen abzeichnen. Der in diesem Beispiel gefundene Bogen repräsentiert einen Übergang von Fett-/Muskelgewebe zu Luft. Im VR ist sogar eine gute Visualisierung möglich. Die Frage ist nur, ob dieses Visualisierungsziel vom Benutzer gewünscht ist. Interessanter ist es, den Bogen, der den Übergang von den Lebergefäßen zu der Leber repräsentiert, zu finden. Dieser kann jedoch aufgrund der geringen Hervorhebung im Histogramm und der kleinen Ausmaße nicht gefunden werden.

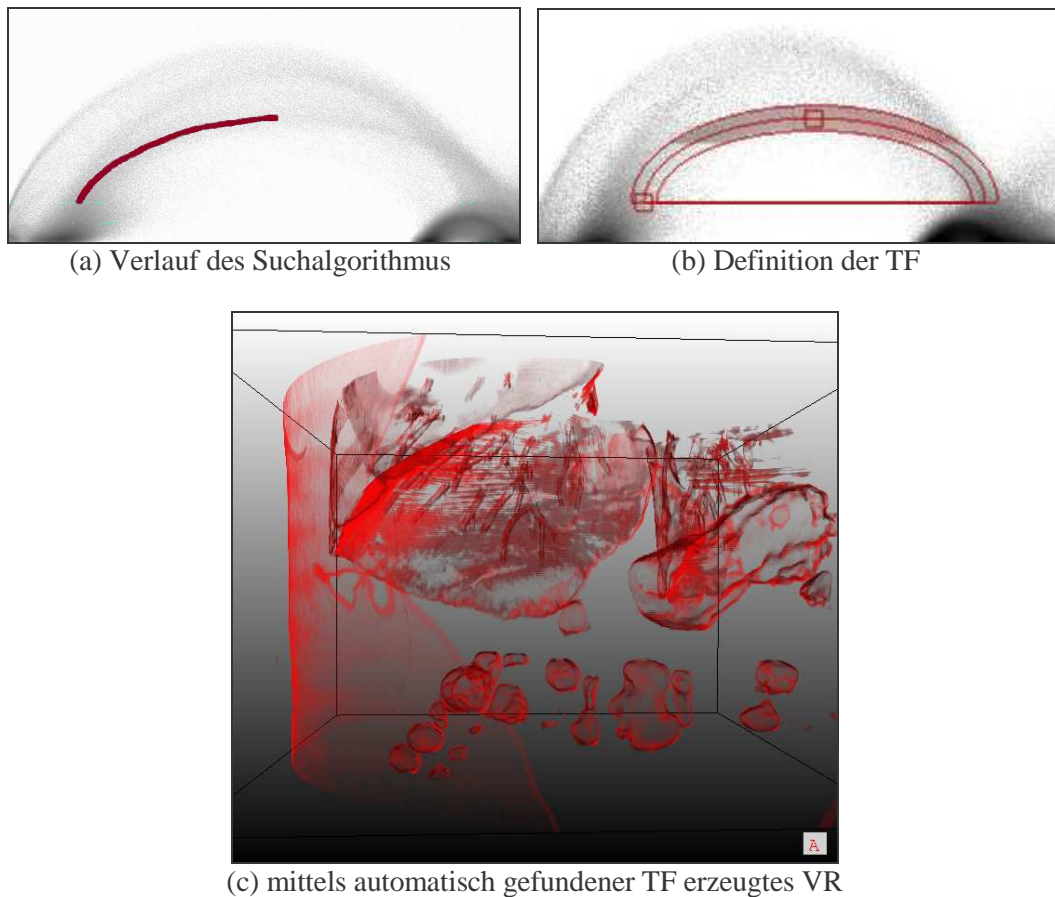


Abbildung 4-47: Automatisch gefundener Bogen in medizinischem Datensatz

4.7.4 Definition von Presets

Presets werden verwendet, um Standardeinstellungen für verschiedene Visualisierungsziele zu speichern und dem Benutzer die idealen Einstellungen für ein Visualisierungsziel zur Verfügung zu stellen. Das Prinzip ähnelt den von Marks et al. vorgestellten Design Galleries. Die Parameter der TFs werden gespeichert und diese sollen bei einem anderen Datensatz wieder

verwendet werden können. So könnten Bögen für einen bestimmten Übergang gespeichert werden.

Für MRT-Datensätze können Presets ausgeschlossen werden, da sich zum einen keine Bögen abzeichnen und zum anderen die Daten untereinander zu unterschiedlich voneinander sind. Es kann keine Standardisierung festgelegt werden.

Bei CT-Datensätzen muss Folgendes beachtet werden:

- Die Intensitätswerte bleiben bei nicht kontrastierten CT-Datensätzen immer gleich.
- Bei weich rekonstruierten Datensätzen sind die Bögen in y-Richtung kleiner als bei hart rekonstruierten. Außerdem ändert sich die Breite der Bogenfläche.
- Vorfilterungen haben Auswirkungen auf die Höhen der Bögen.
- Die Höhe der Bögen variiert mit der Voxelgröße.
- Sinnvoll ist es, einen kompletten Bogen zu speichern, wobei die Bogenbreite vom Benutzer angepasst werden muss. Sie ist nicht vorhersehbar (hängt vom Rauschen im Datensatz ab).
- Gradienten sind von Aufnahme zu Aufnahme konstant, wenn der Rekonstruktionskernel, die Voxelgröße und das Aufnahmegerät gleich bleiben.

Bei weich-rekonstruierten CT-Daten sind Presets sinnvoll. Dagegen ist die Definition von Presets für hart-rekonstruierte Daten eher zwecklos, da schon per Hand keine geeignete Visualisierung erzeugt werden kann und der gradientenbasierte Ansatz weich-rekonstruierte Datensätze vorrausgesetzt.

Eine wichtige Grundlage für die Anwendung von Presets bildet der Sobelfilter. Dieser berechnet die Gradienten, die von Datensatz zu Datensatz variieren. Abweichungen der Gradienten entstehen zum einen durch die Art der Rekonstruktion des Datensatzes und zum anderen durch unterschiedliche Voxelgrößen. Das Problem der Voxelgröße sollte behoben werden, indem die Gradienten mit der Voxelgröße skaliert werden. Leider zeigte sich, dass die Gradienten trotz dieser Skalierung voneinander abweichen. So wurden u.a. ein Datensatz mit kleiner Voxelgröße und ein Datensatz mit großer Voxelgröße untereinander verglichen. Beide Datensätze wurden mit einem Siemens-CT-Scanner aufgenommen und mit dem gleichen Filter rekonstruiert. Die berechneten Gradienten wichen voneinander ab.

Um auszuschließen, dass dieses Problem an den Datensätzen an sich liegt, beispielsweise weil dem Patienten Kontrastmittel injiziert wurde, wurde ein einzelner Datensatz untersucht. Der ursprüngliche Datensatz liegt mit einer Voxelgröße von $0.84 \times 0.84 \times 2.0$ vor. Auf diesem wurde das 2D-Histogramm berechnet. Anschließend wurde der Datensatz neu abgetastet, auf eine Voxelgröße von $1 \times 1 \times 1$ skaliert und dessen Histogramm erstellt. In Abbildung 4-48 (a und b) sind beide Histogramme dargestellt. Die rote Markierung dient als Orientierung. Sie befindet sich in beiden Histogrammen an der gleichen Intensitäts-Gradienten-Position. Es ist erkennbar, dass der große Bogen des rechten Histogrammbildes größer ist als der im linken Bild. Um eine optische Täuschung auszuschließen, wurde das Differenzbild aus dem 1. und dem 2. Histogrammbild berechnet (c). Hierbei kann deutlich erkannt werden, dass beide Histogramme voneinander abweichen. Der große Bogen ist in dem Differenzbild zweimal vorhanden (ein ganz großer Bogen und ein etwas kleinerer). Somit können keine festen Presets für gleiche Rekonstruktionsfilter definiert werden. Der Preset muss von Datensatz zu Datensatz angepasst werden.

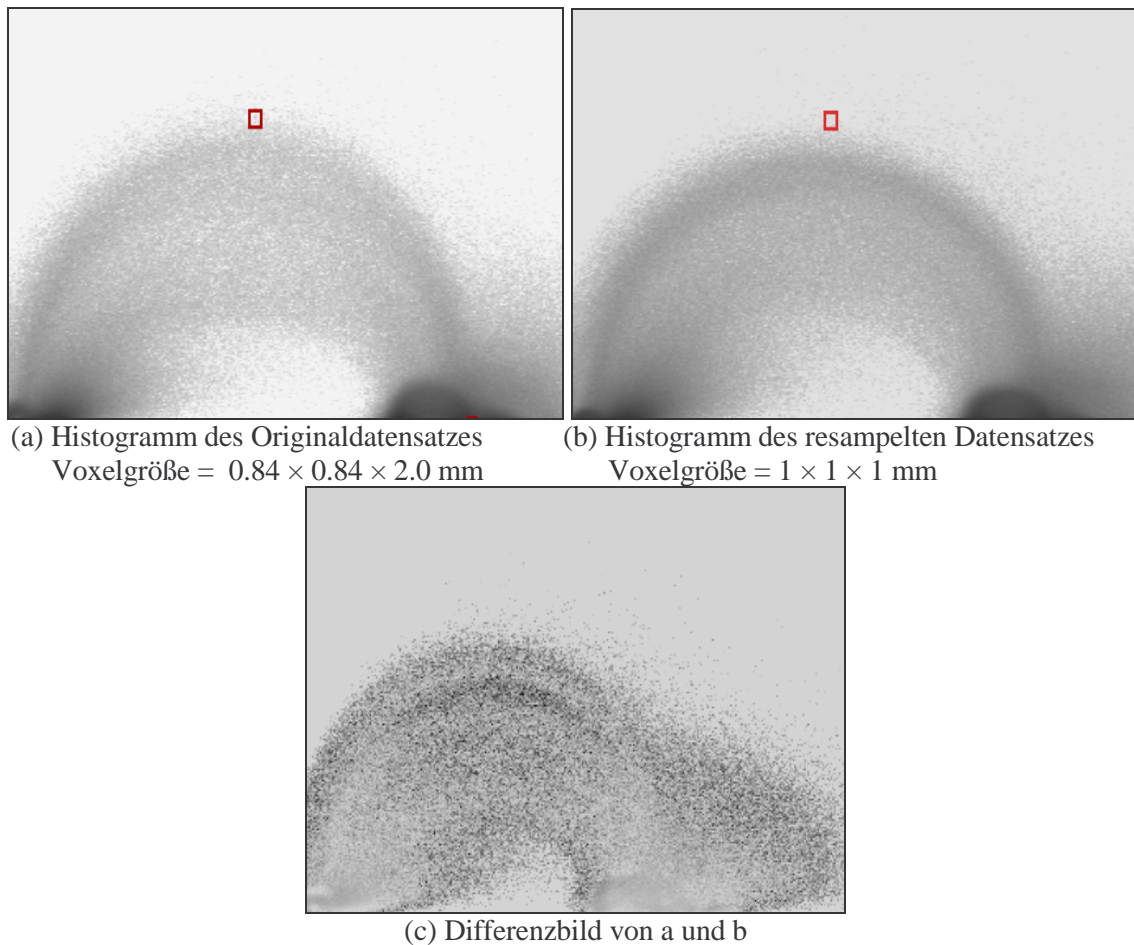


Abbildung 4-48: Histogramme eines Datensatzes

Die Möglichkeit Presets zu definieren, ist bei gradientenbasierten TFs stark eingeschränkt. Es kann festgehalten werden, dass keine festen Presets definiert werden können. Die Presets müssen von Datensatz zu Datensatz angepasst werden. Ein Preset kann aus einem oder mehreren Bögen bestehen. Die Bögen ändern sich bei nicht kontrastierten CT-Datensätzen einzig in ihrer Höhe.

Es ist vorstellbar, dass die Bögen eines weich-rekonstruierten nicht kontrastierten CT-Datensatzes als Presets gespeichert werden. Soll das Preset in einem anderen Datensatz verwendet werden, so muss eine Skalierung der Höhen der Bögen durchgeführt werden. Diese kann entweder vom Benutzer angegeben oder automatisch ermittelt werden. Um die Höhe automatisch zu skalieren, kann eine Histogrammanalyse durchgeführt werden. Der größte Bogen des Histogramms tritt immer deutlich hervor. Die Presets könnten an diesem Bogen ausgerichtet werden. Soll das Preset auf einem anderen Datensatz angewendet werden, so kann in dem neuem Histogramm nach dem großen Bogen gesucht, die Höhe dieses berechnet werden und alle im Preset enthaltenen Bögen mittels dieser Höhe skaliert werden. Der Benutzer kann dann, wenn nötig, die Breiten der Bögen anpassen und erzeugt so auf einfache Weise ein VR.

4.8 Zusammenfassung

In diesem Kapitel wurde die Definition histogrammbasierter gradientenabhängiger 2D-TFs beschrieben. Dabei wurde zuerst das 2D-Histogramm analysiert und aufgrund der Analyse Konzepte zur Entwicklung von 2D-Repräsentationen verglichen. Dabei zeigte sich, dass eine

HE die einfachste und geeignetste Repräsentation für den gradientenbasierten Entwurf ist. Anschließend wurden TFs spezifiziert, die sich für die Kantenbetonung eignen.

Im Anschluss daran sind Interaktionskonzepte vorgestellt worden, die es dem Benutzer ermöglichen, einfach und schnell TFs zu generieren. Zusätzlich wurden Ansätze zur automatischen Suche der Bögen vorgestellt, um die erforderliche Benutzerinteraktion aufs Minimum zu beschränken. Als letztes wurden theoretische Betrachtungen zur Erzeugung von Presets vorgenommen.

In diesem Kapitel wurden alle Konzepte zur Definition von gradientenabhängigen TFs vorgestellt und im nächsten Kapitel wird auf die Umsetzung und Implementierung eingegangen.

5 Realisierung

Dieses Kapitel beschreibt die Realisierung der zuvor konzipierten TFs-Definition, basierend auf den halbelliptischen Bogenflächen. Begonnen wird mit der Vorstellung hierzu verwendeter Programmierwerkzeuge. Anschließend wird die Implementierung der TFs geschildert. Daraufhin folgt die Umsetzung nützlicher Zusatzwerkzeuge, die den Benutzer bei der Suche nach einer geeigneten TF unterstützen. Abschließend wird der Entwurf eines gradientenabhängigen TFs-Editors vorgestellt.

5.1 Programmierwerkzeuge

Ein wichtiger Aspekt bei der Auswahl der Programmierwerkzeuge war die Anbindung an den Volumenrenderer von MeVisLab. Dieser ist ein Modul der Softwareplattform MeVisLab.

Die bei MeVis entwickelten Module (intern als C++-Klassen realisiert) wurden speziell für die Erforschung und Entwicklung im Bereich der medizinischen Bildgebung und deren Visualisierung entworfen. Sie werden dabei unterschieden in Bildverarbeitungsmodule (ML-Module bzw. MeVis Image Processing Library-Module) und in Visualisierungsmodule. Zur Visualisierung wurde OpenInventor benutzt. Von MeVis wurden zusätzlich eigene Inventormodule entwickelt, darunter fällt auch der Volumenrenderer. Alle Module sind in der Programmiersprache C++ implementiert. Als Entwicklungsumgebung wurde Visual C++ verwendet.

5.1.1 OpenInventor

OpenInventor ist eine auf OpenGL beruhende objektorientierte Graphikbibliothek, die es ermöglicht 3D-Objekte zu modellieren und darzustellen. Eine Inventorszene setzt sich aus mehreren Knoten zusammen, wobei diese Knoten geometrische Formen, Materialeigenschaften, Transformationen, Kameras u.a. repräsentieren.

Inventorszene sind baumartig aufgebaut. Hier wird von einem Inventorgraphen gesprochen. Inventorknoten können miteinander verknüpft werden, so dass eine Szene aufgebaut werden kann. Alle Knoten des Szenegraphen werden zur Darstellung von oben nach unten und von links nach rechts traversiert. Die Eigenschaften eines Knotens ergeben sich aus den in den Baum eingefügten Knoten zur Transformation, zur Definition von Materialeigenschaften u.a. Sie wirken sich auf alle nachfolgenden Knoten im Graphen aus, wenn sie nicht von einem weiteren Knoten wieder rückgängig gemacht oder überschrieben werden.

OpenInventor ermöglicht die Visualisierung von 3D-Objekten und bietet zudem eine Ereigniskontrolle für 3D-Interaktionen. Außerdem lassen sich Animationen von 3D-Objekten erstellen und es erlaubt das Selektieren von 3D-Objekten. OpenInventor ist Plattform unabhängig. [Sgi04]

5.1.2 Die Softwareplattform MeVisLab

Die Softwareplattform MeVisLab bildet den Rahmen für die Benutzung der bei MeVis entwickelten Module. MeVisLab läuft auf den Betriebssystemen Windows NT, Windows2000, Windows XP und auf Linux. Es benötigt eine OpenGL-fähige Graphikkarte und mindestens einen PII-Prozessor mit 256MB RAM.

Die ML stellt dem Entwickler eine Menge von Algorithmen zur Verfügung. Über ein in C-implimentiertes Interface kommuniziert MeVisLab mit der ML und den OpenInventor-Klassen. Als eine zentrale Komponente von MeVisLab entwickelt, kann die ML jedoch auch einzeln benutzt oder in andere Softwaresysteme eingebunden werden.

In der Softwareplattform MeVisLab können die einzelnen Module miteinander verknüpft werden. Hier wird von Netzwerken gesprochen. Der Benutzer kann dabei je nach dem gewünschten Aufgaben- und Visualisierungsziel verschiedene Module auf die Oberfläche laden und untereinander verknüpfen. So kann er ein komplexes Netzwerk zusammenstellen, das Aufgaben wie das Einlesen und Speichern von Daten übernimmt und in dem Algorithmen der Bildverarbeitung und der Visualisierung miteinander vereint werden können.

An einem einfachen Beispiel wird dies deutlich:

Ziel ist es, einen medizinischen Datensatz in den Hauptspeicher einzulesen und von diesem Datensatz ein Kantenbild berechnen zu lassen. Im Anschluss daran soll das Ergebnis auf dem Monitor dargestellt werden. Für diesen Prozess sind insgesamt drei Module nötig. In Abbildung 5-1 sind diese Module dargestellt. Das Modul *ImgLoad* übernimmt die Funktion des Einlesens der Daten in den Hauptspeicher. Der eingelesene Datensatz wird über eine Ausgangsschnittstelle an die Eingangsschnittstelle des Moduls *Sobel3D* übergeben. Dieses berechnet das Kantenbild, welches wiederum über die Schnittstellen an das Modul *View2D* übergeben wird. Dieses übernimmt die Aufgabe der Darstellung des Ergebnisbildes auf dem Monitor.

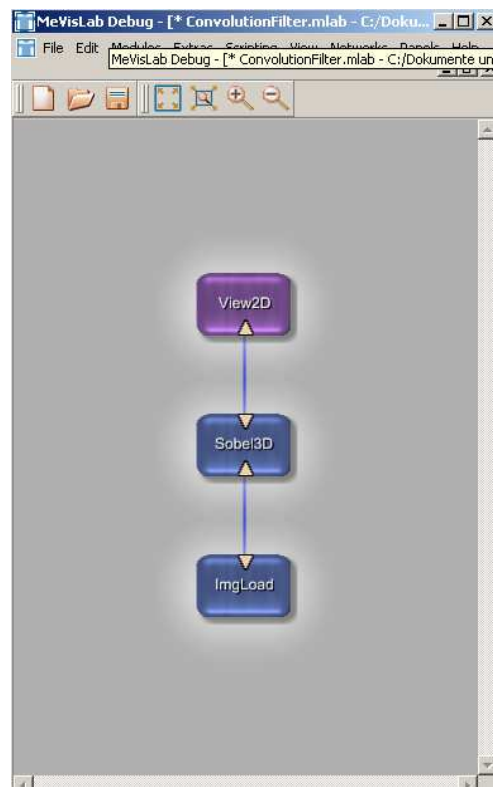


Abbildung 5-1: Die Softwareplattform MeVisLab mit einem Beispielnetzwerk

Einstellungen der Module sind in einem Panel zusammengefasst. Dieses öffnet sich mit einem Doppelklick auf das Modul. Bei dem *ImgLoad*-Modul muss beispielsweise eingestellt werden, welches Bild geladen werden soll, siehe Abbildung 5-2. Außerdem werden Eigenschaften des geladenen Datensatzes angezeigt.



Abbildung 5-2: Panel des *ImgLoad*-Moduls

In MeVisLab wird zwischen drei verschiedenen Arten von Schnittstellen unterschieden. Zum einen gibt es die Ein- und Ausgänge für Bilder. Diese werden an den Modulen dreieckig dargestellt und dienen dazu Bilder zwischen Modulen auszutauschen. Einen weiteren Schnittstellentyp bilden die Inventorschnittstellen. Diese sind halbkreisförmig dargestellt und ermöglichen den Transport von Inventorinformationen. Schließlich gibt es noch die *MLBase*-Schnittstelle, mit der Objekt- und Listenklasseninformationen ausgetauscht werden können. Diese wurden bisher ebenfalls halbmondförmig visualisiert, sind in der neuen Version von MeVisLab jedoch viereckig dargestellt.

In Abbildung 5-3 ist ein kleines Netzwerk, in dem alle möglichen Schnittstellentypen auftreten, dargestellt. Zuerst wird ein Bild mit dem Modul *LocalImage* geladen. Dieses wird über einen Bildausgang an den Viewer *View2D* übergeben. An dem Viewer befindet sich ein über Inventoreingänge verbundenes *SoView2DSphere*-Modul. Mit diesem Modul können Kugeln gezeichnet und im Datensatz visualisiert werden. Werden mehrere Kugeln gezeichnet und sollen diese dauerhaft gespeichert werden, kann die Kugelklasse an das *SaveBase*-Modul übergeben werden, welches den aktuellen Zustand aller in der Klasse enthaltenen Variablen in einer Datei ablegt.

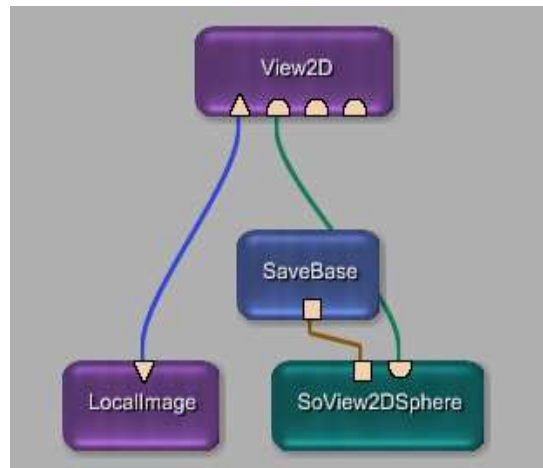


Abbildung 5-3: Bildverarbeitungsnetzwerk mit unterschiedlichen Ein- und Ausgängen von Modulen

Auf den www-Seiten von MeVisLab [MLab04] befinden sich detaillierte Informationen und freie Software zum Testen.

5.1.3 *SoView2D*-Klasse

Eine bei MeVis implementierte Inventorklasse ist die *SoView2D*-Klasse. Alle implementierten Inventorklassen erben von der *SoNode*-Klasse, die in der OpenInventor-Bibliothek definiert ist. Die Ein- und Ausgänge eines Inventorknotens werden ebenfalls von der *SoNode*-Klasse geerbt. Inventormodule können ebenso über Ein- und Ausgänge des Typs *MLBase* verfügen, welches die Grundlage für die Übergabe von Listenklassen an ein anderes Modul bildet.

Die *SoView2D*-Klasse ist eine Unterklasse der *SoView2DExtension*-Klasse, welche sich Inventorfunktionen, wie die Ereigniskontrolle von Maus und Tastatur, zu Nutze macht und das Zeichnen von Objekten in einem Schichtdatensatz ermöglicht. Diese Klasse stellt somit benötigte Funktionen zur Verfügung, sodass bei einem Erben von dieser Klasse, ein Großteil der Funktionalität übernommen werden kann. Wichtige Funktionen, die von dieser Klasse geerbt werden, sind die *evalEvent*-Funktion und die *draw*-Funktion.

Mit der *draw*-Funktion kann in einem Schichtbild gezeichnet werden, wobei diese Funktion die aktuelle Schicht, auf der gezeichnet werden soll, ermittelt und zusätzlich eine Reihe von Funktionen für die Koordinatentransformationen zur Verfügung stellt. Um in einen medizinischen Datensatz zeichnen zu können, müssen Koordinatentransformationen durchgeführt werden. In medizinischen Datensätzen wird zwischen zwei verschiedenen Koordinatensystemen unterschieden, dem Weltkoordinaten- und dem Voxelkoordinatensystem. Diese müssen

zum Zeichnen durch entsprechende Transformationen in Gerätekoordinaten (beispielsweise Bildschirmkoordinaten) umgewandelt werden. Abbildung 5-4 beschreibt die Transformationen der einzelnen Koordinatensysteme.

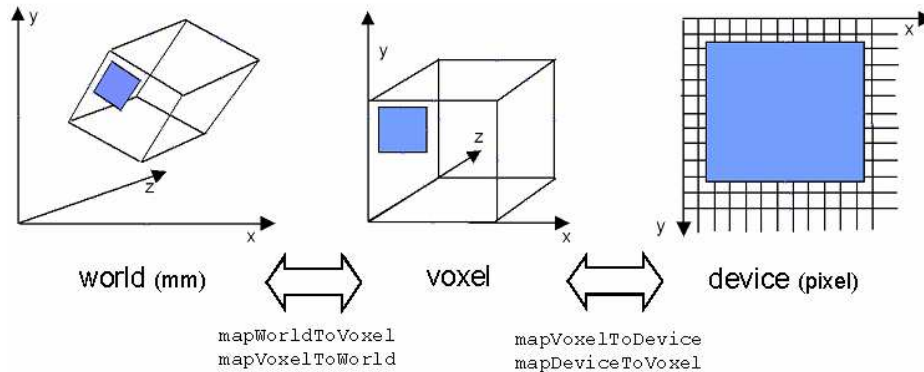


Abbildung 5-4: 3D-Koordinatentransformationen. Quelle: [Lin04]

Die *evalEvent*-Funktion dient dazu, Maus sowie Tastatur-Ereignisse zu überwachen und entsprechende Aktionen auszulösen. In einem *Event-Container* werden die einzelnen Ereignisse verwaltet und der Programmierer kann auf die verschiedenen Ereignisse reagieren.

SoView2D-Module werden mit einem Viewer, der Schichtbilder darstellt, verbunden. In diesem Viewer können dann die entsprechenden Inventorfunktionen benutzt werden.

Die Koordinaten des Histogrammbildes

Das 2D-Histogramm wird in MeVisLab als 2D-Bild dargestellt. An dieser Stelle werden die Koordinaten dieses Bildes erklärt. Wie bereits erwähnt, wird in medizinischen Datensätzen zwischen Welt- und Voxelkoordinaten unterschieden. Das Histogrammbild ist kein medizinischer Datensatz, dennoch findet auch hier diese Unterscheidung statt. In den Voxelkoordinaten spiegeln sich die Bins wieder, d.h. ein Bin entspricht einem Voxel. Die Weltkoordinaten entsprechen auf der x-Achse den Intensitätswerten, die auf die Bins abgebildet werden und auf der y-Achse den Gradientenstärken. Soll auf dem Histogrammbild eine HE gezeichnet und die Parameter gespeichert werden, so ist darauf zu achten, dass die Weltkoordinaten gespeichert werden, denn nur anhand dieser Koordinaten, kann berechnet werden, welchen Bereich die Bögen abdecken. Auch ist dies gerade bei der Generierung von Presets wichtig, da bei einem neuen Datensatz, die Intensitätswerte auf andere Bins (Voxel) abgebildet werden. Werden die Weltkoordinaten abgespeichert, kann die Abbildung der Werte auf Bins völlig außer Acht gelassen werden. Die HE wird an der richtigen Position (Intensitäts-/Gradientenposition) im Bild gezeichnet.

5.2 Generierung gradientenabhängiger 2D-TFs

Dieser Abschnitt beschäftigt sich mit der Generierung von 2D-TFs. Dabei wird zuerst darauf eingegangen, wie die halbelliptischen Bogenflächen gezeichnet und welche Klassen dafür benötigt werden. Im Anschluss daran wird gezeigt, wie ausgehend von der internen Repräsentation eine LUT berechnet wird.

5.2.1 Entwurf einer *SoView2D*-Klasse für Halbellipsen

Die *SoView2D*-Klasse bietet die gesamte Funktionalität, die zum Zeichnen einer HE benötigt wird. Es werden Zeichen-Funktionen und Ereigniskontroll-Funktionen benötigt. Die HE kann mit zwei Kontrollpunkten gezeichnet werden. Diese zwei Kontrollpunkte kann der Benutzer mit der Maus festlegen. Daher ist es nötig, Mausereignisse zu überwachen. Wenn der Benutzer durch Betätigung der linken Maustaste im Histogrammbild einen Punkt auswählt, so wird der erste Punkt, der Extrempunkt der HE, festgelegt. Das Ereignis wird erst beendet, wenn der Benutzer die linke Maustaste loslässt. Der Punkt, an dem der Mauszeiger zu diesem Zeitpunkt ist, ist der zweite Punkt, der Fußpunkt. Die Mausereignisse werden in der *evalEvent*-Funktion überwacht. Betätigt der Benutzer in dem Fenster des Viewers die linke Maustaste, so wird die *draw*-Funktion aufgerufen. In der *draw*-Funktion werden die Zeichen-Funktionen gesteuert. Hierbei kommen OpenGL-Befehle zum Einsatz. Die HE wird während der Bewegung des Mauszeigers zum Fußpunkt gezeichnet, sodass der Benutzer den Verlauf erkennt und realisiert, ob der Fußpunkt an der richtigen Stelle gesetzt wird.

Dem Benutzer muss außerdem die Möglichkeit gegeben werden, den Verlauf der Ellipse zu korrigieren, sodass entsprechende Folgeereignisse überwacht werden müssen. Klickt der Benutzer mit der linken Maustaste in das Histogramm, so wird überwacht, ob sich der Mauszeiger in der Nähe einer der Kontrollpunkte befindet. Ist dies der Fall, so wird bei gedrückter linker Maustaste die HE entsprechend geändert.

Mit der Entfernen-Taste kann die gesamte HE gelöscht werden. Das Modul wird mit einem Viewer verbunden, in dem das Histogrammbild dargestellt ist und an das die Inventorfunktionen zum Zeichnen der HE-förmigen Bogenflächen übergeben werden.

Um eine Kontrolle über die Farbzugeordnungen zu erhalten, wird eine *Stylepalette* mit dem HE-Modul verbunden. Die *Stylepalette* ist ein MeVis-eigenes Inventor-Modul, welches es ermöglicht, bis zu 12 Farben festzulegen. Diese Farben werden den Bögen zugeordnet. Die Farben der Bögen bilden die Farbe der TF. Das implementierte Modul wurde *SoView2DParable* genannt. Es ist in der Lage, Parabeln 2. und 4. Ordnung und HEs zu zeichnen. In Abbildung 5-5 ist eine HE dargestellt. Zu erkennen sind die beiden Kontrollpunkte, die den Extrempunkt und Fußpunkt spezifizieren. Der Benutzer legt die mittlere HE an. Durch die Angabe einer Höhe (Abstand am Extrempunkt) und einer Breite (Abstand am Fußpunkt) kann er zwei äußere HEs anlegen, wodurch die halbelliptische Bogenform vervollständigt wird.

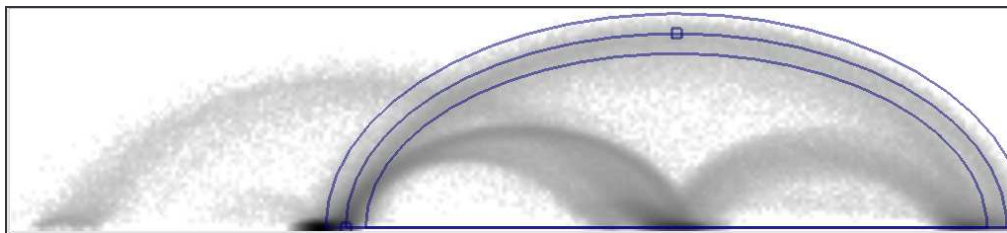


Abbildung 5-5: Eine halbelliptische Bogenfläche

5.2.2 Entwurf einer Listenklasse zur Speicherung der Halbellipsen

Für die Verwaltung mehrerer HEs werden diese in einer eigenen Objektklasse abgelegt. Mehrere HEs sind in einer Listenklasse zusammengefasst.

Sollen alle Einstellungen der Halbellipsen dauerhaft gespeichert werden, bietet MeVisLab die Möglichkeit, den aktuellen Zustand aller in der Klasse enthaltenen Variablen in Dateien abzuliegen. Dafür wird von der Klasse *MLBase* geerbt. Aus dieser Klasse werden Funktionen über-

laden, die das Speichern von Klasseninformationen realisieren. Die C++-Objekte werden intern in einer hierarchischen Struktur abgelegt, wofür ein XML-TreeNode verwendet wird. Mit den Funktionen *addStateToTree()* und *readStateFromTree()* werden Objekte zu einem XML-Baum hinzugefügt bzw. aus dem Baum ausgelesen. Diese Funktionen werden automatisch aufgerufen. Der Programmierer muss die für die eigene Klasse relevanten Variablen zum XML-Baum mittels *addChild()* hinzufügen und kann diese mittels *readChild()* aus dem Baum auslesen. In MeVisLab stehen die Module *SaveBase* und *LoadBase* zur Verfügung, mit denen es möglich ist, den angelegten XML-Baum in einer xml-Datei abzulegen bzw. aus einer xml-Datei die Variablen einer Klasse einzulesen.

5.2.3 Berechnung der 2D-LUT

Nachdem die Eigenschaften der TF definiert sind, muss im nächsten Schritt eine LUT berechnet werden. Der GVR benötigt ein quadratisches RGBA-Bild.

Um von der halbelliptischen Bogenfläche zu der 2D-LUT zu gelangen, wurden die Bibliotheken der ML zur Erzeugung eines Bildes genutzt. Dabei wird das page-Konzept der ML genutzt, d.h. ein Bild wird in mehrere Teilbilder (pages) zerlegt und es wird nie auf dem ganzen Bild gerechnet, was zu einer beschleunigten LUT-Berechnung führt.

Eigentlich muss für jedes Pixel der LUT überprüft werden, ob es in dem Bereich einer halbelliptischen Bogenform liegt. Das page-Konzept der ML kann in der Art ausgenutzt werden, dass jede halbelliptische Bogenform von einer Box eingeschlossen wird. Die LUT (Ergebnisbild) wird in mehrere pages zerlegt. Jede page wird getestet, ob diese in einer Boundingbox der halbelliptischen Bogenform liegt. Falls dies der Fall ist, so wird für jedes Pixel der page überprüft, ob es die HE-förmige Bogenfläche schneidet. Falls die page keine Boundingbox schneidet, kann die nächste page überprüft werden. Dadurch reduziert sich der Aufwand der LUT-Berechnung.

Die Entscheidung, ob ein Pixel den Bereich der halbelliptischen Bogenform schneidet, kann mithilfe von zwei HEs getroffen werden, einer HE mit großem und einer mit kleinem Radius. Mit der impliziten Beschreibung der beiden HEs kann die Lagebeziehung eines Punktes in Bezug zu den HEs bestimmt werden, d.h. es kann berechnet werden, auf welcher Seite der HEs der Punkt liegt.

Liegt der Punkt zwischen den beiden HEs, so werden im nächsten Schritt Transparenz- und Farbwerte des Pixels berechnet.

Da die LUT bei älteren Graphikkarten auf eine Auflösung von 256^2 Pixel und dadurch die Präzision sehr stark eingeschränkt wird, wurde noch ein Supersampling (Überabtastung eines Pixels) implementiert, welches eine genauere LUT erzeugt. Der Wert des Pixels ergibt sich aus dem Mittelwert aller Abtastpunkte. Dieses Prinzip kann bei der Berechnung der LUT angewendet werden, wodurch der Bogen genauer approximiert wird. Diese Option wirkt sich jedoch nur auf das Erscheinungsbild des Bogens in der LUT aus. Auf das VR hat dieses Option wenig Einfluss, da der Rand des Bogens nur von geringer Bedeutung ist. Außerdem verzögert sich die LUT-Berechnung.

5.3 Werkzeuge zur vereinfachten Erkennung von Bögen

Um dem Benutzer den Umgang mit dem Entwurf der TFs etwas zu erleichtern, sind die nachstehenden zusätzlichen Module von Bedeutung. Diese Module wurden in Abschnitt 4.6.3 vorgestellt.

5.3.1 JointHistMask

Das *JointHistMask* dient dazu, die entwickelte LUT im Schichtdatensatz darzustellen. Dafür wird der Schichtdatensatz neu berechnet, wobei jedes Voxel die Farbe des Wertes, der in der LUT steht, erhält. Kombiniert mit dem Originaldatensatz kann der Benutzer erkennen, welche Organe im VR gefärbt werden und welche transparent sind. Hier findet der Arzt seine altbekannte Visualisierungsmethode wieder. Er kann zuerst den Schichtdatensatz analysieren und dann die LUT angeben, während er gleichzeitig in dieser Ansicht überprüfen kann, ob das was gefärbt wird, seinen Visualisierungszielen entspricht.

5.3.2 Kantendetektion

Dieses Tool dient dazu, den Benutzer bei der Suche nach dem zu einer Kante korrespondierendem Bogen zu unterstützen. In MeVisLab können Netzwerke entwickelt und als sog. Makros zusammengefasst werden. Die wichtigsten Eigenschaften der Module werden auf einer für das Makro definierten Oberfläche zusammengefasst. Zur Realisierung dieses Tools wurden verschiedene Module aus MeVisLab miteinander kombiniert und in einem Makromodul zusammengefasst. Von einem mit der Maus interaktiv selektiertem Voxel der Schichtdatendarstellung werden der Intensitäts- und der Gradientenwert mittels des Moduls *GetVoxelValue* ermittelt. Anschließend wird das Bin des Histogramms berechnet, welches diese Intensitäts-Gradienten-Kombination repräsentiert. Es wird ein Bild (mit Ausmaßen des Histogramms) mit dem entsprechenden Bin erzeugt. Wird noch ein weiteres Voxel des Datensatzes erkundet, so wird das neue Bin zu dem Bild hinzugefügt. Dies ist realisiert über das *MemCache*-Modul, welches das alte Bild im Speicher hält. Außerdem wird die Position des aktuellen Bins über einen Marker nach außen gegeben. Dieser und der *Overlay*-Operator können mit dem histogrammdarstellenden Viewer verbunden werden, so dass darin sichtbar wird, welches Bin aktuell und welche vorher gefundenen Bins die Intensitäts-Gradientenkombination repräsentieren. Der Aufbau des Makros ist in Abbildung 5-6 dargestellt.

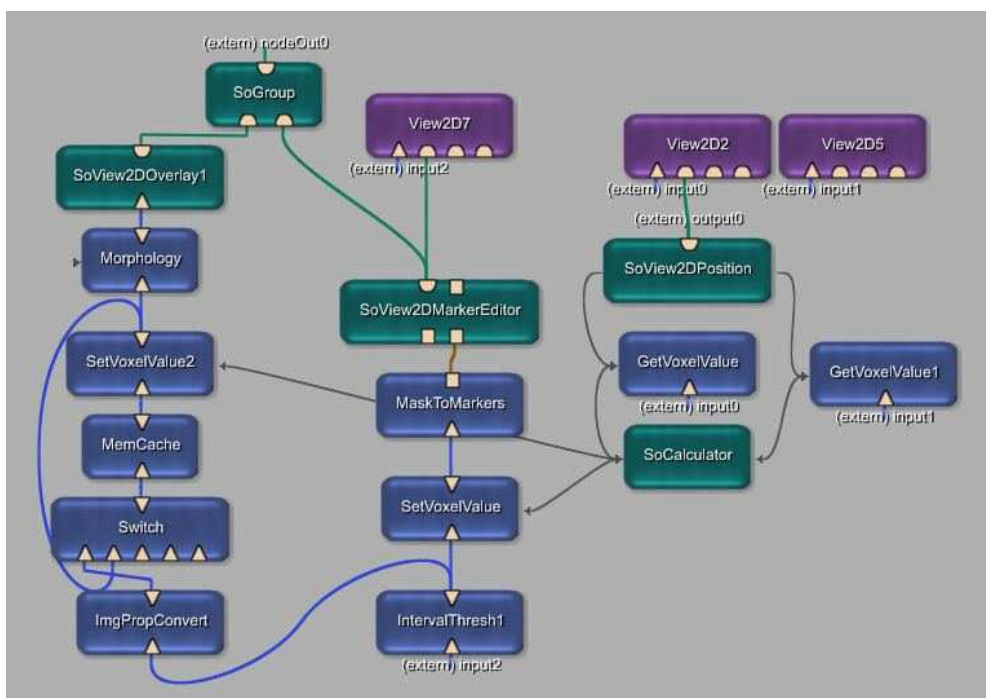


Abbildung 5-6: Aufbau des Makromoduls zur Kantendetektion

Das korrespondierende Bin wird im Histogramm markiert und gefärbt. Das Panel wurde in der Art entworfen, dass der Benutzer mehrere Voxel entlang einer Kante erkunden kann, wenn er den *Draw*-Button drückt. Dann wird jede neue Intensitäts-Gradienten-Kombination im Histogramm abgetragen. Wird der *Reset*-Button betätigt, werden alle gefärbten Bins in der ursprünglichen Intensität dargestellt. Anschließend kann eine neue Kante erkundet werden., siehe Abbildung 5-7. Außerdem kann er noch die Farbe der im Histogramm colorierten Bins festlegen und die Größe angeben. Im *TextMode* kann er die Art der Beschriftung des Markers festlegen.

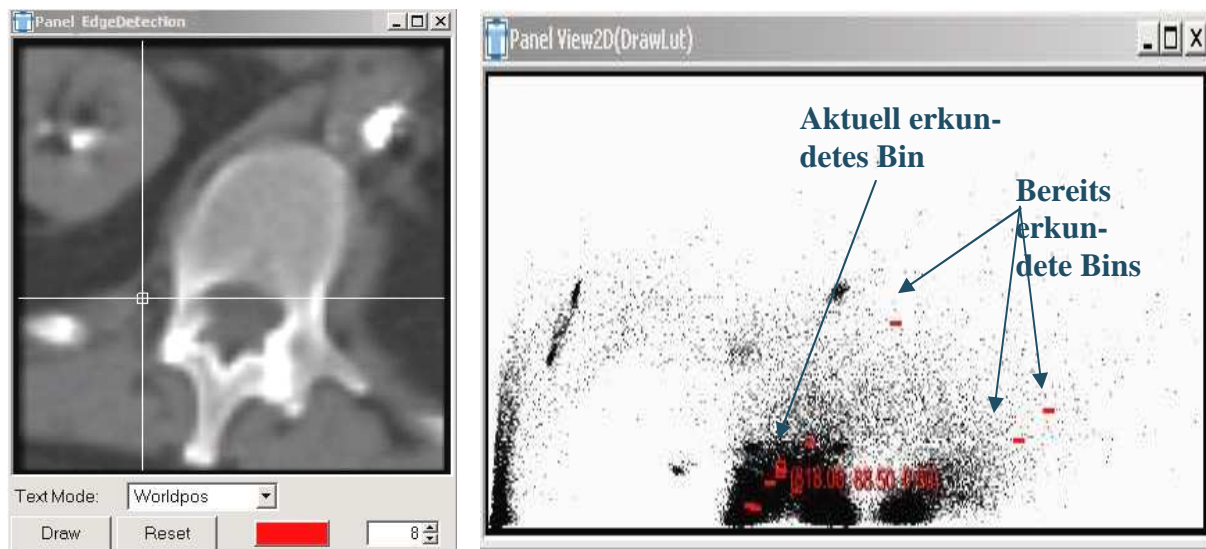


Abbildung 5-7: Makromodul für die Kantendetektion
Im Panel des Makromoduls ist der Schichtdatensatz (links) dargestellt, in dem ein Voxel mit der Maus selektiert werden kann. Im Histogramm (rechts) wird das korrespondierende Bin gefärbt, wobei an das aktuelle Bin die Worldkoordinaten geschrieben werden können. Mit dem *Draw*-Button können mehrere Bins gefärbt werden. Die Farbe lässt sich beliebig wählen. Mit dem *Reset*-Button werden alle gefärbten Bins gelöscht.

5.3.3 Kantendetektion mithilfe einer Segmentierungsmaske

Dieses Tool dient dazu, bereits vorhandenes Wissen von Segmentierungsmasken zur Definition von TFs auszunutzen. Dafür wurde ebenfalls ein Makromodul generiert. Diesem Makromodul werden der Intensitäts-, der Gradientendatensatz und eine Segmentierungsmaske übergeben. Zwei morphologischen Operatoren (*Morphology*-Modul) dilatieren und erodieren die Segmentierungsmaske. Die erodierte Maske wird im *Arithmetic2*-Modul von der dilatierten differenziert. Dadurch wird eine Kontur der Segmentierungsmaske erzeugt. Anschließend werden die zu der neuen Konturmaske korrespondierenden Intensitäts- und Gradientenwerte aus den Eingangsdatensätzen in den *Mask*-Modulen ermittelt. Im *JointHist* wird aus den beiden Kantenbildern (Intensitätsbild und Gradientenbild) ein neues Histogramm berechnet. Dieses wird beispielsweise mit einer roten Färbung mithilfe des *Overlay*-Operators über das ursprüngliche Histogramm gelegt. Durch die Färbung des überlagernden Histogramms können

die aus der Maske resultierenden Bins erkannt und eine TF definiert werden. In Abbildung 5-8 ist der genaue Aufbau des Makromoduls dargestellt.

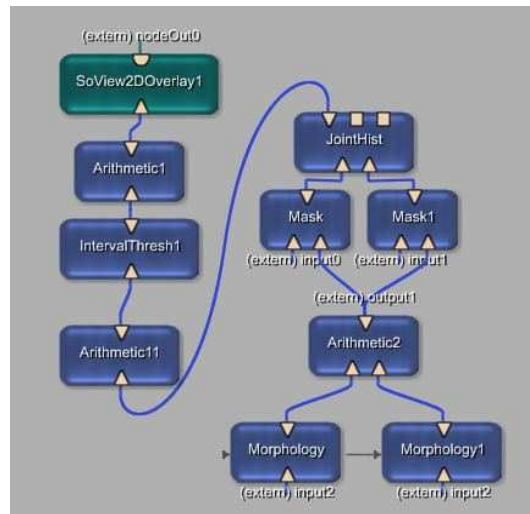


Abbildung 5-8: Aufbau des Makros zur Kantendetektion einer Segmentierungsmaske

5.4 Entwurf eines Editors

Die Berechnung der LUT ist mithilfe eines kleinen Netzwerkes möglich, siehe Abbildung 5-9. In der Abbildung sind alle grundlegenden Module zur Berechnung der LUT und zur Generierung des VRs zusammengefasst.

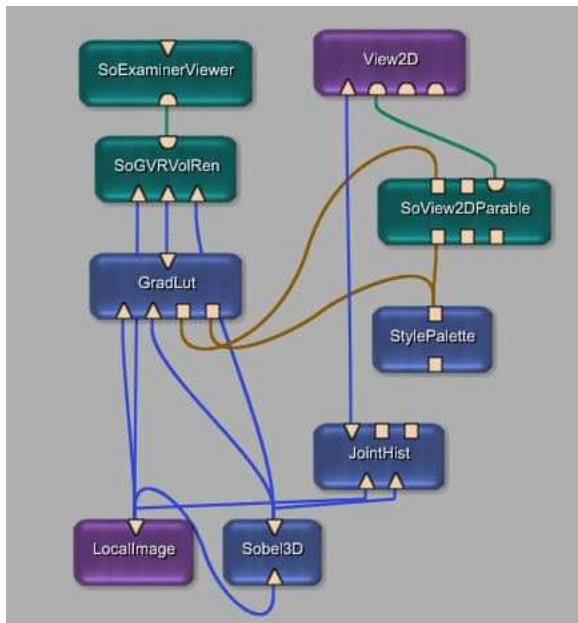


Abbildung 5-9: Beispielnetzwerk für die Erstellung einer LUT.

Mit dem Modul *LocalImage* wird ein Bild geladen. Anschließend werden die Gradienten mit dem *Sobel3D* berechnet. Aus dem Gradientenbild und dem Intensitätsbild wird ein Histogramm im *JointHist*-Modul erzeugt. In dem Viewer *View2D* kann mittels des *SoView2DParable*-Moduls eine HE-förmige Bogenform gezeichnet werden. In *GradLut* wird die LUT berechnet, die dann zusammen mit dem Intensitäts- und Gradientensatz an den VR *SoGVRVolRen* übergeben wird. In dem *SoExaminerViewer* wird das gerenderte Bild dargestellt. Alle benötigten Einstellungen müssen über die Panel der Module für jedes Modul extra vorgenommen werden. So muss das Panel für die Histogrammberechnung geöffnet werden und die Einstellungen vorgenommen werden. Um Parameter des Renderers zu ändern, muss das Panel des GVRs aufgerufen werden. Dieses Vorgehen ist für den Benutzer zu umständlich, daher ist es sinnvoll, das Netzwerk in einem Makromodul zusammen zu fassen und alle nötigen Einstellungen in einem einzelnen Panel darzustellen.

Mithilfe des in Abbildung 5-9 dargestellten Netzwerks ist es möglich gradientenabhängige TFs zu erzeugen. Jedoch soll dem Benutzer ein übersichtliches Modul in die Hand gegeben werden, welches noch mehr Funktionalität bietet. In den vorherigen Abschnitten wurde gezeigt, was für die Definition von Transferfunktionen nötig ist. Der Benutzer muss zuerst die Gradienten und anschließend ein Histogramm, welches in einem Viewer angezeigt wird, berechnen lassen. Es müssen Histogrammparameter (Binsize) eingestellt werden. Außerdem soll dem Benutzer eine Schichtdatendarstellung zur Verfügung gestellt werden, in der der Datensatz betrachtet und der Wirkungsbereich der TF erkannt werden kann. Zusätzlich müssen die Parameter der TFs eingestellt, verschiedene Vorverarbeitungsfilter (Gauß/Median) angeboten, Presets angelegt/gelöscht und ein weiterer Viewer für das VR unter der Oberfläche vereint werden. Auch die Komponenten für die Kantendetektion dürfen nicht außer Acht gelassen werden.

Daher beinhaltet dieses Makro folgende Einstellungen:

- Parametrisierung der Gradientenberechnung
- Einstellung der Parameter zur Histogrammberechnung
- verschiedene Glättungsfilter
- Histogrammdarstellung, auf der die Repräsentation der TF gezeichnet werden kann
- Einstellung der Parameter der TFs
- Schichtdatendarstellung des Volumens + mit LUT-Färbung und Kantendetektionskomponente + Kantendetektion mittels Segmentierungsmaske
- Laden und Speichern von Presets
- Parameter der LUT
- VR + Einstellungen für das VR

Im Anhang A befindet sich die englische Dokumentation, so dass hier auf eine detailliertere Beschreibung verzichtet wird.

5.5 Zusammenfassung

In diesem Kapitel wurden zuerst die für die Implementierung benötigten Programmierwerkzeuge wie MeVisLab, die ML und OpenInventor vorgestellt. Anschließend wurde ein Konzept zum Entwurf einer *SoView2D*-Klasse und einer Listenklasse für halb elliptische Bogenflächen erläutert. Auf die Berechnung der LUT wurde im darauf folgenden Abschnitt eingegangen. Außerdem wurde erläutert, welche Werkzeuge zusätzlich zur Anwendung kommen und wie diese realisiert wurden. Schließlich wurde auf den Entwurf eines Editors für die TFs-Definition eingegangen.

6 Ergebnisse

In diesem Kapitel werden die Ergebnisse vorgestellt und diskutiert. Dabei werden zuerst für den kantendetektierenden TFs-Entwurf gut geeignete Datensätze getestet. Im Anschluss daran werden gradientenabhängige TFs auf CT- und MRT- Datensätzen, wie sie im klinischen Alltag vorliegen, angewendet.

6.1 Für den 2D-TFs-Entwurf gut geeignete Datensätze

Zuerst wurden Datensätze mit optimalen Aufnahmeparametern getestet. Optimale Aufnahmeparameter sind definiert durch eine hohe Abtastrate, weiche Rekonstruktionsfilter und die Datensätze enthalten Gewebe, die sich klar und deutlich voneinander differenzieren.

6.1.1 Zahndatensatz

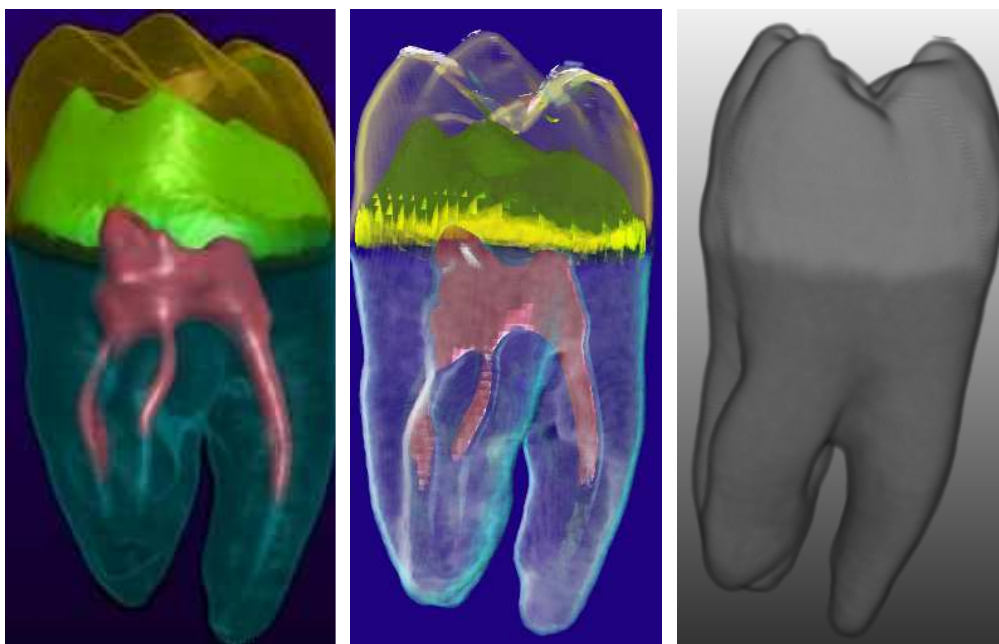
Ein guter Datensatz ist beispielsweise der Zahn, der in dieser Arbeit schon häufig verwendet wurde. An diesem ließen sich die Zusammenhänge zwischen TF und Volumenrendering sehr gut verdeutlichen. Im Histogramm dieses Datensatzes treten die Bögen stark hervor und die TF kann leicht definiert werden. Dieser Zahn wurde vor allem von der Gruppe um Kindlmann und Kniss verwendet, um die Visualisierungsergebnisse ihrer TFs zu verdeutlichen.

Eigenschaften des Datensatzes:

Der Datensatz wurde mit einem High-Resolution-CT-Gerät aufgenommen.

Größe: $103 \times 97 \times 183$, isotrope Voxelgröße, Rekonstruktion: weich

Die Materialien heben sich deutlich voneinander ab.



Zahn mit dreidimensionaler TF. Quelle: [KK01]

Zahn mit zweidimensionaler TF

Zahn mit eindimensionaler TF (ansteigende Rampe)

Abbildung 6-1: Zahn im Vergleich

Das VR dieses Zahnes wurde von Kindlmann mit einer dreidimensionalen TF erstellt, siehe Abbildung 6-1 (links). Es sieht auf den ersten Blick besser aus als das, welches nur mit einer zweidimensionalen TF (siehe Abbildung 6-1 (mitte)) erzeugt wurde. Jedoch tritt diese Wirkung einzig wegen der bei Kindlmann zum Einsatz kommenden komplexen Beleuchtung auf. Im Detail sind beide Zähne gleich gut dargestellt. Die Strukturen konnten mit den zweidimensionalen TFs genauso gut herausgearbeitet werden wie mit den dreidimensionalen. Der gelbe Rand im mittleren Bild entsteht durch die Beleuchtung. Das grüne Material wird so stark beleuchtet, dass es gelb erscheint.

Mit einer eindimensionalen TF können die äußeren Strukturen nicht gleichzeitig mit den inneren Strukturen dargestellt werden. Hier überlagern sich die Intensitätsbereiche der einzelnen Gewebe zu stark.

6.1.2 Visible-Human-Kopf

Das Visible-Human-Projekt wurde 1986 von der „National Library of Medicine“ ins Leben gerufen. Ziel ist es, anhand detaillierter Datensätze den menschlichen Körper bis in kleine Details zu studieren.

Eigenschaften des Datensatzes:

Der Datensatz wurde mit einem CT-Gerät aufgenommen.

Größe: $128 \times 256 \times 256$, isotrope Voxelgröße: $1 \times 1 \times 1$ (mm), Rekonstruktion: weich,

Auch bei diesem Datensatz heben sich die Materialien deutlich voneinander ab

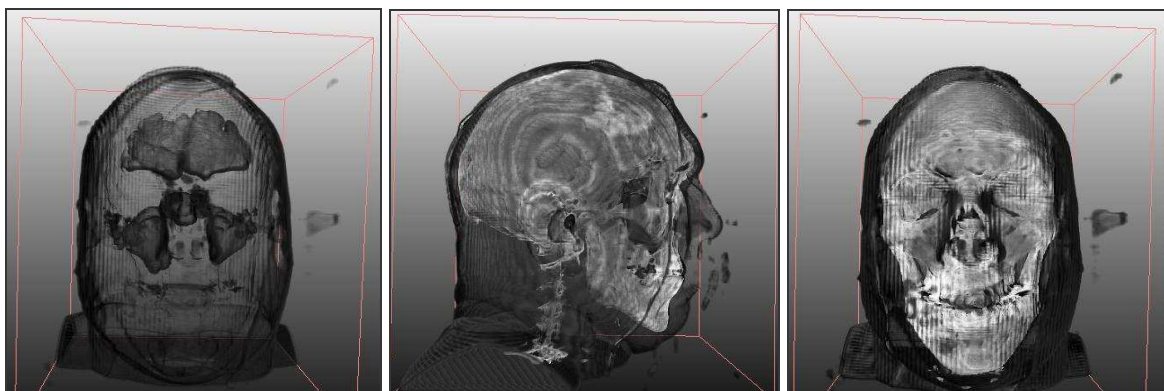


Abbildung 6-2: VR des Visible-Human-Kopfes mit zweidimensionaler TF
Anwendung einer TF zur Visualisierung der Haut (links), Anwendung
einer TF zu Visualisierung der Haut und Knochen (mitte und rechts)

Auch dieser Datensatz ist für den Entwurf zweidimensionaler TFs geeignet. Auch bei diesem Datensatz zeichnen sich die Bögen stark im Histogramm ab. Hierbei können vor allem die Haut und die Knochen gut dargestellt werden. Dabei ist vor allem zu sehen, wie klar und deutlich die einzelnen Flächen dargestellt werden können. Es sind klare Strukturen erkennbar.

6.1.3 Plexiglas-Phantomdatensatz

Ein Datensatz, der sich ebenfalls als gut geeignet erwies, ist der CT-Datensatz einer Plexiglas-scheibe. Dieser Datensatz ist ein Phantomdatensatz, der die Eigenschaften von kontrastierten

Leberdatensätzen simuliert. Der Datensatz wurde von J. Drexler et al. (MeVis) freundlicherweise zur Verfügung gestellt, die in [DKHLFSP04] die Genauigkeit von Gefäßmessungen testen.

Der hier vorliegende Datensatz ist die CT-Aufnahme einer Plexiglasscheibe, siehe Abbildung 6-3. Die Plexiglasscheibe entspricht aufgrund ihrer homogenen Oberfläche in etwa der Oberfläche der Leber. In diese Plexiglasscheibe wurden Löcher mit Durchmessern von 1 mm, 2 mm, 3 mm, 5 mm und 10 mm gebohrt. Die Löcher wurden mit Kontrastmittel aufgefüllt, sodass die Löcher den Gefäßen der Leber entsprechen. 7 Löcher besitzen einen Durchmesser, der größer ist als 2 mm.

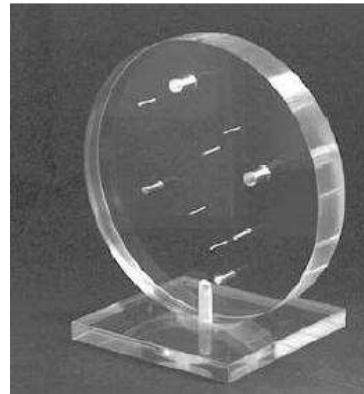


Abbildung 6-3: Fotografie der Plexiglasscheibe

Eigenschaften des Datensatzes

Größe: $512 \times 512 \times 251$, Voxelgröße: $0.74 \times 0.74 \times 0.8$ (cm), Rekonstruktion weich (B30f)

Das VR dieses Phantoms wurde zum einen intensitätsbasiert durch Angabe von einem Zentrum und einer Breite für Grauwert und Transparenz und zum anderen gradientenbasiert erzeugt, siehe Abbildung 6-4.

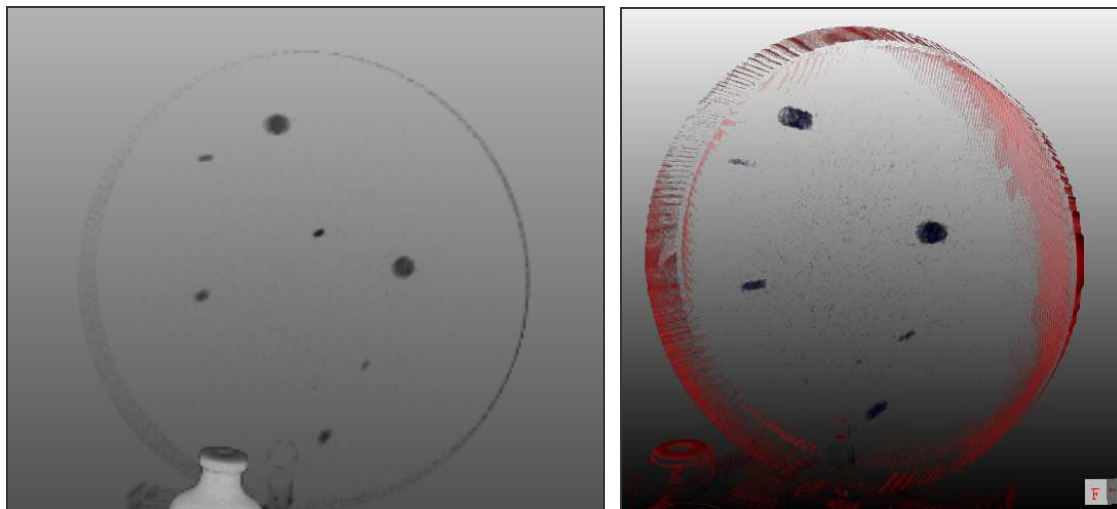


Abbildung 6-4: Die Plexiglasscheibe mit intensitätsbasierter 1D-TF (links) und gradientenbasierter 2D-TF (rechts)

In diesen beiden VRs fiel auf, dass das mittels intensitätsbasierter TF generierte genau die 7 Löcher darstellt, die einen Durchmesser besitzen, der größer als 2 mm ist. Das VR, das mittels gradientenbasierter 2D-TF generiert wurde, stellte nur 6 Löcher dar, was darauf hindeutet, dass dieser Ansatz ungenauer ist. Jedoch stellte sich heraus, dass das Loch, welches fehlt, etwas Kalk enthält. Dies führt dazu, dass der Intensitätswert an dieser Stelle von den Intensitätswerten der anderen Löcher etwas abweicht und sich dies in den Gradienten bemerkbar macht. Die 6 Löcher haben einen geringeren Gradienten als der, der fehlt. Das fehlende Loch kann dargestellt werden, indem eine zusätzliche TF definiert wird, die einen höheren Gradienten einschließt. Somit ist der gradientenbasierte TFs-Entwurf in diesem Fall besser geeignet, weil er genauer ist, als der intensitätsbasierte.

Die Bögen im Histogramm dieses Datensatzes zeichnen sich nicht klar und deutlich ab. Der Bogen, der die 6 Löcher visualisiert ist von geringer Größe. Um diesen Bogen zu finden wurde die Kantendetektionskomponente verwendet, mit deren Hilfe der Bogen schnell gefunden werden konnte.

6.1.4 Ein Schweinedatensatz mit kontrastierter Leber

Hier liegt ein kontrastierter Leberdatensatz eines Schweins vor. Die kleine Voxelgröße und auch der weiche Rekonstruktionsfilter bilden eine ideale Grundlage, um gradientenabhängige TFs zu definieren.

Eigenschaften des Datensatzes:

Größe: $480 \times 264 \times 221$, Voxelgröße: $0.5 \times 0.5 \times 0.8$ (cm), Rekonstruktion: weich (B20f)

In Abschnitt 6.1.3 wurde an einem Phantomdatensatz gezeigt, dass sich theoretisch Lebergefäße gut durch gradientenbasierte 2D-TFs darstellen lassen. Daher ist bei diesem Datensatz das Ziel, die Lebergefäße hervorzuheben.

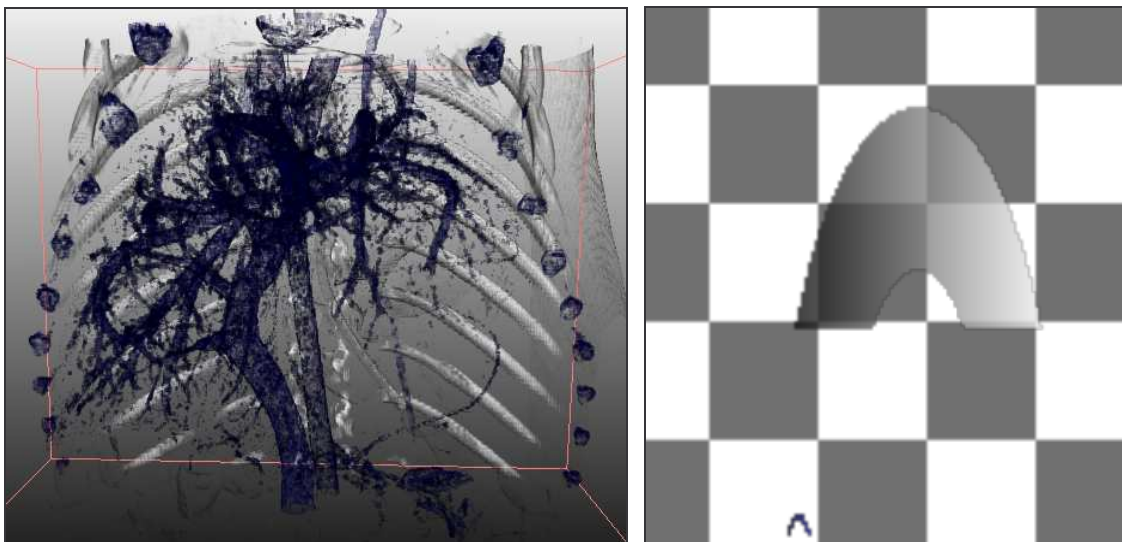


Abbildung 6-5: Lebergefäßvisualisierung eines Schweinedatensatzes

VR des Schweinedatensatzes(links), entsprechende LUT (rechts). Anhand der LUT ist erkennbar, wie klein der Bereich des blauen Bogens zur Darstellung der Lebergefäße ist. Demzufolge ist es schwer, diesen Bogen mit bloßem Auge im Histogramm zu finden.

Im VR dieses Datensatzes konnten die Gefäße der Leber gut visualisiert werden, siehe Abbildung 6-5(links). Durch die Kontextinformationen der Rippen lässt sich die Lage der Lebergefäße abschätzen. Die Kanten wurden mit der Kantendetektionskomponente untersucht, wobei die Bins markiert wurden, die die Lebergefäße repräsentieren. Dadurch wurde der Aufwand erheblich reduziert. An den markierten Bins wurde eine halb elliptische Bogenform angelegt, Breite und Höhe dieser festgelegt und anschließend eine Farbe, ein Farbverlauf, eine Transparenz und ein Transparenzverlauf definiert. Um die Rippen bzw. Knochen darzustellen, musste außerdem noch eine weitere HE-förmige Bogenform angelegt werden. Die Position ließ sich anhand der Intensitätswerte bestimmen. Knochen liegen in einem Be-

reich mit großen HU-Werten. Für diese HEs mussten dann wiederum Farbe und Transparenz festgelegt werden.

In Abbildung 6-5(rechts) ist die LUT, die das VR erzeugt abgebildet. Zu sehen sind ein großer grauer Bogen, der die Rippen visualisiert und ein kleiner blauer Bogen, der die Gefäßdarstellung ermöglicht.

6.2 Visualisierung einer nicht-segmentierten kontrastierten Leber

Hier wurden kontrastierte Leberdatensätze untersucht. Zum einen wurde versucht, die Leber selbst darzustellen und zum anderen die Lebergefäße. Dabei fiel auf, dass vor allem die Versuche die Lebergefäße darzustellen, teilweise zu guten Ergebnissen führten.

6.2.1 Visualisierung der Gefäße und der Leberoberfläche

An einem Beispiel kann die Gefäßvisualisierung verdeutlicht werden. Da die Gefäße der Leber ausreichend kontrastiert sind, heben sie sich stark von der Leber ab. Das führt zu einem Bogen im Histogramm auf dem eine TF definiert werden kann.

Eigenschaften des Datensatzes:

Größe: $365 \times 310 \times 154$, Voxelgröße: $0.625 \times 0.625 \times 1.0$ mm, Rekonstruktionskernel: FC14 (weich rekonstruiert), Toshiba CT

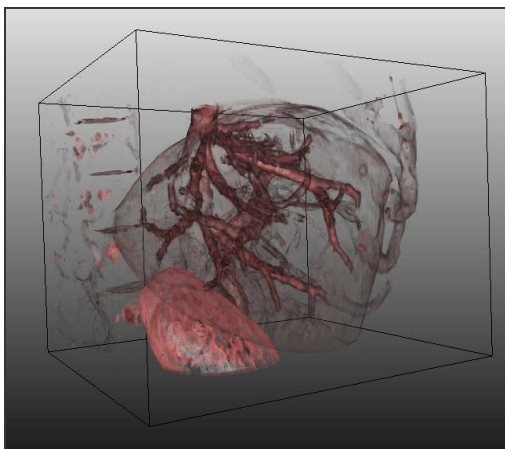


Abbildung 6-6: VR einer kontrastierten Leber mit Darstellung der Lebergefäße

In diesem Beispiel war es möglich, die Leber zu visualisieren. Zu beachten ist, dass der Volumendatensatz nur die Leber selbst und einen kleinen Bereich um die Leber beinhaltet, d.h. evtl. störende Bereiche sind bereits entfernt. Die Gefäße lassen sich in den Datensätzen gut mit den 2D-TFs visualisieren. Die Visualisierung der gesamten Leber ist jedoch schwieriger, da der Kontrast zu dem umgebenden Gewebe geringer ist. Das führt dazu, dass in dem Histogramm nur kleine Bögen entstehen, die teilweise schlecht oder gar nicht selektiert werden können. Außerdem ist der Interaktionsaufwand mittels Trial-and-Error-Verfahren hoch. Es muss viel probiert werden, bis das gewünschte Visualisierungsziel erreicht ist.

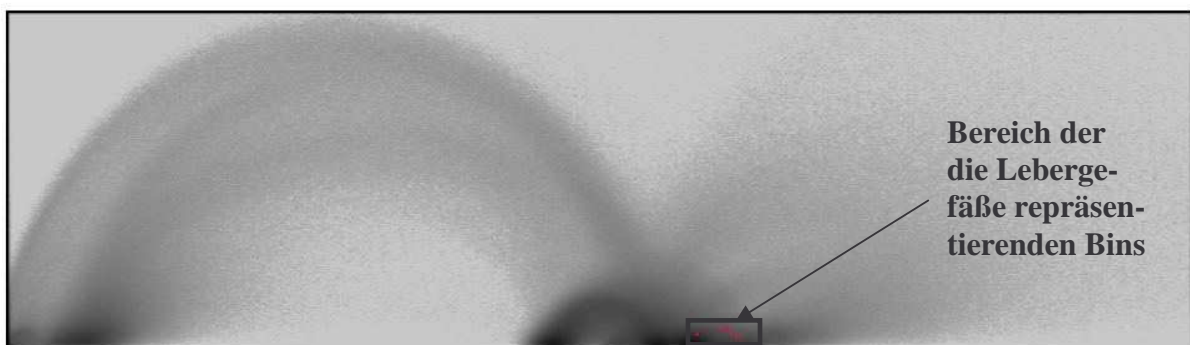


Abbildung 6-7: Histogramm des Leberdatensatzes

Bei allen Leber und Lebergefäßvisualisierungen wurden die Kanten mit der Kantendetektionskomponente erkundet. Dies führte dazu, dass im Histogramm die zu den Voxel korrespondierenden Bins erkannt wurden und eine TFs über die Bins gelegt werden konnte. Wird der Datensatz nicht mit der Kantendetektionskomponente exploriert, so benötigt der Benutzer sehr viel Zeit, um mittels Trial-und-Error zu einer geeigneten TFs zu gelangen.

6.2.2 VR der Lebergefäße mit TFs und mit Segmentierungsmasken

Hier wurde die Eignung der TFs für die Gefäßdarstellung genauer untersucht und mit segmentierten Lebergefäßen verglichen.

Eigenschaften des Datensatzes:

Größe: $336 \times 273 \times 107$, Voxelgröße: $0.684 \times 0.684 \times 1.8$ mm, Rekonstruktionskernel: B31f (weich)

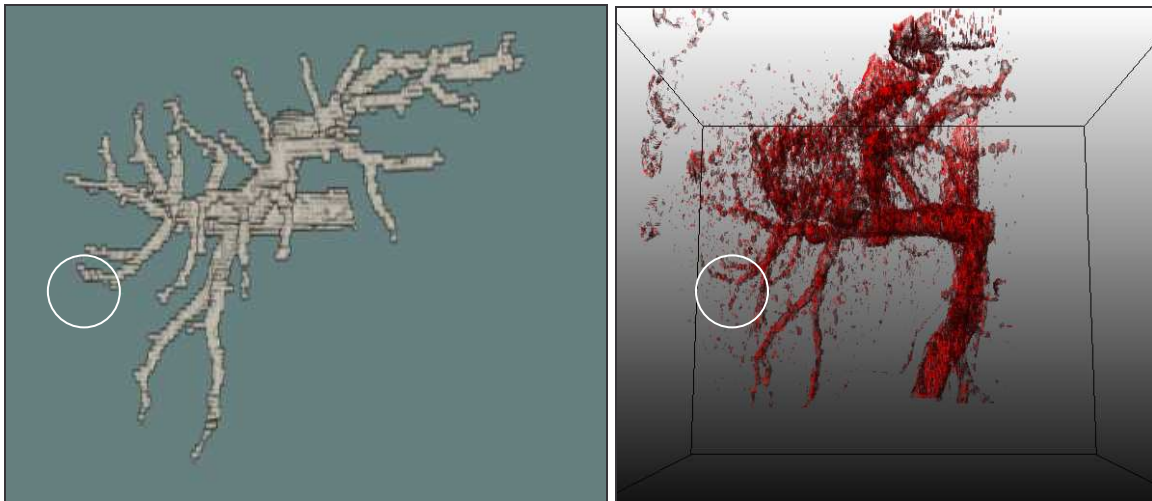


Abbildung 6-8: VR von Lebergefäßen

Links: segmentierte Lebergefäße, Rechts: mit 2D-TFs erzeugte Lebergefäße. Der Kreis verdeutlicht, dass im mittels TFs erstellten VR Verästelungen dargestellt werden können, die bei der Segmentierung verloren gegangen sind.

An diesem Beispiel ist zu sehen, dass die TF teilweise gut eingestellt werden kann. Mitunter können sogar Gefäße gefunden werden, die bei der Segmentierung verloren gegangen sind, siehe Abbildung 6-8. Der Zeitaufwand, um die geeignete TF zu finden, ist mitunter jedoch hoch. Die TF muss fein justiert werden, damit vernünftige Ergebnisse im Volumenrendering entstehen. Um dieses Bild darzustellen, wurden außerdem zusätzliche Funktionen des Renderers ausgenutzt, in dem einzig das Teilvolumen gerendert wurde, in dem nur die Gefäße enthalten sind. Die Frage ist, ob ein Volumenrendering der Gefäße sinnvoll ist. In der Regel will der Arzt zusätzlich den Kontext sehen, d.h. die Leber müsste auch noch dargestellt werden. Soll dies für dieses Beispiel umgesetzt werden, so entsteht zu viel Rauschen im VR, so dass kaum noch etwas erkannt werden kann, siehe Abbildung 6-9

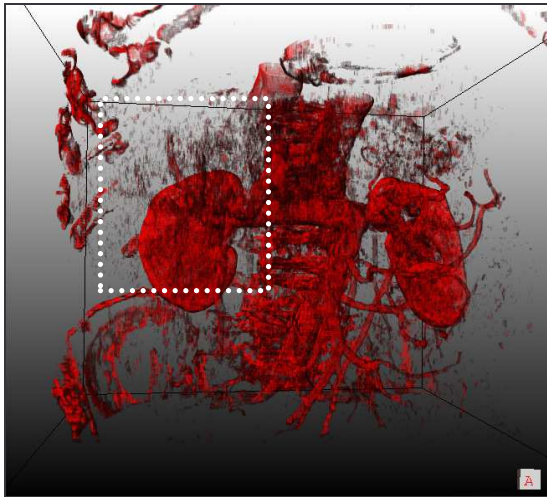


Abbildung 6-9: Lebergefäße mit Kontextinformation.
Der markierte Bereich stellt den Ausschnitt aus Abbildung 6-8 links dar.

In diesem Beispiel treten bei dem Rendering des gesamten Volumens die Nieren zu stark hervor. Sie enthalten genauso viel bzw. etwas mehr Kontrastmittel als die Leber und liegen somit in dem gleichen Definitionsbereich der TF. Die Lebergefäße sind durch die feinen Strukturen nur schwach dargestellt und sind kaum noch erkennbar. Der Arzt kann mit dieser Art der Visualisierung keine Operation planen. Der Verlauf und die Lage der Lebergefäße sind nur schwer bis gar nicht einschätzbar.

6.3 Visualisierung der Lungenbronchien und -gefäße

In diesem Abschnitt wird untersucht, ob die Visualisierung von Lungenbronchien und -gefäßen mittels der Definition von 2D-TFs möglich ist. Hierbei ist zu beachten, dass die Bronchien sehr klein sind und nur in hart rekonstruierten Datensätzen klar und deutlich zu erkennen sind. Deshalb wurden für dieses Visualisierungsziel drei verschiedene Datensätze betrachtet, die weich, weich bis hart und hart rekonstruiert wurden.

Eigenschaften der Datensätze:

Datensatz 1: Größe: $76 \times 76 \times 43$, Voxelgröße: $0.57 \times 0.57 \times 1$ mm, Rekonstruktionskernel: weich (B30f)

Datensatz 2: Größe: $76 \times 76 \times 43$, Voxelgröße: $0.57 \times 0.57 \times 1$ mm, Rekonstruktionskernel: weich-hart (B40f)

Datensatz 3: Größe: $76 \times 76 \times 43$, Voxelgröße: $0.83 \times 0.83 \times 2$ mm, Rekonstruktionskernel: sehr hart (B50f)

In den nächsten vier Abbildungen basiert das linke Bild auf dem weich-rekonstruiertem Datensatz, das mittlere auf dem weich- bis hart-rekonstruiertem und das rechte auf dem hart-rekonstruiertem. Ziel ist das VR der Bronchien.

In Abbildung 6-10 sind die Histogramme mit der definierten TF dargestellt. Es ist erkennbar, dass die Stärke der Kanten zunimmt. Die Bögen haben unterschiedliche Höhen. Außerdem sieht das linke Histogramm etwas verrauschter aus als die beiden anderen. In Abbildung 6-11 ist eine Schichtdatendarstellung des jeweiligen Datensatzes mit den von der TF eingeschlossenen Voxeln abgebildet. In Abbildung 6-12 ist das VR der Lungenbronchien mittels einer intensitätsbasierten TF dargestellt. Hierbei ist erkennbar, dass in dem hart-rekonstruierten Datensatz das VR stark verrauscht ist. Das Rauschen des hart-rekonstruierten Datensatzes spiegelt sich auch in dem mit gradientenabhängigen TFs erzeugten VR wieder, siehe Abbildung 6-13, in der das linke VR verrauschter ist als die beiden anderen.

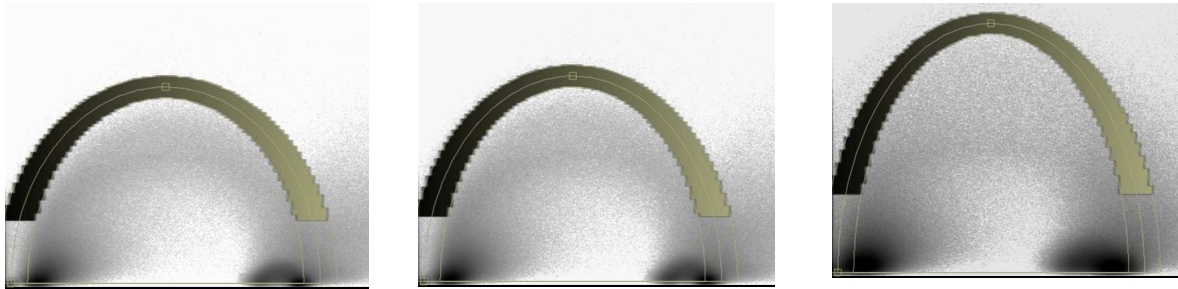


Abbildung 6-10: Histogramm überlagert mit der TF

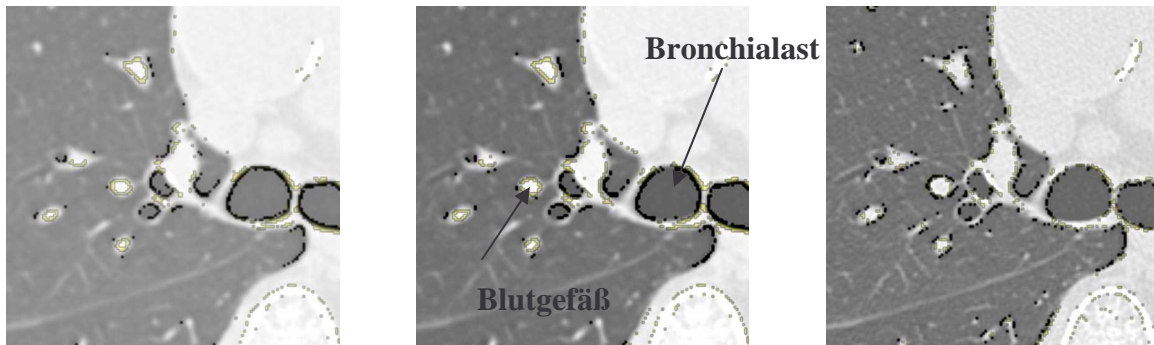


Abbildung 6-11: Schichtdarstellung des Datensatzes mit Färbung der von der TF dargestellten Voxel



Abbildung 6-12: VR der Bronchien mit intensitätsbasierter TF (Rampe mit Zentrum und Breite)



Abbildung 6-13: VR der Bronchien mit der in Abbildung 6-10 definierten gradientenabhängigen TF

links: weich-rekonstruiert, mitte: weich-hart-rekonstruiert, rechts: hart-rekonstruiert

In allen VRs der drei Datensätze mittels gradientenabhängiger TF konnten die großen Bronchien dargestellt werden, jedoch ist es nur schwer möglich zusätzlich die kleinen Bronchien ohne die Blutgefäße darzustellen. Die Definition dieser TF war hier nicht einfach zu realisierbar. Hier konnte die Kantendetektionskomponente nicht richtig weiter helfen. Durch den Partialvolumeneffekt gehen Strukturen wie die Bronchien nicht verloren, jedoch ändert sich der Intensitätswert und er überlagert sich bzw. schneidet sich mit den Intensitätswerten der Blutgefäße. In Abbildung 6-11 (in der Mitte) ist zum einen der Bronchialast von der LUT eingeschlossen, zum anderen ist erkennbar, dass zusätzlich auch Blutgefäße von der TF erfasst werden. Die Bronchien lassen sich nur schwer von den Blutgefäßen differenzieren. Kommt hier die Kantendetektionskomponente zum Einsatz, so wird kein eindeutiger Bogen gezeichnet. Es kann nicht genau eingeschätzt werden, wo der Bogen verläuft und welche Breiten und welche Höhe eingestellt werden müssen.

Sollen die kleinen Bronchien visualisiert werden, so kann ein Bogen definiert werden, der einen größeren Bereich abdeckt, siehe Abbildung 6-14 (a). Der HE-förmige Bogenfläche umfasst einen großen Bereich. Im Schichtdatensatz (b) ist erkennbar, dass große Bronchien, kleine Bronchien und alle Blutgefäße von diesem Bogen erfasst werden. Im VR (c) ist es nicht möglich, Bronchien von den Blutgefäßen zu unterscheiden.

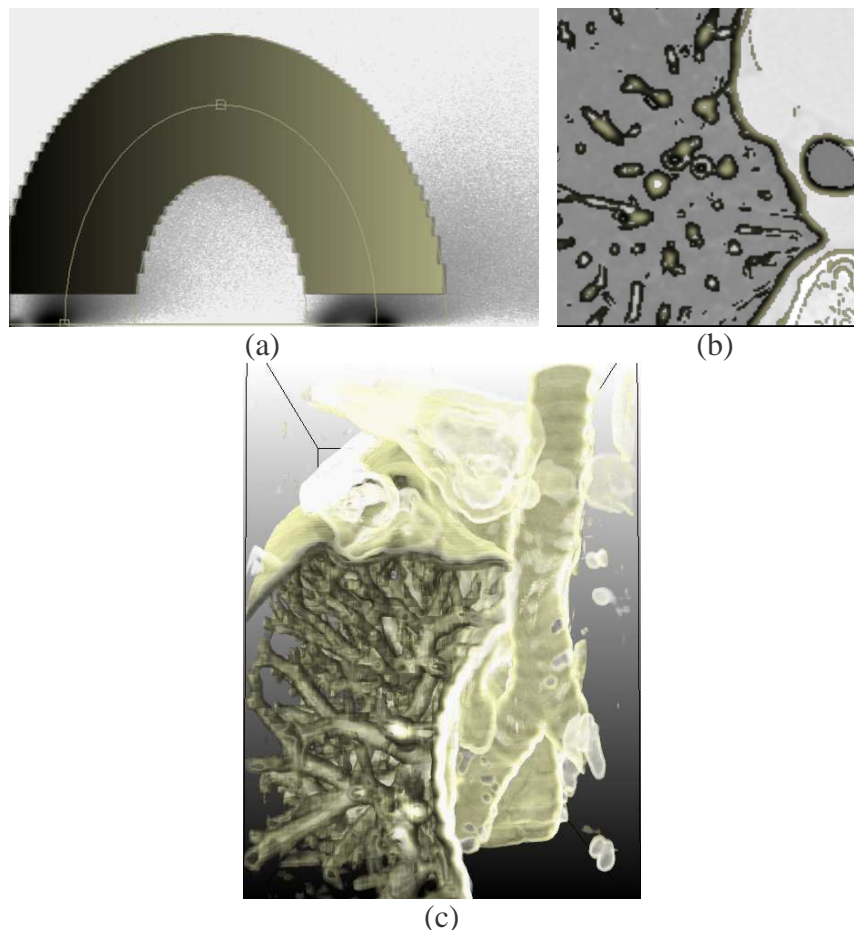


Abbildung 6-14: Definition einer TF für das VR großer und kleiner Bronchien und der Lungengefäße in einem weichrekonstruierten Datensatz

Dass die Bronchien nicht gut von den Blutgefäßen unterschieden werden können, kann anhand der Histogramme dieser Datensätze erklärt werden. In Abbildung 6-15 ist das

Histogramm eines der Datensätze abgebildet und die zur Darstellung der Gefäße und Bronchien relevanten Bögen sind eingezeichnet. In diesem Histogramm wird deutlich, dass die beiden relevanten Bögen sich überschneiden. Der große Bogen repräsentiert den Übergang von Luft zu den Lungenblutgefäßen und der kleine den von Luft zu den Bronchien. Beide Bögen beginnen im selben Fußpunkt und überschneiden sich. Das führt dazu, dass der kleine Bogen die Darstellung der Gefäße beeinflusst und umgekehrt beeinflussen die Gefäße die Darstellung der Bronchien. Dieses Problem kann auch nicht in der Art behoben werden, dass dieser Bereich weggelassen wird, da dies dazu führt, dass die Gefäße kaum noch dargestellt werden. Dass der Bogen der Bronchien einen großen Bereich umfasst, liegt vor allem am Partialvolumeneffekt. Dadurch werden die kleinen Bronchien nicht korrekt abgebildet und die HU-Werte und Gradientenwerte variieren. Das führt dazu, dass die korrespondierenden Bins der kleinen Bronchien sich hauptsächlich in dem linken unteren Bereich des Bogens bei -1000 HU bis -700 HU befinden statt bei ca. 50 HU.

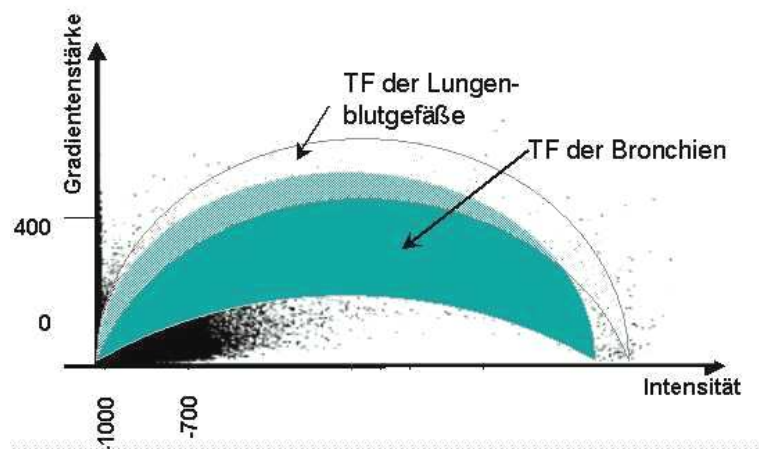
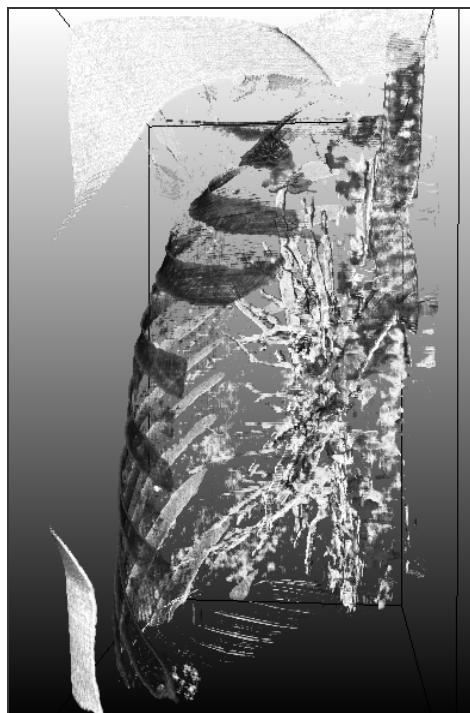


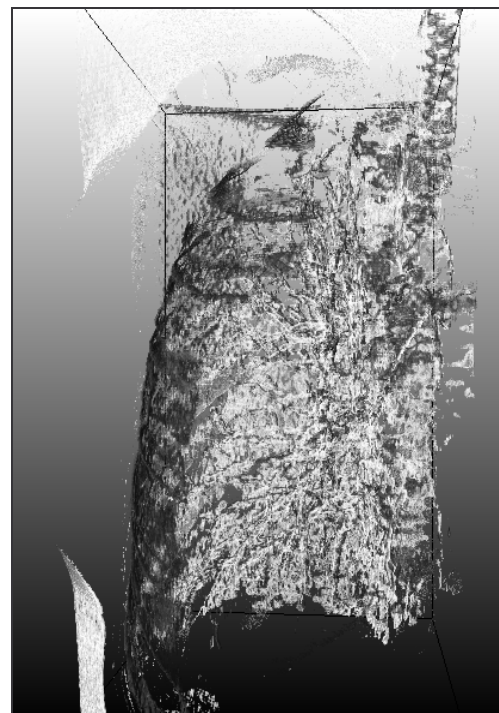
Abbildung 6-15: Histogramm des weich-rekonstruierten Lungendatensatzes

Lungengefäße:

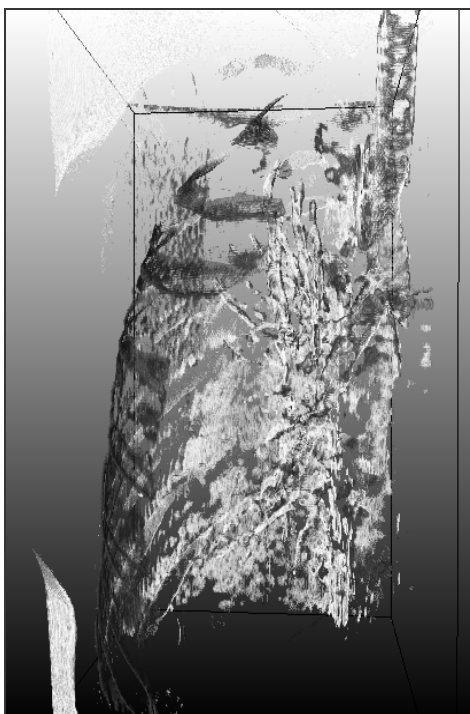
Die Lungenbronchien ohne Visualisierung der Blutgefäße darzustellen, erwies sich als kompliziert und schwer realisierbar. Außerdem ist die Qualität der Darstellung stark von dem verwendeten Rekonstruktionsfilter abhängig. In Abbildung 6-16 sind Lungenvisualisierungen der Gefäße dargestellt. Hier sind Visualisierungen verschieden rekonstruierter und gefilterter Datensätze gegenübergestellt. Die Gefäße des weich-rekonstruierten Datensatzes können im VR gut dargestellt werden, während bei dem hart-rekonstruierten viel Rauschen im VR auftritt und keine deutlichen Kanten dargestellt werden können. Wird der hart-rekonstruierte Datensatz mittels Medianfilter vorverarbeitet, so bessert sich das VR, welches durch anschließendes Gaußfiltern noch weiter verbessert werden kann.



weich-rekonstruiert



hart-rekonstruiert



hart-rekonstruiert + mediangefiltert

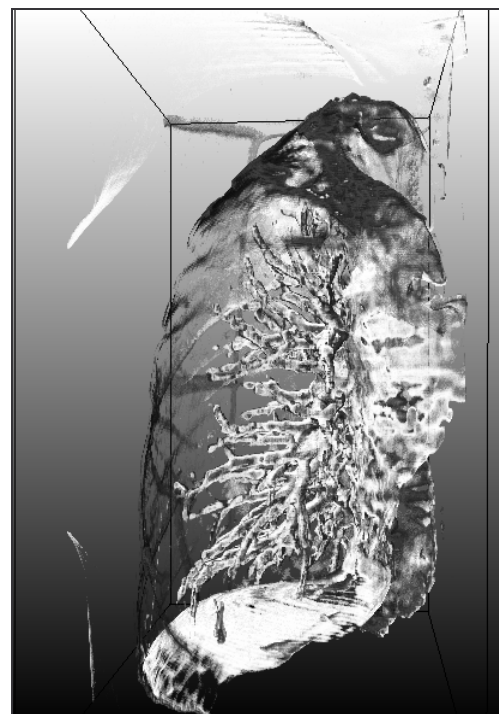
hart-rekonstruiert + mediangefiltert +
gaußgefiltert

Abbildung 6-16: Lungenvisualisierungen verschieden rekonstruierter und vorverarbeiteter Datensätze mittels gradientenabhängiger TF

6.4 Visualisierung der Knochen

Mit gradientenbasierten TFs kann eine gute Visualisierung der Knochen erreicht werden. Die Knochen können in den Datensätzen als Kontextinformation eingeblendet oder ausgeblendet

werden. Das Ausblenden der Knochen funktioniert jedoch nicht immer. So liegen die Intensitäts- und Gradientenwerte der Knochen häufig in den gleichen Wertebereichen wie kontrastierte Gefäße, was wiederum dazu führt, dass die Knochen mit derselben TF wie die Gefäße dargestellt werden. Dies tritt beispielsweise bei kontrastierten Leberdatensätzen auf, wo Lebergefäße und Rippen mit der Definition des gleichen Bogens visualisiert werden.

Eigenschaften des Datensatzes:

Datensatz 1: Größe: $512 \times 512 \times 209$, Voxelgröße: $0.57 \times 0.57 \times 1$, Rekonstruktionskernel: weich (B30f)

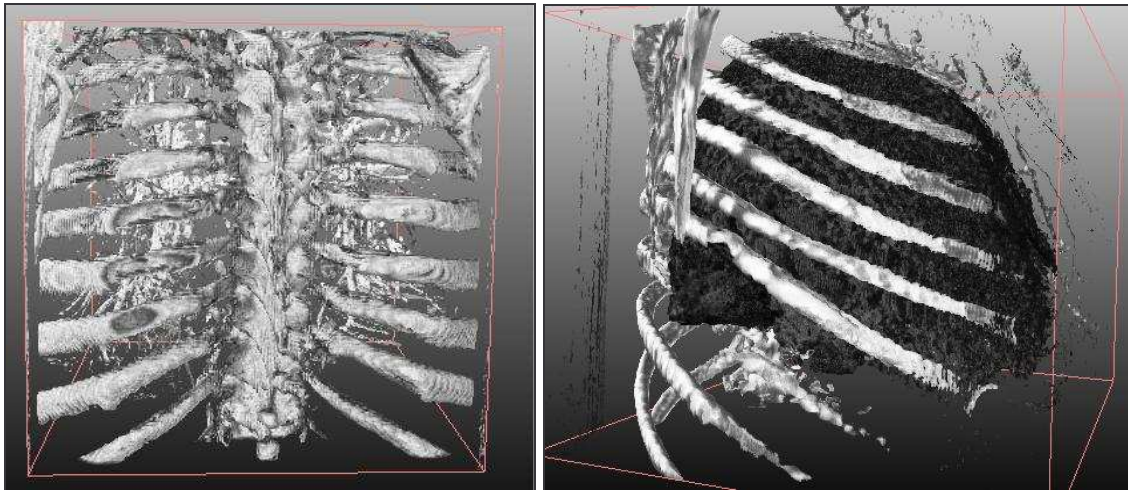


Abbildung 6-17: Knochenvisualisierung mittels gradientenabhängiger TFs
Links: Wirbelsäule mit Rippen, Rechts: Rippen- und Lungenvisualisierung

6.5 Abdomendatensatz

Interessant ist die Visualisierung eines kompletten Abdomendatensatzes. Hierbei ist wichtig, zu untersuchen, was im VR dargestellt werden kann. Ist es beispielsweise möglich, die Organe im VR darzustellen!

Eigenschaften des Datensatzes:

Datensatz 1: Größe: $512 \times 512 \times 209$, Voxelgröße: $0.57 \times 0.57 \times 1$, Rekonstruktionskernel: weich (B30f), nicht kontrastiert

In Abbildung 6-18 sind mögliche Visualisierungen dargestellt. Zum einen kann das Skelett (a) gut dargestellt werden, zum anderen sind auch Darmvisualisierungen (b, c) an den Stellen möglich, an denen der Darm mit Luft gefüllt ist. Diese Darstellungen (Knochen und Darm) sind möglich, da hier ein großer Gradient vorliegt und die Bögen im Histogramm leicht angelegt werden können. In (d) wurde für die TFs-Definition ein weiterer Bogen hinzugefügt. Dieser repräsentiert den Übergang von Muskeln und Fett. Es ist erkennbar, dass die Qualität des VRs abnimmt und Strukturen kaum noch wahrgenommen werden können. In (e) sind nur noch das Fett, Bindegewebe und die Muskeln dargestellt. Die Muskeln können gerade noch erkannt werden, jedoch ist das Ergebnis nicht ausreichend. Die Organe können mit diesem

Verfahren nicht dargestellt werden. Dazu ist der Kontrast zwischen Fett, Muskeln und Organen zu gering. Das Fett verdeckt die Organe. Sie können nicht dargestellt werden.

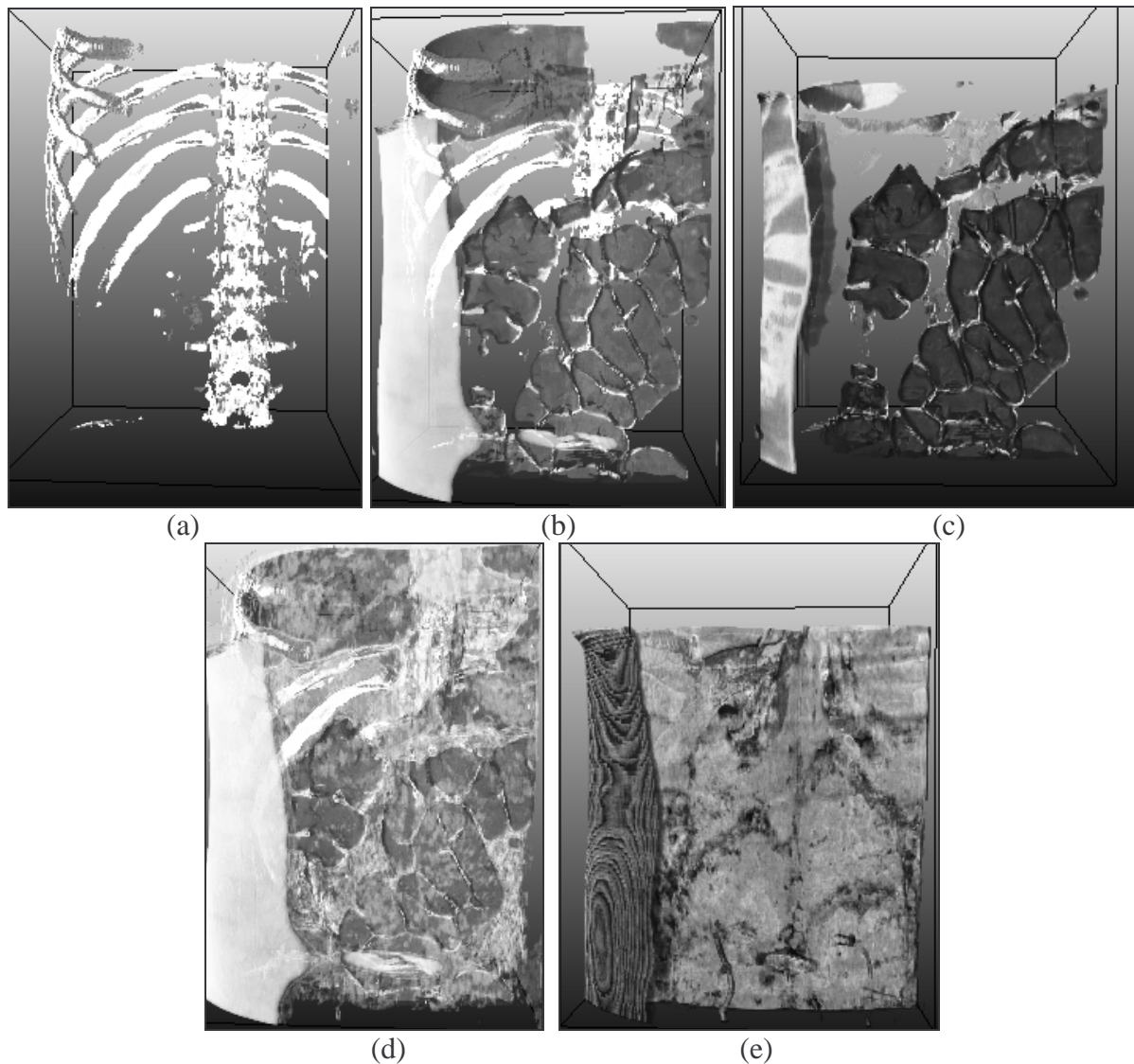


Abbildung 6-18: VR eines Abdomendatensatzes mittels gradientenbasierter TFs

6.6 MRT-Daten

Es musste festgestellt werden, dass bei MRT-Daten kein geeignetes Volumenrendering mittels der hier vorgestellten TFs erzeugt werden kann. Durch hohe Intensitätsschwankungen und Inhomogenitäten der MRT-Bilder werden im Histogramm keine klaren Bögen sichtbar. Es ist möglich Bögen zu setzen und unter Betrachtung der Schichtbilder scheinen auch einige geeignet zu sein, jedoch wird im Volumenrendering kein scharfes Bild sichtbar. In den MRT-Bildern ist von vornherein ein zu starkes Rauschen und zu viele Inhomogenitäten.

Es wurde versucht mittels Vorfilterung zu besseren Ergebnissen zu gelangen. In Abbildung 6-19 ist ein realer medizinischer T1-gewichteter Kopfdatensatz visualisiert, wobei der Datensatz des rechten Bildes mittels Median- und Gaußfilter vorverarbeitet wurde. In beiden Bil-

dem ist erkennbar, dass die 2D-TFs keine geeignete Visualisierung ermöglichen. Strukturen wie das Gehirn können nicht hervorgehoben werden.

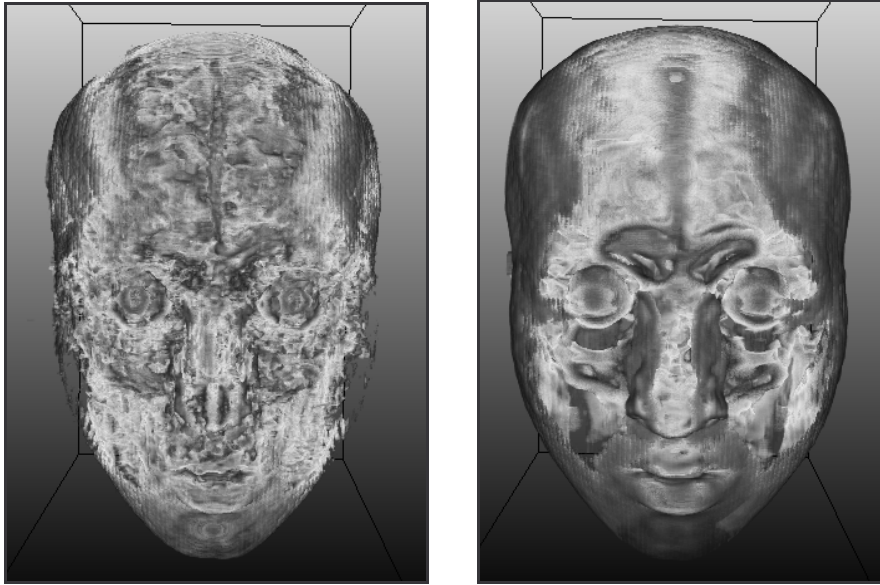


Abbildung 6-19: VR eines realen medizinischen Kopf-Datensatzes
Links: unbearbeiteter MRT-Datensatz, Rechts: median- und gaußgefiltert. Die Kopfoberfläche kann visualisiert werden, jedoch sind innere Strukturen wie die Gehirnmasse nicht erkennbar.

Bisher konnten einzig mit einem MRT-Phantomdatensatz der Datenbank BrainWeb [CZK98] brauchbare VRs erzeugt werden, siehe Abbildung 6-20.

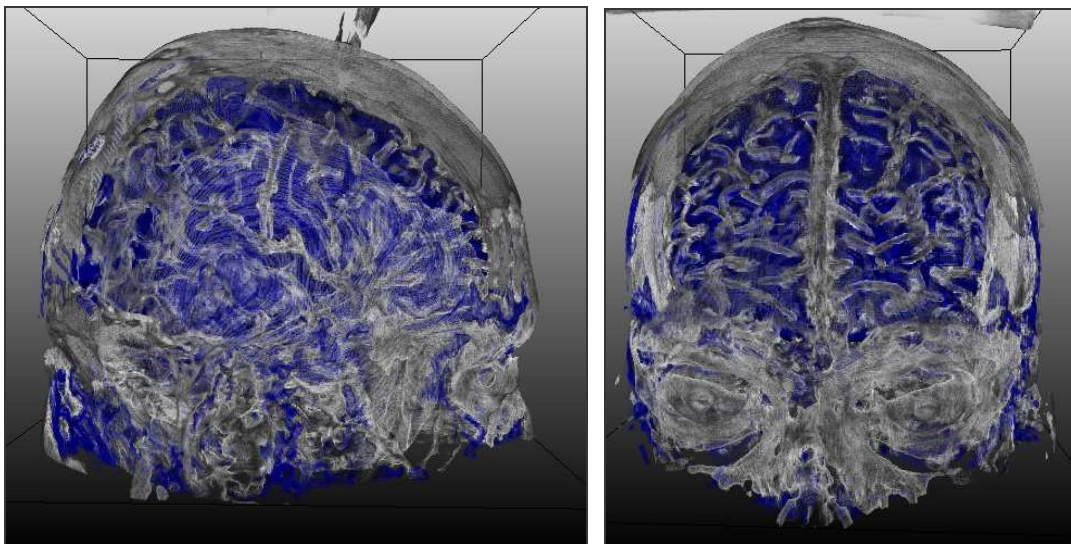


Abbildung 6-20: VR eines Phantom-MRT-Kopfdatensatzes
Die Gehirnmasse ist gut erkennbar.

7 Zusammenfassung und Ausblick

Das VR medizinischer Bilddaten mittels der Definition von TFs ist eine große Herausforderung. Wenn es möglich ist, einfach und schnell geeignete TFs zu definieren, bietet diese Visualisierungsmethode eine gute Möglichkeit der Unterstützung von Radiologen in Diagnostik und Therapie, beispielsweise zur Planung von Operationen. Ziel dieser Arbeit war es, kantendetektierende 2D-TFs zu entwickeln und deren Eignung für das medizinische VR zu untersuchen.

Die Untersuchung bisheriger Verfahren zeigte, dass der Ansatz von Gordon Kindlmann, der mittels 3D-TFs ein VR erzeugt, eine geeignete Grundlage für den Entwurf kantendetektierender TFs darstellt. Sein Ansatz basiert dabei auf der Auswertung eines 3D-Histogramms. In dem in dieser Arbeit vorgestellten Entwurf gradientenabhängiger TFs wurden anhand der Auswertung eines 2D-Histogramms, welches die Verteilungen der Intensitäts- und Gradientenwerte charakterisiert, TFs definiert. Dazu wurden erkennbare Strukturen des Histogramms untersucht und anhand dessen eine Repräsentation für TFs entwickelt.

In Abbildung 7-1 ist der Prozess der TFs-Definition in einer Pipeline zusammengefasst. Zuerst wird auf dem tomographischen Datensatz ein Gradientenstärkebild erzeugt und anschließend ein 2D-Histogramm berechnet. Anhand von erkennbaren Strukturen werden TFs definiert, woraufhin eine LUT berechnet werden kann. Die LUT wird dem Renderer übergeben, der anschließend das Ergebnisbild erzeugt.

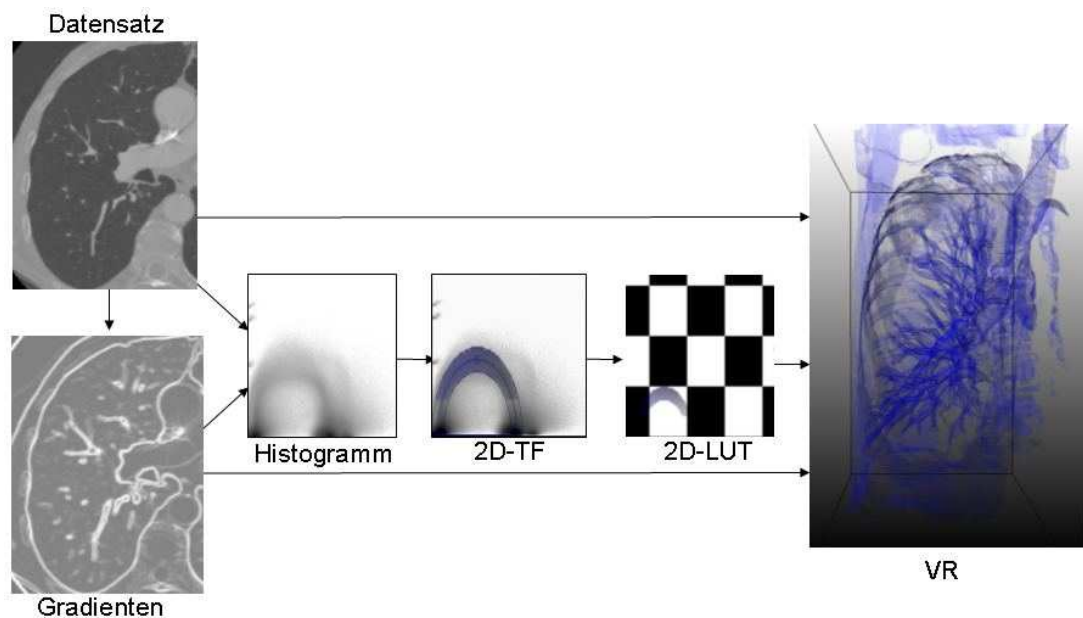


Abbildung 7-1: Vom skalaren 3D-Feld zum VR

Anhand von Beispiel-Datensätzen konnte gezeigt werden, dass HE-förmige Bogenflächen für die Repräsentation der 2D-TFs gut geeignet sind. In medizinischen Datensätzen variiert die Schärfe der Bögen in Abhängigkeit der verwendeten Rekonstruktionskernel und der Voxelgröße. Die Anwendung der TFs auf verschiedene Datensätze zeigte, dass weich-rekonstruierte Datensätze am besten für die Definition gradientenabhängiger TFs geeignet sind und bei hart-rekonstruierten Datensätzen die Möglichkeiten stark eingeschränkt sind.

Daraus ergeben sich Einschränkungen für die Visualisierung. So können bei weichrekonstruierten Datensätzen Einzelheiten in den Strukturen wie beispielsweise kleine Verästelungen der kleinen Bronchien oder der Lebergefäße durch den Partialvolumeneffekt verloren gehen. Dies macht sich im VR in der Art bemerkbar, dass diese kleinen Strukturen nur schlecht dargestellt werden können und ihre Kanten nicht deutlich hervortreten. Das VR stellt diese Strukturen nur schlecht dar.

Um den Entwurf zweidimensionaler TFs zu unterstützen, wurde eine Kantendetektionskomponente entwickelt, die dazu beiträgt auf intuitive Weise die Bögen in den Histogrammen zu finden. Die Definition von gradientenabhängigen TFs wird dadurch stark vereinfacht. Auf diese Weise können Bögen im Histogramm gefunden werden, die im Histogramm schlecht oder gar nicht erkennbar sind. Mithilfe dieses Tools konnten u.a. die korrespondierenden Bins der Lebergefäße gefunden und eine TF erstellt werden.

Es konnte gezeigt werden, dass der gradientenbasierte Entwurf 2D-TFs gut geeignet ist, um Knochen, die Gefäße der Lunge und Luft-Haut-Übergänge darzustellen. Die Visualisierung der Organe und aller anderen Weichteile dagegen ist nur schwer (mit Kontrastmittel) bzw. gar nicht realisierbar. In diese Richtung muss weiter geforscht und nach neuen Ansätzen gesucht werden.

In dieser Arbeit wurde außerdem versucht, einen automatischen Ansatz zur Suche der Bögen zu entwickeln. Bei synthetischen Daten funktioniert dies ganz gut, jedoch konnte gezeigt werden, dass die Bögen in medizinischen Datensätzen nur schlecht gefunden werden können, weil sie nur undeutlich in den Histogrammen hervortreten. Der Algorithmus kann noch weiter ausgebaut werden. Bisher müssen viele Parameter von Hand eingestellt werden. Der Algorithmus kann dahingehend ausgebaut werden, diese Parameter automatisch anzupassen.

Auch die Definition von Presets ist bei gradientenabhängigen TFs nicht leicht realisierbar. Es konnte gezeigt werden, dass es durchaus sinnvoll sein kann, für einige Visualisierungsziele Presets zu definieren, jedoch müssen diese aufgrund der unterschiedlich rekonstruierten Datensätze angepasst werden. Die Höhen aller im Preset definierten Bögen müssen skaliert werden. Dies ist vielleicht in der Art möglich, dass die Bögen automatisch skaliert werden, indem der Skalierungsfaktor aufgrund einer Histogrammanalyse berechnet wird.

8 Literaturverzeichnis

- [AN95] A. S. Aguado und M. S. Nixon: „A new Hough-Transformation for Ellipse Detection”, In *Research Journal Image, Speech and Intelligent Systems*, University of SouthHampton, 1995
- [BPS97] Chandrajit L. Bajaj und Valerio Pascucci und Daniel R. Schikore: „The Contour Spectrum”, In *Proceedings IEEE Visualization*, S. 167-173, 1997
- [CKLG98] Silvia Castro und Andreas König und Helwig Löffelmann und Eduard Gröller: „Transfer Function Specification for the Visualization of Medical Data”, Insitute of Computer Graphics, Vienne University of Technology, März 1998
- [CZK98] D.L. Collins und A.P. Zijdenbos und V. Kollokian et al.: „Design and construction of a realistic digital brain phantom”, In *IEEE Trans Med Imaging*, Vol. 17, S. 479-489, 1998
<http://www.bic.mni.mcgill.ca/brainweb>
- [DGMP04] Deutsche Gesellschaft für medizinische Physik: „Leitfaden zur Bewertung und Optimierung der Strahlenexposition bei CT-Untersuchungen“, Juli 2004
http://www.dgmp.de/Page_Service/Leitfaden_CT_022004.pdf
- [DKHLFSP04] Drexl J. und Knappe V. und Hahn HK und Lehmann K. und Frericks BB. und Shin H. und Peitgen H.: „Accuracy analysis of vessel segmentation for a LITT dosimetry planning system“, In *Medical Robotics, Navigation and Visualization (MRNV 2004)*, March 11-12. 2004, Remagen
- [Far90] Gerald Farin: *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, San Diego, 1990
- [FBT98] Shiaofen Fang und Tom Biddlecome und Mihran Tuceryan: „Image-Based Transfer Function Design for Data Exploration in Volume Visualisation”, In *IEEE Visualization*, S. 319-326, 1998
- [Fol90] J. D. Foley and A. vanDam, S. K. Feiner, and J. F. Hughes: *Computer Graphics: Principles and Practices (2nd Edition)*, AddisonWesley, 1990
- [Gal00] Michael Galanski, *Interaktive Volumendarstellungsverfahren für die Schnittbildgebung*, Abteilung Diagnostische Radiologie I, Medizinische Hochschule Hannover, Forschungsbericht 2000
- [HHKP96] Taosong He und Lichan Hong und Arie Kaufman und Hanspeter Pfister: „Generation of transfer functions with stochastic search techniques”, In *Proceedings of IEEE Visualization*, S. 227-234, 1996.

- [HLP03] H.K. Hahn, F. Link und H.-O. Peitgen: "Concepts for Rapid Application Prototyping in Medical Image Analysis and Visualization", In *SV - Simulation und Visualisierung*, 2003
- [HRH03] Dieter Hönigmann und Johanne Ruisz und Christoph Haider: „Adaptive Design of a Global Opacity Transfer Function for Direct Volume Rendering of Ultrasound Data”, Vol. 17 No. 1, In *Proceedings IEEE Visualization*, 2003
- [HSCSM00] Jian Huang und Naeem Shareef und Roger Crawfis und Ponnuswamy Sadayappan und Klaus Mueller: „A Parallel Splatting Algorithm With Occlusion Culling”, In *3rd Eurographics Workshop on Parallel Graphics and Visualization*, September 2000
- [KG01] A. König und E. Gröller: „Mastering Transfer Function Specification by Using VolumePro Technology”, In *Proc. Spring Conference on Computer Graphics*, 2001.
- [KHK02] Euijin Kim und Miki Haseyama und Hideo Kitajima: „Fast and Robust Ellipse Extraction from Complicated Images”, No 138-5, In *ICITA*, 2002
- [Kin03] Benjamin King: *Intuitive Einstellung von Transferfunktionen für das direkte Volume Rendering*, Diplomarbeit an der Universität Hannover, Juli 2003
- [Kin98] Gordon Kindlmann and James W. Durkin: „Semi-automatic generation of transfer functions for direct volume rendering” In *Proceedings of IEEE Volume Visualization*, S. 79-86, 1998.
- [Kin99] Gordon Kindlmann: *Semi-automatic generation of transfer functions for direct volume rendering*, Master Thesis, Program of Computer Graphics, Cornell University
- [KK01] Joe Kniss und Gordon Kindlmann: „Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets“, In *IEEE Visualization*, Presentation, 2001
- [KKH01] Joe Kniss und Gordon Kindlmann und Charles Hansen: „Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets“, In *IEEE Visualization*, S. 255-262, 2001
- [KKH02] J. Kniss und G. Kindlmann und C. Hansen: „Multidimensional Transfer Functions for Interactive Volume Rendering“, In *IEEE Transactions on Visualization and Computer Graphics*, Vol. 8, No. 3, S. 270--285, 2002
- [KM94] Victor D. Köchli und Borut Marincek: *Wie funktioniert MRI*, Springer-Verlag Berlin Heidelberg, 1994

- [KPHSM03] Joe Kniss und Simon Premože und Charles Hansen und Peter Shirley und Allen McPherson: „A Model for Volume Lighting and Modeling”, In *IEEE Transactions on Visualisation and Computer Graphics*, 2003
- [KPILH] Joe Kniss und Simon Premože und Milan Ikits und Aaron Lefohn und Charles Hansen und Emil Praun: „Gaussian Transfer Functions for Multi-Field Volume Visualisation“, Vol. 17 No. 2, In *IEEE Visualization*, 2003
- [KWTM03] Gordon Kindlmann und Ross Whitaker und Tolga Tasdizen und Torsten Möller: „Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications“, Vol. 17 No. 4, In *IEEE Visualization*, 2003
- [LC87] W. Lorensen und H. Cline: „Marching Cubes: A High ResoLUTion 3D Surface Construction Algorithm“, In *Computer Graphics*, Vol. 21 No. 4, S. 163–169, 1987
- [Lev88] M. Levoy: „Display of Surfaces form Volume Data” *IEEE Comp. Graph. & Appl.*, Vol. 8 No. 5, S. 29-37, 1988.
- [LFB98] D. H. Laidlaw, K. W. Fleischer, and A. H. Barr, „Partial-volume Bayesian classification of material mixtures in MR volume data using voxel histograms”, In *IEEE Transactions on Medical Imaging*, Vol. 17, No. 1, S. 74-86, Feb. 1998
- [Lin04] Florian Link: „View2DExtensions”, <http://www-intern.cevis.uni-bremen.de>, Juli 2004
- [LL94] P. Lacroute and M. Levoy: „Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transform”, In *Comp. Graphics*, 28(4), 1994
- [Mai02] Tomáš Mainzer: „Genetic Algorithm for Ellipse Detection”, Department of Computer Science and Engineering, University of West Bohemia, Univerzitní 8, 30614 Pilsen, Czech Republic, 2002
- [Mar97] J. Marks und B. Andalman und P.A. Beardsley und W. Freeman und S. Gibson und J. Hodgins und T. Kang und B. Mirtich und H. Pfister und W. Ruml und K. Ryall und J. Seims und S. Shieber: „Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation”, In *Proceedings of ACM SIGGRAPH*, S. 389-400, 1997
- [ML94] Stephen R. Marschner and Richard J. Lobb: „An Evaluation of Reconstruction Filters for Volume Rendering”, In *IEEE Visualisation*, 1994
- [MLab04] MeVisLab Homepage: <http://www.MeVisLab.de>, Juli 2004
- [Mor95] Heinz Morneborg: *Bildgebende Systeme für die medizinische Diagnostik*, Publicics MCD Verlag, 3. Aufl., 1995

- [MWW04] Medicine Worldwide: <http://www.m-ww.de>, Juli 2004
- [Nov94] K. Novins. *Towards Accurate and Efficient Volume Rendering*. PhD thesis, Cornell University, 1994.
- [PLB01] H. Pfister und B. Lorensen und C. Bajaj und G. Kindlmann und W. Schroeder und L. Avila und K. Martin und R. Machiraju und J. Lee: „The Transfer Function Bake-off“, In *IEEE Computer Graphics & Applications*, Vol. 21 No. 3, S. 16-22, 2001.
- [PP02] Peter Prinz und Ulla Kirch-Prinz. *C++ Lernen und professionell anwenden*, 2. vollständige Auflage, mitp-Verlag, Bonn, 2002
- [PP04] Bernhard Preim und Heinz-Otto Peitgen: *Medizinische Visualisierung: Methoden und Anwendungen in der Ausbildung und Therapieplanung*, 2004
- [PWH01] Vladimir Pekar und Rafael Wiemker und Daniel Hempel: „Fast Detection of Meaningful Isosurfaces for Volume Data Visualization“ In *IEEE*, 2001
- [Rad04] Radiologieforum: <http://www.radiologieforum.de>, Juli 2004
- [Rez01] Christoph Rezk-Salama: *Volume Rendering Techniques for General Purpose Graphics Hardware*. Dissertation, Universität Erlangen-Nürnberg, Erlangen, Deutschland, 2001.
- [RS01] Christof Rezk-Salama und Michael Scheuering: „Multitexturbasierte Volumenvisualisierung in der Medizin“, In *Handels, Horsch, Lehmann, Meinzer (Hrsg.): Bildverarbeitung für die Medizin 2001*, S. 137-141; Berlin, Springer, 2001
- [SBKP01] Will Schroeder und Chandrajit Bajaj und Gordon Kindlmann und Hanspeter Pfister: „The Transfer Function Bake-Off“, In *IEEE Visualization*, S. 523-526, 2001
- [Sgi04] <http://oss.sgi.com/projects/inventor>, August 2004
- [TLL03] Fan-Yin Tzeng und Eric B. Lum und Kwan-Liu Ma: „A Novel Interface for Higher-Dimensional Classification of Volume Data“, In *IEEE Visualization*, 2003
- [TLM01] Shivaraj Tenginakai und Jinho Lee und Raghu Machiraju: „Salient Iso-Surface Detection With Model-Independent Statistical Signatures“ In *Proceedings IEEE Visualization*, S. 255–262, 2001
- [TSH98] U. Tiede und T. Schiemann und K.-H. Höhne: „High-Quality Rendering of Attributed Volume Data.“, In *IEEE Visualization, Late Breaking Hot Topics*, 1998

-
- [Wat02] Alan Watt: *3D-Computergraphik*, Addison-Wesley Verlag, 3. Auflage, 2002
- [Web02] Regina Helena Weber: *Einfluss der Scanparameter auf das Abbildungsergebnis bei der CT- Angiographie*, Dissertation, Ludwig-Maximilians-Universität München, 2002
- [Wer94] Josie Wernecke: *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor*, Release 2. Addison-Wesley, 1. Aufl., März 1994a
- [XJ02] Yonghong Xie und Qiang Ji: „A new efficient ellipse detection method“, In *IEEE*, 2002
- [ZPBG01] Matthias Zwicker und Hanspeter Pfister und Jeroen van Baar und Markus Gross: „EWA Volume Splatting“, In *Proceedings of IEEE Visualization*, 2001

A Anhang

GradLutEditorMacro

Purpose

The **GradLutEditorMacro** helps to create gradient dependent two-dimensional transferfunctions on a histogram and to compute the corresponding 2D-LUT for the GVR.

Usage

Creation Transferfunctions (TFs):

This explains the usage of an example. To start it is needed to compute a 2D-Histogram with dimensions intensity and gradient strength. You have to specify the size of dimension of them. With the button „Compute Histogram“ the calculation is initiated. Now you can see the computed histogram in the left viewer. There you can define elliptic arcs. To create a TF, you have to know, that an elliptic arc represents an edge in the dataset. If you want to see the edge between air and blood then you have to draw one specific arc which represents this. You can look on the intensity; in CT the air is at -1024 HU and the blood lies at 0...200. The gradient is variable. If you find the searched arc, than you can draw an elliptic arc with the left mouse-button to get the representation of the TF. To draw the elliptic arc click at first on the extrempoint and than by holding the mousebutton pressed go to the footpoint. In the next step you have to define the transparency (Alpha) and the alpha distribution.

Than the colordistribution has to be defined. The color of the ellipse is also the color for the TF. You can change the color on the right side of the histogramviewer.

The last step you can define is the “Using of TF”. This is very important because it defines the strength of an egde. The ellipse is drawn until the footpoint of the arc. But the point of interest is the peak, because there lies the strongest edges. To generate a nice Volumerendering the ellipse has to be filled until a limit. The limit is defined by the “Using of TF”. So you can e.g. say the ellipse has to be filled until 50%.

If you now select the panel GVR, you can see your own TF in Volumerendering.

Inputs

- Loaded Image
- Gradient Strength Image
- Segmentation Mask

Outputs

- Gradient Strength Image
- 2D-LUT to use in GVR

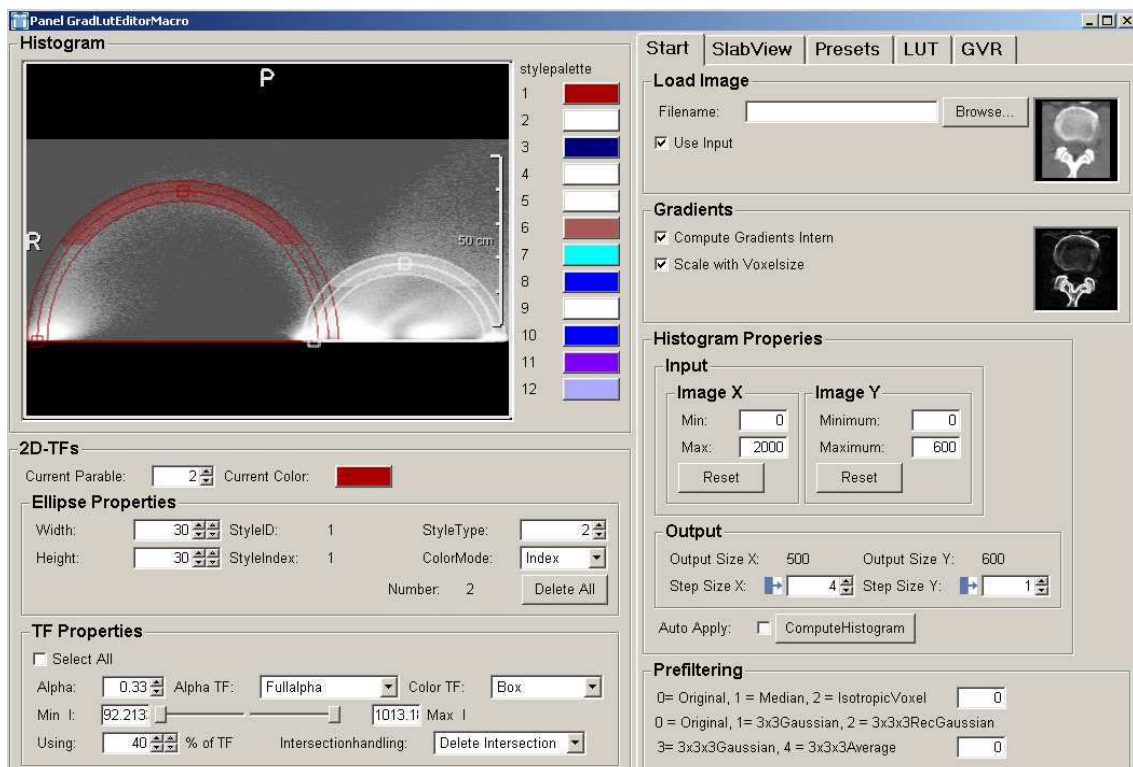
Mouseevents in the HistogramViewer

Draw a halfelliptic arc: press left mousebutton, holding the mousebutton pressed, if you leave the button, the halfellipse is ready (principle of rubberbanding)

Delete a halfellipse with the “Delete”-Key or with “Backspace”, (note: the current selected arc is deleted or the last drawn ellipse)

With the right mouse-button you can select a halfellipse. This option is useful, if you don't want to change the position of ellipse, but rather the parameters of TF

The Panel



Parameters

2D-TFs

Current Parable: the index of the current selected ellipse

Current Color: color of the current selected ellipse

Ellipse Properties:

Width/Height: defines the width/height of the ellipse to get two-dimensional representation

StyleId/StyleIndex: the indices of current selected ellipse to order colors of stylepalette

StyleType: you can define your own type for every ellipse

ColorMode: this field defines in which way the ellipses are coloured regarding the connected style palette. See also SoView2DMarkerEditor

Number: shows the number of created ellipses

Delete All: clears all ellipses

TF Properties:

Select All: if this field is on, the changes in this box are applied to all ellipses

Alpha: transparence (value between 0 and 1 (1=full opacity))

Alpha Distribution: controls the transparence distribution

- FullAlpha: in the whole arc is the same alpha-value
- EqualDistribution: the alpha-value goes down in y-direction from the extrempoint to the footpoint
- EqualToDist: the alpha-value goes also down in y-direction but only to a specific height, which is defined in UsingTF

Color Distribution: controls the color distribution

- box: the same color in the whole arc
- tent: on the side is black and in the middle it is the color
- ramp down: the color goes from the left of ellipse to black on the right
- ramp up: the color goes from black on the left to the color on the right

Min I/ Max I: controls the intensity area

Using: controls strength of edge

IntersectionHandling: handles the intersection of ellipses

Delete Intersection - deletes the colorValue in intersection area

First/Last: select the first/last drawn ellipse

Minimum Alpha/ Maximum Alpha: select the ellipse with lowest/highest transparency

Start

Before starting you have to do some settings:

LoadImage: you can browse an image or you can select the input image

Gradients: The gradients can be computed internal or you can put a gradient-strength dataset on the second input.

If you compute the gradients internal, you can scale the gradient strength with voxelsize. This is meaningful at the using of presets.

Histogram Properties: you can specify the histogram properties, see also the module JointHist, press the „compute Histogram“-Button to initialise computing.

Prefiltering: you can manipulate the incoming dataset, to get a less noisy and smoothed dataset

SlabView

Lut Mask: shows the voxels in slabview in the color of the lut

Use Alpha: shows the voxels in slabview in the color and the alphavalue of the lut, sometimes it is meaningful to deactivate this, because the alphavalue is so small that nothing can be recognised

Input Mask: shows the computed input contour mask in slabview

Explore Data: this box can help you to understand the dataset:

If you press the „Draw“-Button and you click with your mouse in the SlabViewer, the corresponding bin of the clicked voxel is signed in the histogram. The next selected voxels are also signed until you click the „Reset“-Button, then all signed bins are deleted.

You can also adjust the color of the signed bin and the size and you can select, which annotation should have the corresponding bin (e.g. worldpos, voxelpos)

ShowInputMask: If you have an input mask, then click the „compute histogram“-Button. The size of the mask-hist is like the size of intensity-gradient-histogram.

With **showInputMask** you can see the corresponding bins of the mask in the SlabViewer

MaskStrength: you can adjust the strength of the edges in the mask

ShowImportantHU: shows important CT-intensities in the histogram, e.g. – 1024(air), -700(bronchia), 0 (water) etc.

Presets:

LoadPresets: load a preset

SavePresets: save a preset

Save Thumbnails: save a thumbnail of Volumerendering for a new preset

LUT

Lut Range: adjust the range of lut 256 = 8bit

Lut Size X: on older graphic cards = 256

Lut Size Y: the GVR assists only LUTs of power 2, so the Lut Size Y has to be the same as Lut Size X

Supersampling: approximate better arcs

Show Checker: see the alphavalue of LUT

GVR

GVR on: enable the rendering on and off

Properties: adjust the alphablending, the samplingrate and the quality of rendering: this options influence the TF

Light: properties of scenelight: for gradient depended TFs it is important to use specular light.

SubVolume: set Focus to a subvolume

Create Snapshot: get a thumbnail for a new preset