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

Diplomarbeit

Entwurf distanzabhängiger 2D-Transferfunktionen für die medizinische Volumenvisualisierung

Andreas Tappenbeck

Angefertigt am Centrum für Medizinische Diagnosesysteme und Visualisierung der Universität Bremen

1. Gutachter: Prof. Dr. Bernhand Preim
2. Gutachter: Dr. Volker Dicken

Bearbeitungszeitraum: 01.04.04 - 30.09.04

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig und nur unter Verwendung der angegebenen Literatur angefertigt habe.

Andreas Tappenbeck

Bremen, 22.09.2004

Danksagung

Besonderen Dank möchte ich meinen beiden Betreuern Prof. Dr. Preim und Dr. Volker Dicken für die hervorragende fachliche Betreuung aussprechen. Die zahlreichen Diskussionen mit Dr. Volker Dicken waren sehr fruchtbar und gaben immer wieder neue Anregungen.

Darüber hinaus gilt mein Dank Wolf Spindler für seine Unterstützung bei der Implementierung, Stephan Dachwitz für die Bereitstellung des verwendeten Volumenrenderers sowie Diana Stölzel, die mir mit vielen nützlichen Tipps zur Seite stand. Nicht zuletzt möchte ich mich beim Centrum für medizinische Diagnosesysteme und Visualisierung (MeVis) für die Bereitstellung der verwendeten Datensätze und Software bedanken.

Weiterhin möchte ich meinen Eltern danken, die mich während meines Studiums uneingeschränkt unterstützt haben.

Inhaltsverzeichnis

1 Einleitung

Ein entscheidender Vorteil tomographischer Aufnahmeverfahren wie CT und MRT liegt in der Erzeugung überlagerungsfreier, digitaler Bilddaten. Das aufgenommene Bildmaterial wird dabei als Volumendatensatz repräsentiert, welcher zur Befundung üblicherweise in Form von Schichtbildern visualisiert wird. Über diese Darstellungsform hinaus konnte sich in den letzten Jahren Volumenrendering als zusätzliches Visualisierungsverfahren etablieren. Ein aus der Weiterentwicklung der Scannverfahren resultierendes Anwachsen der Datenvolumen macht zunehmend den Einsatz dreidimensionaler Darstellungstechniken erforderlich. Ermöglicht wurde dies nicht zuletzt durch die rasante Entwicklung von Grafikhardware, die die Renderingzeiten für Volumenvisualisierungen nach und nach in interaktionsfähige Bereiche rücken ließ und so auch die Hersteller medizinischer Workstations dazu veranlasste, Volumenrendering anzubieten. Für den klinischen Einsatz sind die Methoden des direkten Volumenrenderings (DVR) von besonderem Interesse. Sie kommen bereits beim Erzeugen von Übersichtsdarstellungen oder bei der Volumenvisualisierung kleinerer Schichtstapel zum Einsatz.

Medizinische Datensätze enthalten eine Fülle von Informationen, von denen oft nur ein geringer Teil von diagnostischer Relevanz ist. Da beim DVR Projektionen des gesamten Datensatzes erzeugt und dabei essentielle Informationen von unwesentlichen überlagert werden würden, ist es erforderlich eine Klassifikation der Bilddaten durchzuführen. Dies kann mit Hilfe von Transferfunktionen (TFs) geschehen, die Bildeigenschaften auf Grauwert bzw. Farbe und Opazität abbilden. TFs können in Abhängigkeit von verschiedenen Bildeigenschaften definiert werden. Häufig kommen Eigenschaften wie Intensität, Gradientenstärke oder Betrag der zweiten Ableitung in Gradientenrichtung zum Einsatz. Durch die Auswertung ihrer Intensität können Gewebe im begrenzten Maße differenziert werden. Problematisch ist hierbei, dass die Zuordnung von Intensitäten und Gewebe nicht eindeutig ist. Mit Hilfe der Gradientenstärke und dem Betrag der zweiten Ableitung in Gradientenrichtung ist es möglich Kanten im Datensatz, also Übergänge zwischen verschiedenen Geweben, darzustellen.

Ein neuer Ansatz, der in dieser Arbeit vorgestellt wird, beruht auf einer distanzabhängigen Definition von TFs. Die Abbildungseigenschaften von TFs werden hierzu in Abhängigkeit von der Distanz zu Referenzflächen im Datensatz, wie beispielsweise Organoberflächen, definiert. Ziel ist es, eine distanzabhängige Steuerung von Farbe und Transparenz zu erreichen, was beispielsweise eine anatomische Reformatierung (Schichtdarstellungen von Organen) oder ein distanzabhängiges Ausblenden bzw. Färben von Gewebe ermöglicht. Im Mittelpunkt stehen dabei 2D-TFs, die sowohl von Intensitätswerten als auch von Distanzen zu Referenzflächen abhängig sind.

Die vorliegende Arbeit ist wie folgt gegliedert:

Kapitel 2 gibt einen Überblick über die Grundlagen medizinischer Volumenvisualisierung. Dabei wird auf die Akquirierung von Volumendaten mit Hilfe tomographischer Aufnahmeverfahren eingegangen. Danach wird ein Überblick über DVR-Verfahren gegeben. Anschließend wird die Bedeutung von TFs herausgearbeitet sowie Möglichkeiten für deren Definition diskutiert.

Bisher ist sehr wenig über die Definition zweidimensionaler TFs bekannt. **Kapitel 3** beschäftigt sich damit, geeignete Methoden hierfür bereitzustellen. Diese werden aus bereits bekannten Ansätzen abgeleitet. Sie sind für die Definition distanzabhängiger TFs erforderlich.

In **Kapitel 4** werden distanzabhängige Transferfunktionen (DTFs) vorgestellt. Im Mittelpunkt steht dabei deren Definition und Anwendung bei der Visualisierung von CT- und MRT-Volumendaten. Darüber hinaus wird die Möglichkeit der Verwendung von Presets diskutiert.

In **Kapitel 5** wird die Implementierung eines Editors für die Definition von DTFs vorgestellt.

In **Kapitel 6** werden Ergebnisse diskutiert, die mit Hilfe von DTFs erzielt wurden. Darüber hinaus werden mögliche Anwendungen besprochen.

Kapitel 7 gibt eine Zusammenfassung sowie Anregungen zu weiterführenden Arbeiten.

2 Grundlagen

Dieses Kapitel führt in die Grundlagen des DVRs tomographischer Schichtdaten ein. Zu Beginn werden die wichtigsten tomographischen Aufnahmeverfahren vorgestellt. Diese Verfahren erzeugen Volumendatensätze deren Eigenschaften als nächstes diskutiert werden. Danach wird auf Visualisierungstechniken für medizinischen Volumendaten eingegangen. Ein weiterer Abschnitt gibt einen Überblick über verschiedene DVR-Verfahren. Anschließend werden die Rolle von TFs und Verfahren für deren Definition erläutert.

2.1 Tomographische Bildgebung

Bildgebende Aufnahmeverfahren sind die Basis der radiologischen Diagnostik. Eine besondere Rolle kommt dabei den tomographischen Aufnahmeverfahren zu, da sie überlagerungsfreie Darstellungen des menschlichen Körpers erzeugen und damit Bildmaterial mit hoher Aussagekraft bereitstellen. Im Folgenden werden die wichtigsten Verfahren vorgestellt.

2.1.1 Computertomographie (CT)

Bei der CT handelt es sich um ein spezielles Röntgenverfahren, welches Anfang der 70'er Jahre von dem englischen Ingenieur Godfrey N. Hounsfield (1919-2004) entwickelt wurde. Basis hierfür war das von Wilhelm Conrad Röntgen (1845-1923) entwickelte konventionelle Röntgenverfahren. Mit Hilfe der CT konnten erstmals Weichteilgewebe, welche sich in ihrer Dichte nur geringfügig unterscheiden, wie z.B. die graue und weiße Hirnsubstanz, differenziert werden. Darüber hinaus waren erstmals echte 3D-Rekonstuktionen der menschlichen Anatomie möglich. Im Laufe weniger Jahre entwickelte sich die CT zu einem der wichtigsten medizinischen Aufnahmeverfahren.

Funktionsprinzip CT

Senkrecht zur Körperachse des Patienten rotiert eine Röntgenröhre. Diese erzeugt einen fächerförmigen Röntgenstrahl, der den Körper in der gewünschten Ebene durchdringt. Dabei wird er in Abhängigkeit von der Dichte sowie der Dicke des durchstrahlten Materials abgeschwächt. Gegenüber der Röntgenröhre befindet sich eine Detektoreinheit, die die auftreffende Strahlung in elektrische Signale umwandelt. Diese werden an einen Rechner weitergeleitet, der sie auswertet. Für die Rekonstruktion einer Schicht ist es erforderlich, diesen Prozess mehrmals aus verschieden Winkeln zu wiederholen. Aus den Intensitätsprofilen wird anschließend über eine inverse Radon-Transformation ein Schichtbild rekonstruiert. Hierbei werden den gemessenen Dichten Intensitätswerte, so genannte Hounsfield-Units (HU) zugeordnet. Für verschiedene Gewebetypen ergeben sich unterschiedliche Intensitäten (Lunge: -600HU bis -700HU, Fett: -100HU bis -200HU, Leber: 50HU bis 60HU, Nieren 50HU bis 70HU, Knochen: 100HU bis 3000HU) (Luft: -1000HU, Wasser: 0HU) Auf diese Weise lassen sich mehrere Schichtbilder hintereinander erzeugen und zu einem Volumendatensatz zusammensetzen. In [6] wird ein detaillierter Einblick in die technischen Einzelheiten dieses Verfahrens gegeben. In [31] können die Einflüsse verschiedener Scanparameter recherchiert werden.

Spiral - CT

Neuere CT-Geräte arbeiten im Spiralverfahren. Hierbei wird der Patient mit konstanter Geschwindigkeit entlang seiner Längsachse durch die Strahlenebene bewegt, während die Strahlenquelle sowie die Detektoreinheit ebenfalls mit konstanter Geschwindigkeit rotieren. Dabei können mehrere Axialebenen gleichzeitig abgetastet werden, was die Aufnahmezeit erheblich verkürzt.

Typische Anwendungsgebiete für den Einsatz von CT sind beispielsweise:

Kopf:	bei Blutungen, Schädelbasisfrakturen, Gehirntumoren, Gehirnödemen
Knochen:	bei komplizierten Frakturen, Osteoporose, Bandscheiben-Vorfällen
Lunge:	Wegen des guten Kontrastes von Lungengewebe in CT-Aufnahmen wird für Untersuchungen der Lunge sehr häufig CT eingesetzt. Aus Kosten- und Verfügbarkeitsgründen kommt alternativ die konventionelle Röntgentechnik zum Einsatz.
Magen, Darm:	im Bereich der Tumordiagnostik

Vor- und Nachteile

Ein großer Vorteil der CT liegt darin, dass hohe Auflösungen innerhalb der Schichten erreicht werden (512×512). Dabei sind sehr kleine Schichtabstände (<0.6 mm) möglich. Ein weiterer Vorteil liegt in der hohen Aufnahmegeschwindigkeit. Es werden nur wenige Minuten für eine CT-Aufnahme benötigt.

Ein Nachteil der CT liegt in der Strahlenbelastung. Das damit verbundene Risiko muss bei der Indikationsstellung berücksichtigt werden. Die hohe Aussagekraft der mittels CT akquirierten Daten rechtfertigt jedoch oft die Durchführung. Ein weiteres Problem besteht in einem geringen Weichteilkontrast, der durch den Einsatz von Kontrastmittel und Fensterung (Kontrastspreizung bei der Darstellung) nur teilweise behoben werden kann.

CT-Datensätze und TFs

CT-Datensätze sind standardisiert, d.h. einem Gewebetyp sind in unterschiedlichen Datensätzen dieselben Intensitätswerte zugeordnet (ohne Kontrastmittel). Der Definitionsbereich erstreckt sich von -1024HU bis 3071 HU. Diese Eigenschaft wirkt sich sehr positiv auf das Erzeugen von TFs aus, da über die Auswahl von Intensitätsintervallen verschiedene Gewebetypen klassifiziert werden können. Außerdem ist es möglich, allgemeingültige TFs (Presets) zu definieren, die auf verschiedene Datensätze anwendbar sind.

2.1.2 Magnetresonanztomographie (MRT)

Ein weiteres tomographisches Aufnahmeverfahren ist die MRT. Sie steht seit Mitte der 80er Jahre als medizinisches Aufnahmeverfahren zur Verfügung.

Funktionsprinzip MRT

Die MRT basiert auf der Messung kernmagnetischer Resonanz. Durch das Anlegen eines starken statischen Magnetfeldes orientieren sich die Spins (Eigendrehimpulse) von Atomkernen mit ungerader Protonen- und Neutronenanzahl, im Wesentlichen die von Wasserstoffatomen, im zu untersuchenden Gewebe in Richtung der Feldlinien. Diese Orientierung wird mit Hilfe eines gepulsten elektromagnetischen Feldes gestört. Beim Abschalten dieses Feldes richten sich die Spins erneut entlang der Feldlinien des Magnetfeldes aus. Dabei erzeugen sie geringe elektromagnetische Impulse, die mit Hilfe eines Gradientenfeldes und spezieller Empfängerspulen gemessen werden. In der Praxis wird nicht eine Frequenz, sondern ein ganzes Spektrum eingestrahlt und anschließend das elektromagnetische Echo gemessen. Die ermittelten Daten werden danach durch einen Rechner ausgewertet und in Schichtbilder umgerechnet. Eine detailliertere technische Beschreibung wird in [6] gegeben.

Bei der Bildaufnahme wird zwischen der Messung der longitudinalen und der transversalen Relaxationszeit unterschieden. Die longitudinale Relaxationszeit beschreibt die Geschwindigkeit, mit der die Gleichgewichtsverteilung parallel zum Magnetfeld wieder erreicht wird. Wird sie beim Messvorgang ausgewertet, wird von einer T1-Gewichtung gesprochen. Derartiges Bildmaterial eignet sich beispielsweise sehr gut für die Differenzierung zwischen grauer und weißer Hirnsubstanz. Die transversale Relaxationszeit beschreibt die Geschwindigkeit, mit der die Gleichgewichtsverteilung senkrecht zum Magnetfeld wieder erreicht wird. Das so aufgenommene Bildmaterial wird als T2-gewichtet bezeichnet. Es zeichnet sich beispielsweise durch eine hohe Intensität von Zerebrospinalflüssigkeit, Ventrikeln und Tumoren aus. T1-gewichtete Bilder stellen flüssigkeitsreiche Strukturen dunkel (Abbildung 2-1), T2-gewichtete Bilder dagegen hell dar (Abbildung 2-2). Darüber hinaus gibt es eine Vielzahl weiterer Parameter, auf die an dieser Stelle nicht weiter eingegangen werden soll.

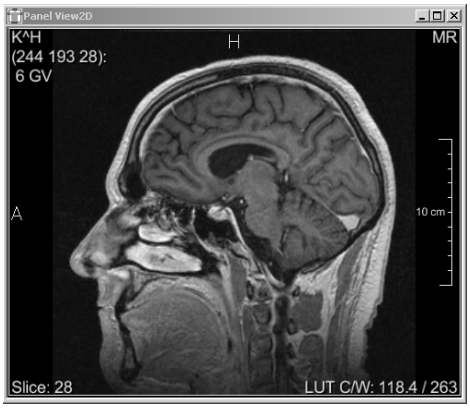


Abbildung 2-1: MRT, T1-gewichtet.

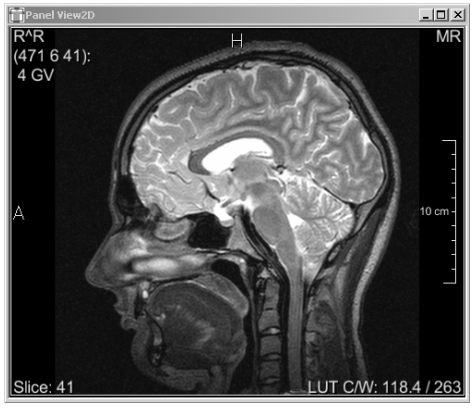


Abbildung 2-2: MRT, T2-gewichtet.

Vor- und Nachteile

Ein wesentlicher Vorteil von MRT ist der hohe Weichteilkontrast, der sehr differenzierte Darstellungen aller Körpergewebe ermöglicht. Je höher der Wasseranteil im untersuchten Gewebe (bei MRT werden Verteilung von Wasserstoffatomen gemessen), desto besser ist der Kontrast. Vor allem nicht-knöcherne Strukturen, wie z.B. Organe, Gelenkknorpel und Gehirn können gut abgebildet werden. Selbst geringfügige Veränderungen im Gewebe, wie kleine Entzündungsherde, sind auf diese Weise diagnostizierbar. Da keine Nebenwirkungen bekannt sind, kann das Verfahren, unter Vernachlässigung der damit verbundenen Kosten, beliebig oft eingesetzt werden.

Ein Nachteil der konventionellen MRT liegt in den langen Aufnahmenzeiten (10 bis 45 Minuten, je nach Größe des Aufnahmenbereichs). Weitere Nachteile ergeben sich aus der relativ geringeren Auflösung (oft 256×256), den großen Schichtabständen (meist > 2 mm) sowie dem Vorhandensein von Bildinhomogenitäten. Außerdem ist, wegen des für die Messung erforderlichen starken Magnetfeldes, keine Untersuchung bei Patienten mit Herzschrittmachern und Metallteilen im Körper, wie z.B. Spiralen, Metallprothesen, Gefäßclips etc. möglich. Weitere Nachteile liegen in den hohen Anschaffungskosten und dem großen Platzbedarf.

Anwendungsgebiete

MRT wird vorwiegend für Untersuchungen an Weichteilen wie Gehirn, Rückenmark, Knorpel, Nieren und Leber eingesetzt. Strukturen, die einen geringen Wassergehalt haben, wie z.B. Knochen, oder luftreiche Regionen wie die Lunge eignen sich dagegen weniger für die Darstellung mit MRT.

MRT-Datensätze und TFs

Die Intensitätswerte in MRT-Aufnahmen sind nicht normiert. Die zahlreichen Aufnahmeparameter wirken sich auf die Intensitätswerte der resultierenden Bilder aus. Hinzu kommt, dass sich über die Aufnahmen lokale und globale Inhomogenitäten erstrecken, so dass keine eindeutige Zuordnung von Intensitätswerten und Gewebetypen möglich ist. Da TFs intensitätsabhängig definiert werden, ergibt sich daraus das Problem, dass sie nur beschränkt auf verschiedene MRT-Datensätze angewendet werden können. Fast immer ist eine Adaption an den Datensatz erforderlich.

2.1.3 Positronenemissionstomographie (PET)

PET gehört ebenfalls zu den tomographische Aufnahmeverfahren. Da sich diese Arbeit jedoch ausschließlich mit dem DVR von CT- und MRT-Datensätzen beschäftigt, wird es im Folgenden nur kurz beschrieben.

Bei diesem Aufnahmeverfahren werden Isotope dargestellt, die durch Aussenden eines Positrons aus ihrem Kern radioaktiv zerfallen. Positronen sind sehr instabil, da sie sich schnell mit ihren Antiteilchen, den Elektronen, vereinigen (Annihilation). Dabei werden beide Massen in Energie umgewandelt, die vom PET-Scanner gemessen und in elektrische Signale umgewandelt wird. Da nicht der Ort der Positronen-Emission, sondern der Ort der Annihilation ermittelt wird (Abstand beträgt zwischen 2 bis 6 mm), entstehen Ungenauigkeiten bei der

Darstellung. Außerdem sinkt der Bildkontrast mit der Abnahme des Isotops im dargestellten Gewebe. Gebiete die nicht vom Isotop erreicht wurden, können nicht abgebildet werden. Im medizinischen Bereich kommen häufig die Isotope Sauerstoff 15 und Fluor 18 zum Einsatz. Das Isotop wird kurz vor dem Messvorgang gespritzt bzw. inhaliert.

Vor- / Nachteile

PET eignet sich sehr gut für die Beobachtung von Stoffwechselvorgängen (Isotope werden an Stoffwechselprodukte angekoppelt). Auch physiologische Vorgänge lassen sich untersuchen.

Nachteile dieses Verfahrens liegen in der geringen Auflösung und Bildschärfe, in den langen Aufnahmezeiten sowie den hohen Anschaffungskosten für PET-Scanner. Ein weiterer Nachteil ist in der Strahlenbelastung zu sehen, die jedoch in ihrer Höhe der Strahlenbelastung anderer nuklearmedizinischer Verfahren entspricht. Außerdem ist die Dauer von Untersuchungen durch die Halbwertszeit der Radionuklide begrenzt.

2.2 Medizinische Volumendatensätze

Tomographische Aufnahmeverfahren erzeugen Schnittbilder des zu untersuchenden Gewebes und fügen diese zu Volumendatensätzen zusammen. Volumendatensätze sind dreidimensionale Felder quaderförmiger Elemente (Voxel). Diese sind in orthogonalen Gittern angeordnet (Abbildung 2-3). Der skalare Wert eines Voxels repräsentiert die diskretisierte Intensität (Abschwächung von Röntgenstrahlen, Wassergehalt, Positronen-Emission etc.) an der korrespondierenden Gewebeposition. Volumendatensätze sind damit als diskrete Rekonstruktion eines kontinuierlichen Signals zu betrachten.

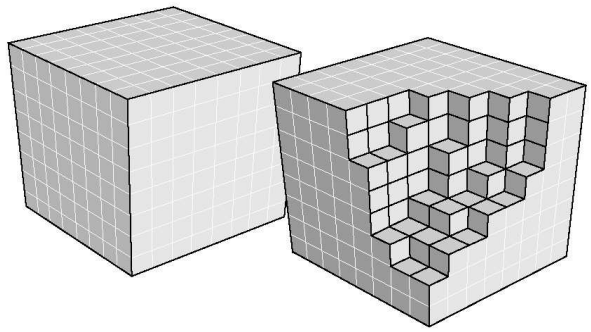


Abbildung 2-3: Volumendatensatz, Quelle: [1]

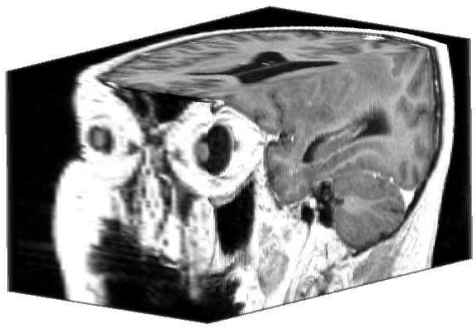


Abbildung 2-4: Ausschnitt eines Volumendatensatz (Kopf-MRT)

Die Ortsauflösung der Messpunkte kann unterschiedlich gewählt werden. Typische Auflösungen liegen beispielsweise im Bereich der CT bei 0.7 × 0.7 × 2 mm und bei der MRT bei 1.2 × 1.2 × 4 mm. Innerhalb einer Schicht der Datensätze ist die Ortsauflösung meist deutlich höher als zwischen den Schichten (anisotrope Daten). Allgemein kann gesagt werden, dass hohe Ortsauflösungen angestrebt werden, um möglichst detaillierte Aufnahmen zu gewährleisten. Abbildung 2-5 und Abbildung 2-6 verdeutlicht diesen Zusammenhang.

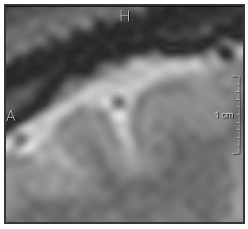


Abbildung 2-5: Kopf-MRT, Voxelgröße: 0.5×0.5×2 mm, Schichtauflösung: 512²

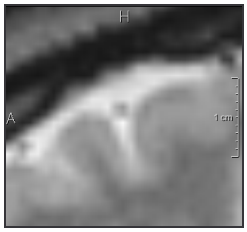


Abbildung 2-6: Kopf-MRT, Voxelgröße: 1×1×2 mm, Schichtauflösung: 256²

Zu beachten ist, dass medizinische Aufnahmesysteme wie CT, MRT, PET physikalisch bedingte Bildstörungen erzeugen (Verzerrungen, Rauschen, Inhomogenitäten, andere Artefakte), die die Information des aufgenommenen Objektes überlagern und sich nicht bzw. nur schlecht von der Nutzinformation trennen lassen. Eine Interpretation der Daten muss immer vor diesem Hintergrund erfolgen.

Medizinische Volumendaten werden üblicherweise im DICOM-Format abgelegt. Dieses Format ermöglicht einen standardisierten Austausch von Bildinformationen. Über die reinen Bilddaten hinaus können zusätzliche Informationen, wie die Einstellung von Aufnahmeparametern, abgelegt werden.

2.3 Visualisierung medizinischer Volumendaten

Schichtbilder

Im klinischen Alltag werden Volumendaten üblicherweise in Form von Schichtbildern visualisiert. Der große Vorteil dieser Visualisierungstechnik liegt in einer überlagerungsfreien Darstellung der Daten, die die Auswertungsmöglichkeit aller Bildinformationen sicherstellt. Dazu werden die Intensitätswerte der Voxel einer Schicht unter Anwendung einer TF auf die Pixel eines Grauwertbildes abgebildet (siehe Abbildung 2-7). Dieser Vorgang wird als Fensterung bezeichnet. Dabei wird ein Intensitätsintervall (Fenster) auf den Daten definiert, dem linear die für eine Darstellung zur Verfügung stehenden Grauwerte zugeordnet werden. Da das menschliche Auge nur eine begrenze Anzahl von Grauwerten unterscheiden kann, genügt für die Darstellung eine Auflösung in 256 Grauwertestufen (8Bit). Mit Hilfe der Fensterung können gezielt bestimmte Regionen des Volumens mit hohem Kontrast dargestellt werden (Kontrastspreizung).

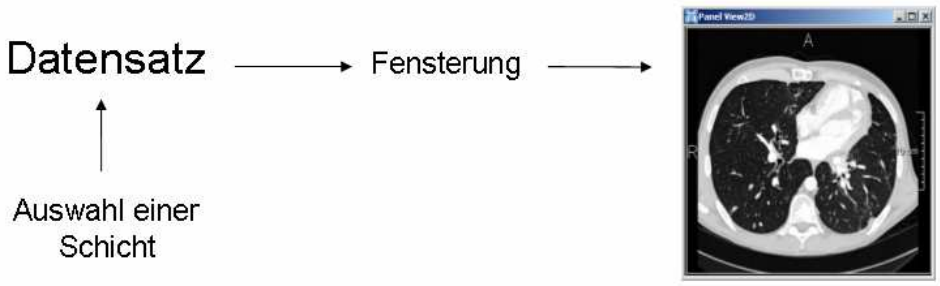


Abbildung 2-7: Visualisierung von Volumendatensätzen mittels Schichtbildern

Ein Nachteil dieser Visualisierungstechnik ergibt sich aus dem Umstand, dass lediglich eine Schicht des Datensatzes dargestellt werden kann. Um Strukturen zu erfassen, die sich über

mehrere Schichten erstrecken, sind alle betreffenden Schichten nacheinander zu betrachten. Eine 3D-Rekonstruktion muss mental durch den Betrachter erfolgen.

Direktes Volumenrendering (DVR)

Als Unterstützung hierfür wird im zunehmenden Maße DVR eingesetzt. DVR-Verfahren vermitteln einen guten dreidimensionalen Überblick über die Daten, sind jedoch nicht überlagerungsfrei. Während der Visualisierung sind verschiedene Interaktionen mit dem dargestellten Volumen, wie Zoom, Clipping und Rotation möglich. Auch hier erfolgt die Darstellung mit Hilfe von TFs. Einfache eindimensionale TFs lassen sich über das Festlegen des darzustellenden Intensitätsintervalls (Einstellen einer Grauwertrampe) definieren. Dieser Vorgang ist mit der Fensterung bei Schichtdatendarstellungen vergleichbar. Darüber hinaus können auch komplexere TFs zur Anwendung kommen. Abbildung 2-8 gibt einen schematischen Überblick.

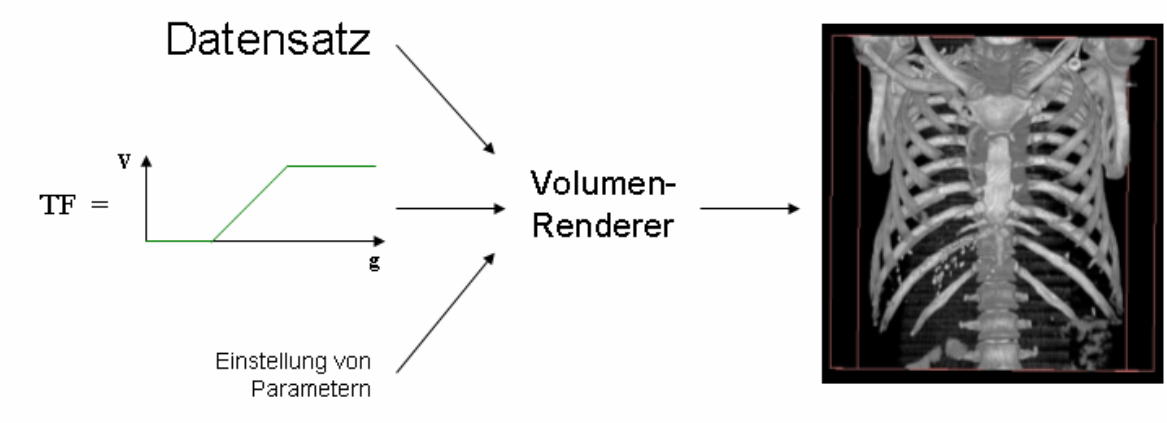


Abbildung 2-8: Überblick DVR

Im klinischen Alltag werden TFs selten über Parameter definiert. Vielmehr hat sich hier das Konzept von Presets durchgesetzt, bei dem vordefinierte TFs beispielsweise, mittels Icons ausgewählt werden. Eine Herausforderung stellt hierbei die Adaption von Presets an die Datensätze dar. Nichtstandardisierte MRT-Aufnahmen oder die Gabe von Kontrastmittel im CT-Bereich erschweren eine Zuordnung zwischen gemessenen Intensitätswerten und Gewebetypen. Im nachfolgenden Abschnitt werden verschiedene DVR-Verfahren vorgestellt.

2.4 Direktes Volumenrendering

DVR ist eine weit verbreitete Technik für die Darstellung diskreter dreidimensionaler Skalarfelder. Skalarfelder werden beispielsweise in der Strömungsvisualisierung, in der numerischen Simulation und der medizinischen Bildgebung erzeugt. DVR-Verfahren interpretieren diese Felder als Volumendatensätze und projizieren sie auf eine 2D-Bildebene. Die Voxel der Volumendatensätze sind hierbei semitransparent. Durch eine Zuweisung von Transparenzen wird ihr Einfluss auf die Projektion festgelegt (Klassifikation). Auf diese Weise kann eine Auswahl der darzustellenden Informationen getroffen werden. Eine Segmentierung oder andere Vorverarbeitungsschritte sind nicht erforderlich. Da diese Technik ohne geometrische Zwischenrepräsentation auskommt, wird sie auch als direktes Volumenrendering bezeichnet.

Es stehen eine ganze Reihe von DVR-Verfahren zur Verfügung. Sie lassen sich grob in zwei Gruppen, die software- bzw. texturbasierten Verfahren, einteilen. Softwarebasierte Verfahren können weiter in Bildraumverfahren (Image-Order) und Objektraumverfahren (Object-Order) unterschieden werden. Im Folgenden werden die am häufigsten verwendeten DVR-Verfahren vorgestellt. Eine im Hinblick auf Performanz, sehr interessante Evaluierung dieser Visualisierungstechniken wird in [11] vorgenommen.

2.4.1 Bildraumverfahren

Bildraumverfahren zeichnen sich durch eine sequentielle Berechnung der Pixel des Ergebnisbildes aus. Die Berechnung der einzelnen Pixel läuft dabei unabhängig voneinander ab und lässt sich deshalb gut parallelisieren. Ein sehr bekanntes, Bildraumverfahren, ist Ray Casting.

Ray Casting

Beim Ray Casting [9, 20] werden ausgehend vom Betrachter Strahlen durch jedes Pixel der Bildebene hindurch in den Objektraum (darzustellendes Volumen) geschickt. Anschließend findet eine Neuabtastung entlang der Strahlen statt. Dieser Vorgang wird als Resampling bezeichnet. Es können unterschiedliche Resamplingstrategien zum Einsatz kommen:

- *Rasterisierung*: Rasterisierung ist die einfachste Methode. Dabei werden alle Voxel bestimmt, die vom Strahl geschnitten werden. Die Intensitätswerte der Abtastpunkte ergeben sich hierbei aus den Intensitäten der geschnittenen Voxel. Diese Methode ist sehr schnell aber ungenau.
- *Äquivalente Abtastung*: Genauere Ergebnisse werden durch eine äquivalente Abtastung erzielt. Dabei werden die Abtastpunkte in konstanten Abständen auf dem Strahl festgelegt. Die Intensitätswerte an diesen Punkten lassen sich über die Nearest-Neighbour-Methode (Abtastpunkte erhalten den Wert des naheliegendsten Voxels) bestimmen oder trilinear aus den umgebenden Voxeln interpolieren. Eine wichtige Aufgabe liegt in dem Festlegen der Abstände zwischen den Abtastpunkten. Sind sie zu groß treten Aliasing-Artefakte auf, sind sie zu klein spiegelt sich dies in einem unnötig hohen Rechenaufwand wieder. Es können auch Verfahren eingesetzt werden, die die Schrittweite adaptiv, beispielsweise in Abhängigkeit von der lokalen Transparenz, variieren.
- *Exakte Schnittpunktberechnung*: Eine exakte Schnittpunktberechnung der Strahlen mit den Seitenflächen der Voxel ist aufwändig, liefert jedoch sehr genaue Ergebnisse. Diese Methode wird aus Performanzgründen praktisch nicht verwendet

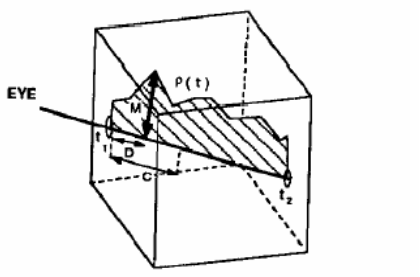


Abbildung 2-9: Ray Casting, Intensitätsverteilung entlang eines Abtaststahles, Quelle: [9]

Die ermittelten Intensitäten werden iterativ zu einem Pixelwert des resultierenden Bildes kombiniert. Dabei werden die Abtastpunkte entlang eines Strahls von hinten ($i=n$) nach vorn ($i=0$) durchlaufen. Die Farbe C_i und Transparenz α_i an einem Abtastpunkt i ergibt sich dabei aus der an diesem Punkt abgetasteten Farbe Cs_i und Transparenz αs_i sowie der Farbe C_{i-1} und der Transparenz α_{i-1} die für den vorherigen Abtastpunkt i-1 bestimmt wurden.

$$C_i = C_{i-1} * (1-\alpha_i) + \alpha s_i * Cs_i$$
$$\alpha_i = \alpha_{i-1} * (1-\alpha_i) + \alpha s_i \qquad \textbf{(2-1)}$$

Das so ermittelte C_0 und α_0 bestimmt die Farbe und Transparenz des mit dem Strahl assoziierten Pixels. Dieser Vorgang wird für jedes Pixel der Bildebene wiederholt. Der hierfür erforderliche Berechnungsaufwand ist angesichts der meist sehr großen Volumendatensätze enorm. Aus diesem Grund ist es sinnvoll, auf Beschleunigungsverfahren zurückzugreifen. Beispielsweise kann die Abtastrate in stark transparenten Bereichen des Datensatzes heruntergesetzt werden, ohne dass sich daraus signifikante Auswirkungen für die Intensität des zu berechnenden Pixels ergeben (adaptive ray sampling). Darüber hinaus ist keine Abtastung in Bereichen erforderlich, die sich hinter opaken Strukturen befinden (early ray termination).

2.4.2 Objektraumverfahren

Bei den Objektraumverfahren ist nicht die Bildebene Ausgangspunkt der Berechnung, sondern die darzustellenden Objekte (Voxel) selbst. Für jedes Voxel des Datensatzes wird dabei sein Beitrag am Ergebnisbild bestimmt. Dies geschieht indem alle Voxel beginnend von hinten nach vorn auf die Bildebenen projiziert werden (back to front - Verfahren).

Das bekannteste Objektraumverfahren ist das Splatting [17, 18, 21]. Es wurde Anfang der 90er Jahre von L.Westover entwickelt. Im Mittelpunkt dieses Verfahrens stehen spezielle Masken, so genannte footprints, die die Projektion von Voxeln auf die Bildebene repräsentieren. Für die Projektion eines Voxels wird zunächst eine Maske berechnet. Dies kann beispielsweise mit Hilfe von Gaußfiltern geschehen (Abbildung 2-10). Die Größe der Maske ist von der Größe des darzustellenden Voxels sowie der Größe des zu berechnenden Bildes abhängig. Die Maske wird anschließend mit dem Intensitätswert des zu projizierenden Voxels gewichtet und auf die korrespondierenden Pixel der Bildebene akkumuliert. Dieser Vorgang wird für alle Voxel des Datensatzes wiederholt. Splatting ist deshalb sehr gut auf Voxelebene parallelisierbar. Bei Parallelprojektion sind die Masken aller Voxel identisch. Bei perspektivischer Projektion muss für jedes Voxel eine neue Maske berechnet werden.

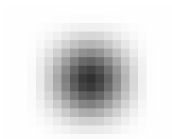


Abbildung 2-10: Splatting, Beispiel für eine mit Hilfe von Gaußfiltern berechnete Maske (footprint).

Objektraumverfahren bieten die Möglichkeit, Kohärenzen in Datensätzen auszunutzen. Voraussetzung dafür ist eine hierarchische Unterteilung der Volumendaten. Hierfür bieten sich Octtrees an. Nach einer hierarchische Unterteilung können vollständig transparente Bereiche

im Datensatz identifiziert und verworfen werden, ohne dass die enthaltenen Voxel ausgewertet werden müssen. Dieses Vorgehen führt, in Abhängigkeit von der Anzahl und Größe der transparenten Bereiche, zu einer mittleren bzw. hohen Beschleunigung. Ein weiterer Vorteil von Objektraumverfahren liegt darin, dass es mit ihrer Hilfe möglich ist, perspektivische Projektionen von Volumendaten zu erzeugen, ohne teure Interpolationen anwenden zu müssen. Außerdem können sie adaptiv arbeiten. Hierbei wird für jedes Voxel entschieden, ob es zum Ergebnisbild beitragen soll oder nicht. Dies kann beispielsweise über die Auswertung eines Transparenzschwellwertes geschehen.

2.4.3 Shear-Warp-Algorithmus

Der 1994 von Lacroute und Levoy vorgestellte Shear-Warp-Algorithmus [12] stellt einen Sonderfall dar, da er sowohl Eigenschaften des Bild- als auch des Objektraumes ausnutzt und so beide Ansätze miteinander kombiniert.

Bei den Bildraumverfahren ist besonders das Resampling entlang der Sichtstrahlen sehr rechenintensiv. Durch eine Vereinfachung dieses Vorgangs ließe sich eine erhebliche Beschleunigung erzielen. Hier setzt der Shear-Warp-Algorithmus an. Er läuft in mehreren Schritten ab. Zuerst wird das darzustellende Volumen in achsenparallele Schichten zerlegt und entlang dieser Schichten geschert, bis die Sichtstrahlen senkrecht auf den Schichten stehen. Im Fall einer perspektivischen Projektion ist hierfür zusätzlich eine Skalierung der Schichten erforderlich. Dabei ist der Skalierungsfaktor aller Voxel einer Schicht konstant. Bei einer Parallelprojektion entfällt dieser Schritt. Anschließend wird eine Projektion des Volumens erzeugt. Dies ist aufgrund der achsenparallelen Sichtstrahlen sehr einfach, da bilineare Interpolation genutzt werden kann. Zusätzlich wirkt sich positiv aus, dass die Gewichte bei der Interpolation für verschiedene Abtastpunkte identisch sind, also nur einmal berechnet werden müssen. In einem letzten Schritt wird die Projektion entzerrt. In Abbildung 2-11 sind diese Vorgänge schematisch dargestellt.

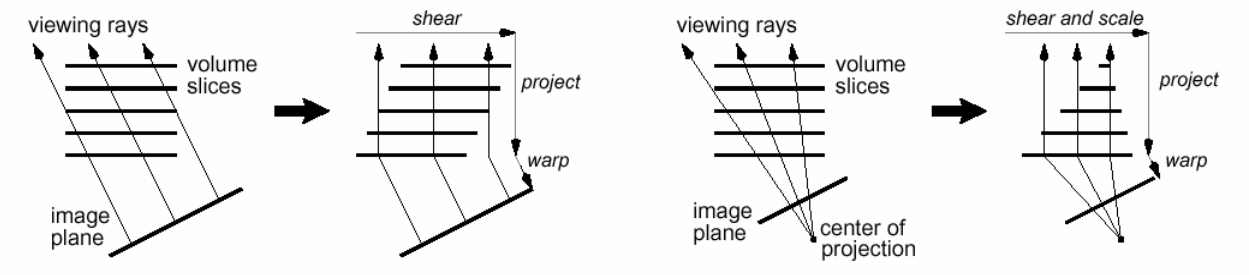


Abbildung 2-11: Shear-Warp-Algorithmus, Parallelprojektion (links), Perspektivische Projektion (rechts). Quelle: [12]

Weil keine Interpolation zwischen den Schichten des Datensatzes erfolgt, sind geringfügige Qualitätsverluste hinzunehmen. Dieses Verfahren ist derzeit das schnellste softwarebasierte DVR-Verfahren. Es kann deshalb als Standard aktueller Softwareimplementierungen betrachtet werden.

2.4.4 Texturbasierte Verfahren

Ein erheblicher Nachteil softwarebasierter DVR-Verfahren liegt in ihrer geringen Performanz, die kaum Interaktivität zulässt, ohne dass Abstriche an der Renderingqualität gemacht werden müssen. Das liegt im Wesentlichen an der hohen Anzahl an Interpolationen, die pro Frame

auszuführen sind. Eine Alternative stellen die texturbasierten DVR-Verfahren [1, 13, 14, 15, 16, 19] dar. Diese nutzen für die Darstellung von Volumendaten gezielt Fähigkeiten der Grafikhardware aus. Um DVR zu erzeugen, ist eine Neuabtastung (Resampling) der Daten erforderlich. Dieser Vorgang ist sehr rechenintensiv. Bei texturbasierten DVR-Verfahren werden diese Interpolationen über die Framebuffer-Arithmetik der Grafikkarte berechnet. Da sich auf diese Weise ein Großteil der Berechnungen auf Hardware verlagern lässt, sind texturbasierte Verfahren sehr schnell. In Abhängigkeit der zur Verfügung stehenden Grafikhardware können verschiedene Verfahren zum Einsatz kommen.

2D-Texturen

Die Verwendung von 2D-Texturen stellt nur geringe Anforderungen an die Grafikhardware. Sie werden daher von aktuellen PC-Grafikkarten unterstützt. Bei diesem Verfahren werden die Fähigkeiten des 2D-Texturmappings ausgenutzt.

Die Volumendaten werden hierfür in den Texturspeicher der Grafikkarte geladen. Falls das darzustellende Volumen größer ist als der Texturspeicher, wird es in Subvolumina (Bricks) zerlegt und nacheinander verarbeitet. Um eine korrekte Interpolation an den Brickübergängen zu gewährleisten, werden die Bricks so angelegt, dass sie sich ein Voxel breit überschneiden. Bisher unterstützten Grafikkarten keine Darstellung volumetrischer Primitive. Die Volumendaten müssen deshalb auf Hilfsgeometrien abgebildet werden (Multiplanar Reformatting). Hierfür werden äquidistante Schichtpolygone verwendet (Abbildung 2-12). Auf diese lassen sich achsenparallele Schichten des darzustellenden Volumens projizieren (2D-Texturmappig). Anschließend werden die mit semitransparenten Texturen versehenen Polygone beginnend von hinten nacheinander in den Framebuffer gerendert. Das Volumen muss hierfür drei mal im Texturspeicher gehalten werden. Wenn der Winkel zwischen Blickrichtung und Schichtennormale 45° überschreitet, wird auf eine günstigere Schichtenaufteilung (Textur) umgeschaltet. Dies ist mit Artefakten und Darstellungsverzögerungen verbunden.

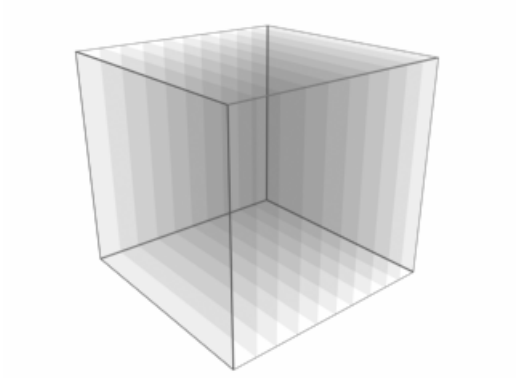


Abbildung 2-12: 2D-Texturen, Approximation eines Volumendatensatzes durch äquidistante Schichtpolygone auf die Schnittebenen des Volumens projiziert werden.

Die Vorteile von 2D-Texturen liegen in erster Linie in ihrer hohen Performanz sowie der guten Verfügbarkeit auf Standard-PCs. Nachteile entstehen durch den hohen Speicherbedarf, durch Bildartefakte und Umschalteffekte sowie einer inkonsistenten Abtastrate. Der größte Nachteil dieses Verfahrens ist jedoch, dass der Abstand der Schichtpolygone nicht frei gewählt werden kann. Er ist durch die Auflösung des Volumendatensatzes vorgegeben. Das führt bei stärkerem Zoomen dazu, dass die einzelnen Schichten deutlich erkennbar werden. Dieses lässt sich durch den Einsatz von 2D-Multitexturen verhindern.

2D-Multitexturen

Da 2D-Multitexturen [1, 19] eine Interpolation zwischen Texturen erlauben, kann der Abstand zwischen den Schichtpolygonen beliebig gewählt werden. Dies ist sinnvoll, um eine höhere Darstellungsqualität zu erreichen. Der Abstand kann beispielsweise an die Stärke des aktuellen Zooms angepasst werden. Grundlage hierfür ist die Fähigkeit von Grafikkarten, mehrere 2D-Texturen gleichzeitig auf Polygone zu mappen. Dabei werden durch Überblenden zwischen zwei Texturen trilinear interpolierte „Zwischentexturen“ berechnet. Auf diese Weise können die Abstände zwischen Schichtpolygonen beliebig klein gewählt werden, was zu exakteren Darstellungen führt. Dies ist ein entscheidender Vorteil gegenüber 2D-Texturen. Die Umschalteffekte werden dadurch allerdings nicht beseitigt. Es muss immer noch auf drei Texturkopien gearbeitet werden. Daraus resultiert außerdem ein hoher Speicherbedarf. Dennoch sind 2D-Multitexturen gut für DVR geeignet. Mit ihnen lassen sich unter Verwendung günstiger Hardware sehr gute Ergebnisse erzielen.

3D-Texturen

Die qualitativ hochwertigsten Ergebnisse werden mit 3D-Texturen erreicht. Die Anforderungen an die Grafikhardware sind hier sehr hoch, so dass dieses Verfahren anfangs nur Spezialsystemen (VolumePro etc.) vorbehalten blieb. Mittlerweile werden auch im PC-Bereich Grafikkarten angeboten, die 3D-Texturen unterstützen (NVidia ab GeForce3, ATI ab Radeon 8500). Der wesentliche Unterschied zu anderen texturbasierten Verfahren liegt darin, dass die Schichtpolygone parallel zur Bildebene erzeugt werden (Abbildung 2-13). Dies ist nur möglich, wenn die Grafikhardware direkte trilineare Interpolation unterstützt. Die Texturen auf den Schichtpolygonen können so trilinear aus dem Texturvolumen interpoliert werden. Dies geschieht seitens der Hardware, was hohe Frameraten ermöglicht. Der Abstand zwischen den Schichtpolygonen kann beliebig gewählt werden. So ist es sehr leicht möglich, Überabtastung zu erreichen, wodurch selbst bei starkem Zoom gute Ergebnisse gewährleistet werden können.

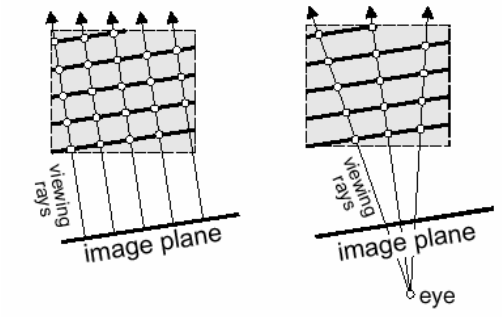


Abbildung 2-13: 3D-Texturen, Sichtstrahlen und Schichtpolygone bei Parallelprojektion (links) und perspektivische Projektion (rechts). Quelle: [1]

Bei diesem Verfahren ist keine Vervielfältigung des Texturvolumens im Grafikspeicher nötig, was den Speicherbedarf gegenüber 2D-Texturen und 2D-Multitexturen erheblich reduziert und Umschalteffekte vermeidet. Nachteilig ist, dass 3D-Texturen bisher kaum von Standard-Grafikhardware unterstützt werden. Eine detaillierte Beschreibung dieses Verfahrens wird in [38] gegeben. Für die Anfertigung der vorliegenden Arbeit wurde auf einer NVidia GeForce Quadro gearbeitet. Diese unterstützt 3D-Texturen sowie 8Bit-LUTs.

2.5 Transferfunktionen

Besondere Bedeutung kommt beim DVR der Klassifikation von Bilddaten zu. Hierbei werden die skalaren Intensitätswerte der Volumendaten auf visualisierbare Parameter abgebildet. Dies kann mit Hilfe von TFs geschehen. Dabei wird auf der Basis von Bildeigenschaften für jedes Voxel eines Volumendatensatzes Grauwert bzw. Farbe und Opazität für die Visualisierung festgelegt (Abbildung 2-14). Mit Hilfe der Opazität lässt sich die Sichtbarkeit von Voxeln steuern. Interessante Bereiche im Datensatz können so opak, weniger interessante semitransparent bzw. vollständig transparent dargestellt werden. Durch den Einsatz von Farbe können weitere Differenzierungen für die Darstellung vorgenommen werden, beispielsweise lassen sich verschiedene Gewebetypen durch unterschiedliche Färbung voneinander abgrenzen. Für den Einsatz von Farbe wird üblicherweise auf den RGBA-Farbraum abgebildet. Ein RGBA-Farbwert wird durch 4 skalare Werte zwischen 0 und 1 repräsentiert. Diese beschreiben die Rot-, Grün-, Blau- und Alphakomponente des Farbraumes. Eine detaillierte Beschreibung des RGBA-Farbraums kann in [5] nachgeschlagen werden.

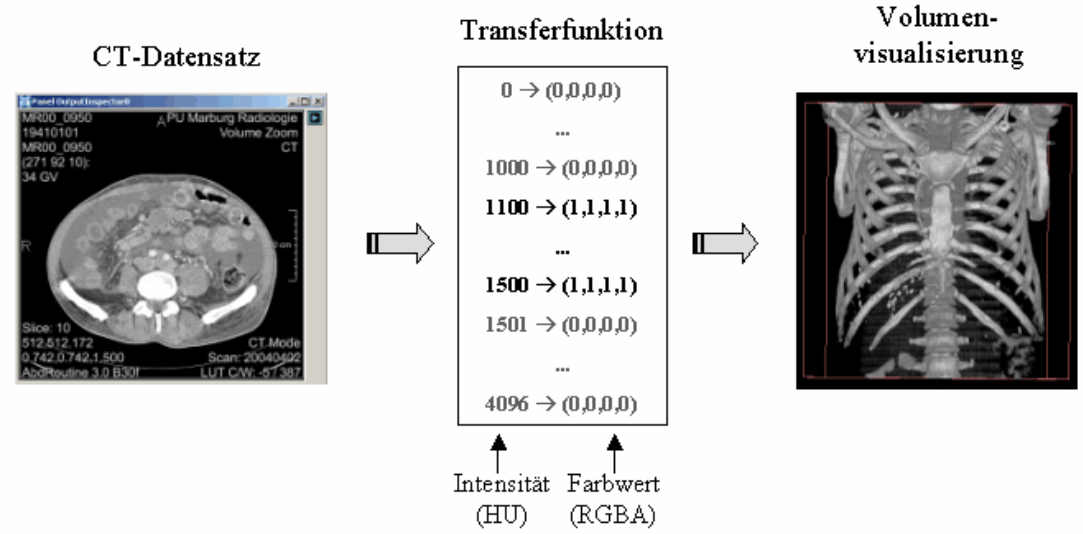


Abbildung 2-14: Klassifikation durch 1D-TFs. Hierbei findet eine Abbildung der Intensitätswerte auf RGBA-Farbwerte statt.

2.5.1 Multidimensionale Transferfunktionen

Die Anzahl der Bildeigenschaften, auf denen eine Klassifikation beruht, bestimmt die Dimension der verwendeten TF. Einfache eindimensionale TFs werden lediglich in Abhängigkeit zur Bildintensität definiert. Die Visualisierungsmöglichkeiten sind hier recht eingeschränkt. Höhere Flexibilität kann durch die Verwendung zusätzlich abgeleiteter Bildeigenschaften wie der Stärke des Gradienten oder der zweiten Ableitung in Gradientenrichtung erreicht werden. Mit einer gradientenabhängigen Opazitätsfunktion ist es beispielsweise möglich, Strukturen an Grenzflächen hervorzuheben. Auch die zweite Ableitung kann hierfür verwendet werden, da ihre Nullstellen auf Kanten im Datensatz hinweisen. Ein weiterer interessanter Ansatz ist die Akquirierung mehrerer Datensätze mit unterschiedlichen Aufnahmeverfahren bzw. Aufnahmeparametern. Die verschiedenen Datensätze können ebenfalls als Dimensionen einer TF aufgefasst werden.

Auf diese Weise lassen sich zusätzliche Freiheitsgrade für den Definitionsprozess erschließen. Dabei ist zu beachten, dass mit den Freiheitsgraden auch der Aufwand bei der Definition steigt. Besonders wenn diese rein interaktiv abläuft, macht sich das deutlich bemerkbar.

2.5.2 Pre / Post - Klassifikation

Für das DVR ist eine Neuabtastung der Volumendaten erforderlich. Diese Neuabtastung wird durch Interpolation zwischen Voxelintensitäten erreicht. Eine Anwendung von TFs kann sowohl davor (Pre-Klassifikation) als auch danach (Post-Klassifikation) erfolgen. Im Falle einer Pre-Klassifikation wird die TF also vor der Interpolation angewendet. Meist bilden TFs auf Farbräume ab, d.h. dass anschließend auf Farbwerten interpoliert werden muss. Daraus entsteht nicht nur der Nachteil, dass mehrere Werte pro Voxel zu interpolieren sind, es kann auch zu Problemen führen, weil die menschliche Farbwahrnehmung nicht wie die Interpolation linear verläuft. Bessere Ergebnisse sind deshalb mit Post-Klassifikation zu erzielen. Hierbei werden TFs auf die bereits interpolierten Skalarwerte angewendet. Eine Interpolation von Farbwerten entfällt.

2.5.3 Lookup-Tabellen

Eine häufig verwendete Repräsentation von TFs sind Lookup-Tabellen (LUTs). Diese repräsentieren TFs diskret. Der Datenraum, aus dem heraus abgebildet werden soll, wird hierfür in gleichgroße Bereiche unterteilt (diskretisiert). Für jeden dieser Bereiche werden Abbildungseigenschaften wie Farbe und Transparenz festgelegt. Um beispielsweise eine eindimensionale LUT mit einer Größe von 256 in Abhängigkeit von den Intensitätswerten eines CT-Datensatzes zu definieren, wird zunächst die Größe der Abschnitte entlang der Intensitätsachse bestimmt. Bei für CT üblichen 4096 verschiedenen HU-Werten im Datensatz ergibt sich eine Abschnittsgröße von 16 (4096:256), was bedeutet, dass für die Intensitätsintervalle [0-15], [16-31], [32-48] usw. Farbwerte festzulegen sind. Soll nun bei der Visualisierung die Farbe eines Voxels bestimmt werden, kann dessen Intensitätswert über das zugehörige Intensitätsintervall auf einen Farbwert abgebildet werden. Zu beachten ist, dass LUTs hardwareabhängig sind. Von Grafikkarten können unterschiedliche LUT-Größen (256^2, 512^2, 1024^2, 2048^2, 4096^2) unterstützt werden. Die Größe einer LUT gibt an, in wie viele Intervalle der Datenraum zerlegt wird. Je kleiner die Intervalle, desto genauer kann die Abbildung, also die Zuweisung von Grauwert bzw. Farbe und Transparenz erfolgen. Die letzten Betrachtungen haben gezeigt, dass bei 4096 möglichen Intensitätsstufen (CT) und einer LUT-Größe von 256 lediglich eine Auflösung des Datenraums in 16'er Schritten möglich ist. Dies bedeutet, dass Intensitätswerte, die weniger als 16 HU auseinander liegen, nicht mehr differenziert werden können. Durch eine Verdopplung der LUT-Größe kann der kleinste auflösbare Abstand von Punkten im Datenraum halbiert werden (512^2 à 8 HU; 1024^2 à 4 HU usw.). Dabei ist zu prüfen, ob die gewünschte LUT-Größe von der verwendeten Grafikkarte unterstützt wird.

LUTs werden intern häufig als Felder verwaltet, wobei die Dimension der LUT die Dimension des verwendeten Feldes bestimmt. Sie können von Grafikkarten direkt zur Darstellung von Volumendaten verwendet werden und stellen daher eine sehr hardwarenahe Repräsentation von TFs dar. Mit ihrer Hilfe ist eine äußerst effiziente Anwendung von TFs möglich.

2.5.4 Entwurf von TFs

In der Vergangenheit wurden verschiedene Verfahren für die Definition von TFs vorgestellt. Einige davon arbeiten automatisch, andere erfordern Interaktion mit dem Anwender. Diese Verfahren lassen sich nach der Art wie TFs festgelegt werden in drei Gruppen einteilen. Eine Gruppe basiert auf ausschließlich interaktiver Definition. Hierfür werden dem Anwender eine Reihe von Parametern zur Verfügung gestellt, über deren Einstellung er die Gestalt der TFs beeinflussen kann. Dabei wird nach dem **Trial and Error** - Prinzip vorgegangen. Die Parameter werden solange angepasst, bis das gewünschte Ergebnis erreicht ist. Dieses Vorgehen erfordert viel Interaktion, liefert jedoch meist gute Ergebnisse, da das Wissen und die Erfahrung des Anwenders in den Definitionsprozess einfließen können. Eine weitere Gruppe bilden die **datenbasierten Verfahren**. Bei ihnen werden die Parameter für die Definition von TFs aus den Bildeigenschaften abgeleitet. Als nützlich haben sich in diesem Zusammenhang Histogrammanalysen sowie die Auswertung von erster und zweiter Ableitung erwiesen. Apriori-Wissen des Anwenders kann hierbei nur begrenzt eingebracht werden. Eine dritte Gruppe wird als **bildbasierte Verfahren** bezeichnet. Diese Verfahren erzeugen eine Vielzahl von Vorschaubildern für deren Generierung die Parameter der TFs innerhalb gewisser Intervalle verändert werden. Diese Parameterintervalle können vorgegeben oder vom Anwender gesetzt werden. Über die Auswahl einer der präsentierten Vorschaubilder werden die Parameter für die Visualisierung übernommen, die bei der Erzeugung des Vorschaubildes verwendet wurden. Dieser Vorgang kann iterativ ablaufen, bis ein Vorschaubild und damit die Parameter der TF den Erwartungen entsprechen. Die verschiedenen Verfahren lassen sich nicht immer eindeutig einer dieser drei Gruppen zuordnen. Im Folgenden werden eine Reihe von Verfahren für die Definition von TFs diskutiert.

Ein erster Ansatz

Mark Levoy war einer der ersten, der sich mit dem DVR von Volumendaten beschäftigte. Sein 1988 vorgestellter Ansatz [33] basiert auf der Darstellung von Kanten in Datensätzen, die er als Isowert-Konturoberflächen bezeichnet. Diese werden durch die Auswertung von Isowerten und Gradientenstärken ermittelt. Das Verfahren ist demnach den datenbasierten Verfahren zuzuordnen. Zunächst wird ein Isowert für die Intensität festgelegt. Alle Voxel, die diese Intensität aufweisen, werden vollständig opak dargestellt. Voxel, deren Intensität nur geringfügig vom Isowert abweicht, erhalten eine Opazität indirekt proportional der Gradientenstärke, d.h. je stärker der Intensitätsunterschied zum Isowert, umso geringer ist die zugewiesene Opazität. Auf diese Weise lässt sich die Stärke einer Kante approximieren. Außerdem können so Aliasingartefakte vermieden werden. Voxel deren Intensität außerhalb eins Intervalls um den Isowert herum (Kantenstärke) liegt, werden vollständig transparent dargestellt. Eine mit diesem Verfahren erzeugte Opazitätsfunktion ist in Abbildung 2-15 illustriert.

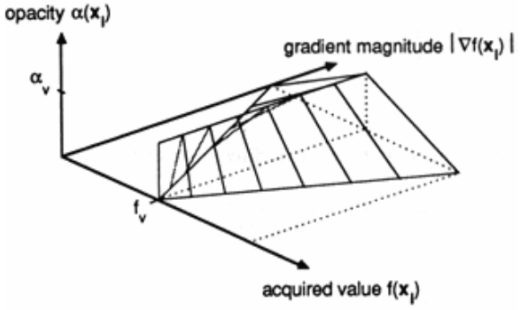


Abbildung 2-15: Opazitätsfunktion nach Levoy. Der Isowert ist mit f_V gekennzeichnet. Quelle: [33]

Ein erheblicher Nachteil dieses Verfahrens liegt darin, dass der Anwender wenig Einfluss auf das Visualisierungsergebnis hat. Er kann lediglich Isowert und Kantenstärke festlegen. Dadurch ist die Flexibilität des Definitionsprozesses sehr eingeschränkt.

Interaktive Definition von TFs

Castro et al. stellen in [3] einen rein interaktiven Ansatz für die Definition von 1D-TFs vor. TFs werden hier als gewichtete Summe von stückweise linearen Komponentenfunktionen betrachtet. Diese Komponentenfunktionen verfügen über wenige Parameter, wodurch eine einfache Interaktion gewährleistet werden kann. Es stehen verschiedene Grundformen der Komponentenfunktionen zur Verfügung (Zelt, Box, Trapez). Als Orientierung bei der Definition der Komponentenfunktionen kommen Histogramme des gesamten Datensatzes bzw. einzelner Schichten zum Einsatz. Auch Intensitätsprofile entlang von Sichtstrahlen werden verwendet. Die Definition von TFs erfolgt durch das Aufspannen von Komponentenfunktionen über einem Histogramm (Abbildung 2-16).

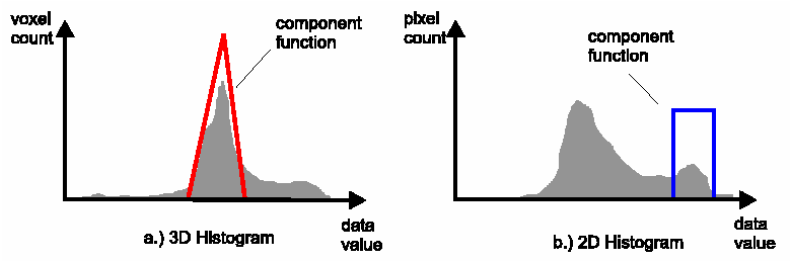


Abbildung 2-16: 1D-TFs nach Castro et al.

Dieses Verfahren zeichnet sich, im Gegensatz zu Levoys Ansatz, durch hohe Flexibilität aus. Es lassen sich beliebig viele Komponentenfunktionen einsetzen. Auf diese Weise können recht komplexe TFs erzeugt werden. Ein Nachteil liegt in der geringen Intuitivität. Die Intensitätsverteilungen der darzustellenden Gewebe sind meist nur schwer im Histogramm aufzufinden. Weitere Betrachtungen zu diesem Verfahren werden in Abschnitt 3.1.2 vorgenommen.

Ein zweiter ausschließlich interaktiver Ansatz wird von König und Gröller in [4] vorgestellt. TFs werden mit Hilfe von parametrisierten Grundfunktionen (Peaks) erzeugt. Es stehen verschiedene Grundfunktionen zur Verfügung. Die Anzahl ihrer Parameter ist gering (2 bis 3). Sie lassen sich deshalb leicht interaktiv festlegen. In Abschnitt 3.1.3 wird genauer auf diesen Ansatz eingegangen.

Interaktive Verfahren sind dadurch dass der Anwender die Möglichkeit erhält sein Wissen in den Definitionsprozess einzubringen besonders gut für die Exploration von Datensätzen geeignet. Aufgrund des zumeist recht hohen Interaktionsaufwandes sind sie für den klinischen Einsatz eher uninteressant.

Nutzung von Histogrammvolumina für DVR

Der von Levoy vorgestellte Ansatz, lediglich die Kanten in Datensätzen darzustellen, weil diese den wesentlichen Anteil der eigentlichen Information tragen, wird von Kindlmann und Durkin in [26, 34] aufgegriffen und verfeinert. Levoy wertet für die Detektion von Kanten zwei Dimensionen des Datenraumes, die Intensitätswerte und die Gradientenstärken, aus. Kindlmann und Durkin fügen dem eine weitere Dimension, die zweite Ableitung in Gradien-

tenrichtung, hinzu. Dies ist sinnvoll, weil aus der zweiten Ableitung zusätzliche Informationen über den Kantenverlauf gewonnen werden können, da ihre Nullstellen auf die Position von Kanten hinweisen. Voxel, an deren Position die zweite Ableitung nahe 0 und die Gradientenstärke hoch ist, sind demnach mit großer Wahrscheinlichkeit Kanten zuzuordnen. Alle drei Dimensionen des Datenraumes werden in einem Histogrammvolumen zusammengefasst. Hier werden die Häufigkeiten von Intensitäten, Gradientenstärken und Werten der zweiten Ableitung gegeneinander abgetragen (Abbildung 2-17). Ausgehend von einem Kantenmodell wird anschließend die Position aller Kanten aus dem Histogrammvolumen ermittelt. TFs sind abzuleiten, indem allen Voxeln proportional der Wahrscheinlichkeit, dass sie zu einer Kante gehören, Opazitäten zugewiesen werden. Untersucht wurden Histogrammvolumen von 80^3 bis 256^3 bei einer Bittiefe von 8. Das vorgestellte Verfahren arbeitet semiautomatisch, das bedeutet, dass Interaktion mit dem Anwender erforderlich ist. Hierfür sind aufgrund des hohen Abstraktionsgrades der Parameter ausgefeilte Interaktionskonzepte erforderlich, die jedoch nicht vorgestellt werden. Dies wurde von Kniss et al. in [32] nachgeholt (Abbildung 2-18).

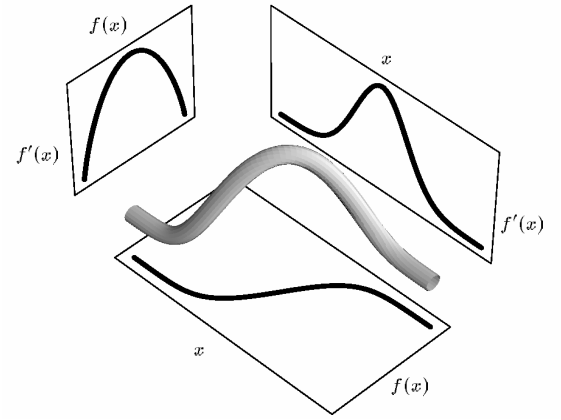


Abbildung 2-17: Ermitteln von Kanten durch die Auswertung eines Histogrammvolumens in dem Intensität x, Gradientenstärke f(x) und 2.Ableitung in Gradientenrichtung f'(x) gegeneinander abgetragen werden. Dabei resultieren aus Kanten im Datensatz dreidimensionale Kurven im Histogrammvolumen. Quelle: [34]

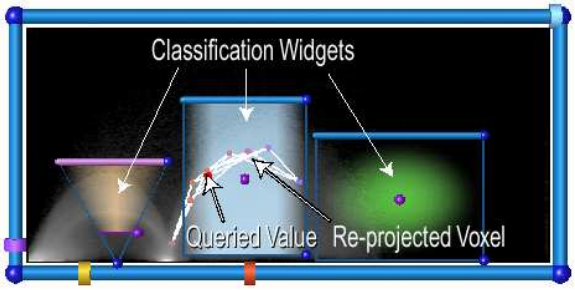


Abbildung 2-18: Darstellung eines 3D-Widgets für die Steuerung von Parametern des Histogrammvolumens. Mit Hilfe von Classification Widgets erfolgt eine Zuordnung von Opazitäts- und Farbverläufen. Quelle: [32]

Durch die zusätzliche Auswertung der zweiten Ableitung in Gradientenrichtung kann die Position von Kanten im Datensatz genauer ermittelt werden. Dadurch lassen sich bessere Ergebnisse erzielen, als mit ausschließlich gradientenabhängigen Ansätzen. Ein Nachteil dieses Verfahrens liegt darin, dass Strukturen, die einen sehr geringen Intensitätsunterschied zu ihren Nachbarstrukturen aufweisen, aufgrund des geringen Gradienten an der umschließenden Kante nicht oder nur sehr schlecht dargestellt werden können. Dieses Verfahren eignet sich jedoch gut für die Darstellung von Kanten mit mittlerem bzw. hohem Gradienten. Ein weiteres Problem tritt bei Strukturen auf, die von Strukturen unterschiedlicher Intensität eingeschlossen sind, da die umschließende Kante verschiedene Gradienten und Intensitäten aufweist und somit nicht einheitlich dargestellt werden kann.

Ein Verfahren zur Bestimmung von Isowerten

Ein weiteres Verfahren für die Visualisierung von Volumendatensätzen sind Isokonturen. Mit ihrer Hilfe lassen sich Untermengen von Datensätzen darstellen, die denselben Intensitätswert (Isowert) aufweisen. Auf diese Weise können Strukturen innerhalb der Daten visualisiert werden. Entscheidend für eine aussagekräftige Visualisierung ist die Wahl eines geeigneten Isowertes. Isokonturen sind vorwiegend für die Visualisierung von CT-Aufnahmen interessant. Bei MRT-Daten liegen meist stärkere Intensitätsschwankungen vor, so dass kein einheitlicher Isowert gefunden werden kann. Auch die Darstellung von Tumoren ist schwierig, da an ihrer Oberfläche nur selten einheitliche Intensitätswerte vorliegen.

Bajaj et al. stellen in [29] das Konturspektrum, eine Benutzeroberfläche für die Auswahl von Isowerten, vor. Dieser Ansatz basiert darauf, Eigenschaften von Isowerten als Funktionsgraphen entlang der Intensitätsachse eines zweidimensionalen Diagramms abzutragen. Als Eigenschaften werden Volumen, Größe der Isooberflächen sowie der mittlere Gradient untersucht. Der mittlere Gradient repräsentiert die Wahrscheinlichkeit, dass sich eine Isooberfläche zwischen zwei verschiedenen Strukturen, also auf einer Kante im Datensatz befindet und spielt deshalb für die Darstellung von Kanten eine wichtige Rolle. Die Eigenschaften werden für jeden Intensitätswert berechnet und in Form von Funktionskurven präsentiert. Auf Grundlage dieser Darstellung können geeignete Isowerte für die Visualisierung von Volumendaten ermittelt werden.

Dieses Verfahrens liefert einen guten Überblick über die Eigenschaften von Isowerten. Dabei lassen sich Daten beliebiger Dimension untersuchen. Außerdem ist es leicht um die Auswertung zusätzlicher Eigenschaften, wie beispielsweise die Krümmung von Isokonturen, erweiterbar. Die Anwendungsgebiete beschränken sich größtenteils auf die Visualisierung von CT-Daten.

Design Galleries

Mit den Design Galleries wird von Marks et al. in [37] ein bildbasierter Ansatz für die Suche nach geeigneten TFs vorgestellt. Dabei werden mittels stochastischer Suche bzw. unter Verwendung von genetischen Algorithmen potentiell sinnvolle Parametervektoren für Repräsentationen von TFs ermittelt. Auf deren Basis werden Vorschaubilder gerendert und dem Anwender präsentiert. Durch die Auswahl eines Vorschaubildes werden die Parameter, die seiner Generierung zu Grunde lagen, für die Visualisierung übernommen.

Die Interaktion ist ausschließlich auf die Auswahl von Vorschaubildern beschränkt. Daraus ergibt sich sowohl der Vorteil, dass sich die Interaktion sehr einfach gestaltet und deshalb auch von unerfahrenden Anwendern leicht auszuführen ist, als auch der Nachteil, dass die Parameter von TFs nur implizit festgelegt werden können. Informationen über den Datensatz, die bei der Suche nach geeigneten TFs hilfreich sein könnten, können nur sehr vage aus den Vorschaubildern abgeleitet werden. Damit ist dieses Verfahren vorwiegend für Anwender geeignet, die wenig Erfahrung mit der Definition von TFs besitzen.

Nutzen von Filteralgorithmen

Ein weiterer bildbasierter Ansatz wurde von Fang et al. in [28] vorgestellt. Die Definition von TFs basiert hier auf einer sequentiellen Anwendung verschiedener Bildverarbeitungsfilter

sowie auf der Anwendung von Lookup-Tabellen. Der Anwender kann über die Parametrierung der Filter Einfluss auf die Ergebnisse nehmen. Das Verfahren gliedert sich in zwei Stufen. In der ersten Stufe finden Abbildungen der Intensitätswerte auf andere Intensitätswerte statt (intensity mappings). Dabei können verschiedene Ziel, wie eine Filterung der Informationen, Rauschunterdrückung, Kantenextraktion usw., verfolgt werden. In einem zweiten Schritt werden die resultierenden Intensitätswerte unter Anwendung einer Lookup-Tabelle auf RGBA-Farbwerte abgebildet (coloring process). Als Lookup-Tabellen kommen lineare Rampen zum Einsatz. Die Mächtigkeit dieses Verfahrens wird maßgeblich durch die verwendeten Filteralgorithmen bestimmt.

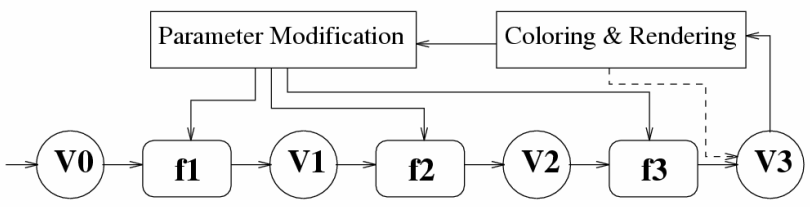


Abbildung 2-19: Datenbasierter TF-Entwurf nach Fang et al. Durch Filteroperatoren (f1, f2, f3) werden die Intensitätswerte eines Ausgangsvolumens (V0) sequentiell auf die Intensitätswerte eines Ergebnisvolumens (V3) abgebildet. Anschließend findet durch Anwendung einer Lookup-Tabelle eine Abbildung der Intensitätswerte auf RGBA-Farbwerte (coloring) sowie ein Rendering des Volumens statt. Der Anwender kann über das Einstellen von Parametern der Filteralgorithmen Einfluss auf das Visualisierungsergebnis nehmen. Quelle: [28]

Darüber hinaus wurden eine Vielzahl weiterer Ansätze veröffentlicht. He et al. [40] stellen ein Verfahren für die stochastische Suche nach TFs vor. Fan-Yin Tzeug et al. [41] beschreiben das Erzeugen vom TFs auf Basis neuronaler Netze. Weitere Verfahren können in [30, 42, 43, 44, 45, 46, 47, 48, 49, 50] recherchiert werden.

2.6 Zusammenfassung

In diesem Kapitel wurde ein Überblick über das DVR medizinischer Volumendaten gegeben. Zunächst wurden einige tomographische Aufnahmenverfahren erläutert. Da sich diese Arbeit mit der Visualisierung von CT- und MRT-Datensätzen beschäftigt, wurden diese Aufnahmeverfahren ausführlich diskutiert. Bei der anschließenden Erläuterung von Visualisierungstechniken für medizinische Volumendaten wurde auf die Verfahren der direkten Volumenvisualisierung und im speziellen auf texturbasierte Verfahren eingegangen. Dabei wurde festgestellt, dass bei dem aktuellen Stand der Grafikhardware 3D-Texuren am besten für die Darstellung von Volumendaten geeignet sind. Für die Erzeugung von Visualisierungen wurde in dieser Arbeit ausschließlich mit 3D-Texturen gearbeitet. Danach wurde auf die Rolle von TFs beim DVR eingegangen. Wesentliche Punkte waren hierbei deren Repräsentation durch Lookup-Tabellen sowie deren Entwurf. Für den Entwurf von DTFs werden Verfahren zur interaktiven Definition von 2D-TFs benötigt. Es stehen bereits einige interaktive Ansätze zur Verfügung. Diese sind jedoch ausschließlich für 1D-TFs ausgelegt und können deshalb nicht genutzt werden. Die bekannten 2D-Verfahren hingegen basieren auf (semi-)automatischer Definition und nutzten dabei zumeist Gradienteninformationen. Auch diese Verfahren sind ungeeignet. Es konnte kein geeigneter Ansatz für die interaktive Definition von 2D-TFs recherchiert werden. Es wird jedoch festgestellt, dass die Verfahren von Castro et al. sowie König und Gröller intuitive und mächtige Werkzeuge für die Definition von 1D-TFs darstellen. Für die Definition von 2D-TFs können diese Verfahren nützliche Anregungen geben. Sie werden deshalb in Kapitel 3 bei der Evaluierung von Methoden für die Repräsentation von 1D-TFs genauer unter-

sucht. Im nachfolgenden Kapitel werden Verfahren für die interaktive Definition von 2D-TFs entwickelt.

3 Interaktive Definition von 2D-TFs

Ein wesentlicher Aspekt des DVRs ist die Definition von TFs (siehe Abschnitt 2.5). Diese legen die Abbildung der zumeist skalaren Volumendaten auf darstellbare Parameter wie Grauwert bzw. Farbe und Opazität fest und bestimmen damit maßgeblich die Sichtbarkeit und das Erscheinungsbild der im Datensatz enthaltenen Strukturen. Bereits geringfügige Änderungen der TF können große Auswirkungen auf das Visualisierungsergebnis haben. Dabei sind die komplexen Zusammenhänge zwischen TF und der resultierenden Visualisierung nur schwer nachvollziehbar. Dieses Problem vergrößert sich mit der Anzahl der abhängigen Dimensionen. Während sich eindimensionale TFs recht intuitiv beispielsweise mit Hilfe von Grauwertrampen beschreiben lassen, fehlt es schon bei zweidimensionalen an geeigneten Modellen und Interaktionskonzepten. Aus diesem Grund werden mehrdimensionale TFs üblicherweise für jede Dimension getrennt definiert. Die resultierenden 1D-TFs lassen sich durch Multiplikation zu mehrdimensionalen TFs verknüpfen. Nachteilig ist jedoch, dass der Definitionsprozess nicht im gesamten Argumentraum, sondern nur separat innerhalb der einzelnen Dimension stattfinden kann. Damit ist lediglich eine indirekte Kontrolle der finalen TF möglich. Ein weiteres Problem ist in den stark einschränkten Definitionsmöglichkeiten zu sehen. Nur ein Bruchteil der Definitionsmöglichkeiten die im mehrdimensionalen Raum bestehen lassen sich durch Multiplikation von 1D-TFs ausdrücken (Abbildung 3-1).

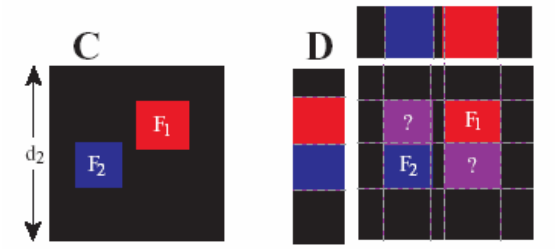


Abbildung 3-1: Einschränkungen bei der Definition von mehrdimensionalen TFs durch Multiplikation von 1D-TFs. Die beiden Eigenschaften F1 und F2 der 2D-TF C sind nicht durch Multiplikation zweier 1D-TFs separabel. Quelle: [30]

Aus diesen Gründen stellt die interaktive Suche nach geeigneten TFs, besonders im mehrdimensionalen Fall, eine große Herausforderung für den Anwender dar. Im Mittelpunkt dieses Kapitels steht die Entwicklung geeigneter Modelle für die interaktive Definition von 2D-TFs. Ziel ist es, durch das Festlegen weniger Parameter maximale Kontrolle über den Verlauf von TFs zu erhalten und so gezielt Einfluss auf das Visualisierungsergebnis nehmen zu können. Die Modellparameter sollten dabei über möglichst intuitive Interaktionskonzepte steuerbar sein.

Zu Beginn dieses Kapitels wird eine Evaluierung von Methoden für die Repräsentation von 1D-TFs vorgenommen. Im Anschluss werden basierend auf den Ergebnissen geeignete Modelle für 2D-TFs abgeleitet. Außerdem werden Interaktionskonzepte für die entwickelten Repräsentationen sowie ein Gittermodel für deren Implementierung vorgestellt.

3.1 Evaluierung von Methoden für die Repräsentation von 1D-TFs

Es hat sich als sehr förderlich für den Definitionsprozess von TFs erwiesen, wenn dem Anwender ein von den eigentlichen Daten (z.B. LUTs) abstrahiertes Modell für die Interaktion zur Verfügung gestellt wird. Nützlich sind in diesem Zusammenhang Meta-Repräsentationen, die über möglichst wenige, gut verständliche, Parameter verfügen und somit die Bereitstellung eines leicht zu bedienenden Interfaces für die Visualisierung ermöglichen. Für 1D-TFs sind eine ganze Reihe solcher Repräsentationsformen bekannt. Im Hinblick auf eine Erweiterung für 2D sollten sie verschiedenen Anforderungen genügen.

3.1.1 Anforderungen an 1D-Repräsentationen

Für eine explorative Analyse ist es wesentlich, dass dem Anwender während der Definition der TF Feedback wie beispielsweise ein Volumenrendering oder eine Ansicht der resultierenden LUT zur Verfügung gestellt wird. Auf diese Weise lässt sich die Bedienbarkeit erheblich verbessern. Um dabei Interaktivität zu gewährleisten, muss die Berechnung von LUTs sehr schnell erfolgen. Eine wesentliche Anforderung besteht deshalb in der Möglichkeit einer schnellen LUT-Generierung. Darüber hinaus sollte die Repräsentation über wenige möglichst intuitive Parameter verfügen. Die Anzahl der Parameter, die ein Anwender zu definieren hat, um TFs zu erzeugen, ist möglichst gering zu halten. Je weniger Parameter benötigt werden, desto einfacher ist der Definitionsprozess. Diese Anforderung ist für die Bereitstellung einer benutzerfreundlichen Schnittstelle essentiell. Da 2D-TFs repräsentiert werden sollen, steht außerdem eine einfache Erweiterungsmöglichkeit der Repräsentation für 2D im Vordergrund.

Zusammenfassend sind folgende Anforderungen an Repräsentationen für 1D-TFs zu stellen:

- hohe Flexibilität beim Definitionsprozess von TFs
- Definition mit Hilfe weniger intuitiver Parameter
- Möglichkeit der Erweiterung auf 2D
- Unterstützung einer effizienten Berechnung von LUTs

Im Folgenden werden einige 1D-Repräsentationen vorgestellt und hinsichtlich der aufgestellten Anforderungen untersucht.

3.1.2 Stückweise lineare Funktionen - gewichtet

Ein Ansatz für die Repräsentation von 1D-TFs wird in [3] vorgestellt. Ziel ist es 1D-TFs für das DVR von medizinischen Volumendaten in Abhängigkeit von der Intensität festzulegen.
TFs werden hier als gewichtete Summe von Komponentenfunktionen CF_i betrachtet. Eine Opazitätsfunktion $O(x)$ wird definiert als:

$$O(x) = \sum_{i=1}^{n} w_i(x) \cdot CF_i(x) \qquad \textbf{(3-1)}$$

$$i \in N$$

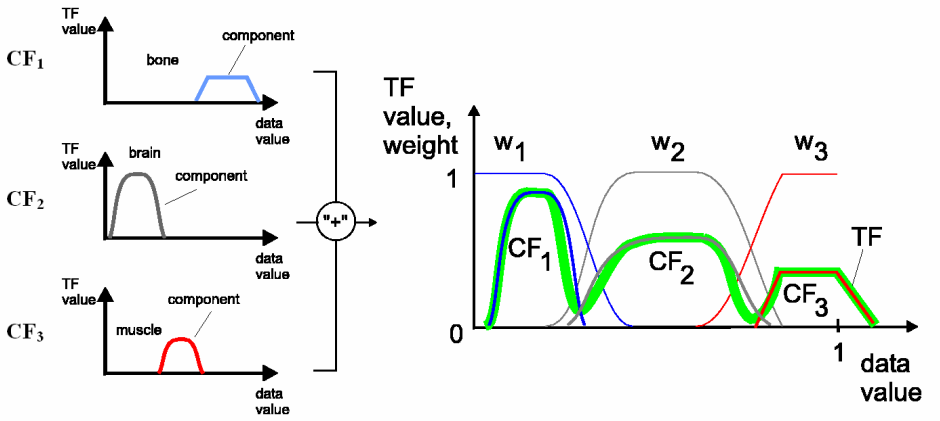


Abbildung 3-2: 1D-TF als Summe gewichteter Komponentenfunktionen. Quelle: [3]

Eine Komponentenfunktion CF_i läst sich mit Hilfe weniger Parameter beschreiben (siehe Abbildung 3-3). S_a, S_b, S_c, S_d legen die Position der Stützstellen fest. Über die Parameter C_a, C_b, C_c, C_d wird jeder Stützstelle eine Farbe zugeordnet. Die Amplitude der Komponentenfunktion ist durch deren Parameter A_i festgelegt.

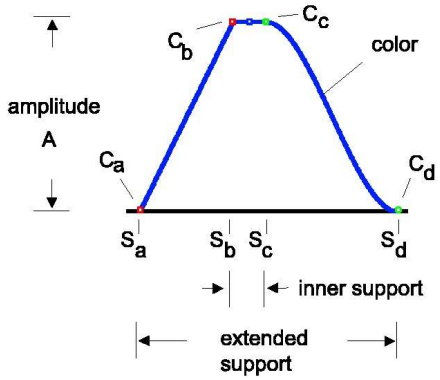


Abbildung 3-3: Parametrisierung der Komponentenfunktion. Quelle:[3]

Die Komponentenfunktionen sind wie folgt definiert:

$$CF_i(x) = \begin{cases} 0 & 0 \le x < Sa_i \\ A_i \cdot cfl_i(x) & Sa_i \le x < Sb_i \\ A_i & Sb_i \le x < Sc_i \\ A_i \cdot cfr_i(x) & Sc_i \le x < Sd_i \\ 0 & Sd_i \le x < 1 \end{cases} \qquad \textbf{(3-2)}$$

Deren Wertebereich ist dabei eingeschränkt durch:

$$CF_i(x) \neq 0 \Leftrightarrow Sa_i < x < Sd_i \qquad \textbf{(3-3)}$$

Darüber hinaus dürfen sich die Stützstellenintervalle $[Sb, Sc]$ mehrerer Komponentenfunktionen nicht überschneiden. Die Funktionen cfl_i und cfr_i werden durch Polynome geringen Grades beschrieben. Die Eigenschaften dieser Polynome bestimmen die Gestalt der Komponentenfunktion. Auf diese Weise können verschiedene Typen von Komponentenfunktionen unterschieden werden. Abbildung 3-4 zeigt eine Auswahl der wichtigsten Grundtypen.

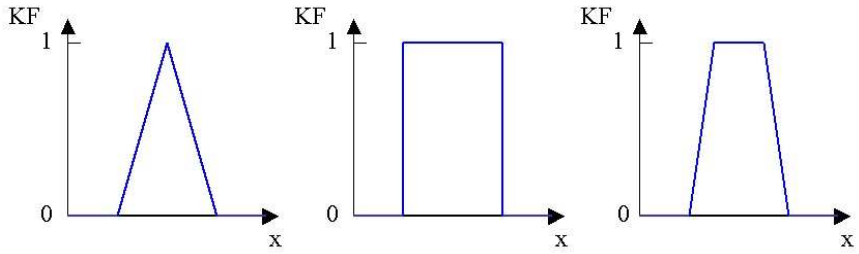


Abbildung 3-4: Typen von Komponentenfunktionen.

Bei der Definition der TFs werden Gewichte w_i verwendet. Diese sind definiert durch:

$$w_i(x) = \begin{cases} 0 & 0 \le x < Sc_{i-1} \\ p_i(x) & Sc_{i-1} \le x < Sb_i \\ 1 & Sb_i \le x \le Sc_i \\ 1 - p_{i+1}(x) & Sc_i < x \le Sb_{i+1} \\ 0 & Sb_{i+1} < x \le 1 \end{cases} \qquad \textbf{(3-4)}$$

Dabei gilt:

$$\sum_{i=1}^{n} w_i(x) = 1 \quad \forall x \in [0,1] \qquad \textbf{(3-5)}$$

Die Funktionen p_i lassen sich durch Polynome geringen Grades darstellen.
Eine Farbfunktion $Col(x)$ kann analog zur Opazitätsfunktion definiert werden:

$$Col(x) = \sum_{i=1}^{n} w_i(x) \cdot Col_i(x) \qquad \textbf{(3-6)}$$

Bewertung

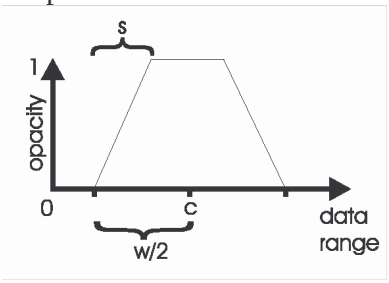
Die Anzahl der Parameter ist hoch. Für jede Komponentenfunktion müssen die Position der Stützstellen (S_a, S_b, S_c, S_d), die Farbe an den Stützstellen (C_a, C_b, C_c, C_d) sowie die Amplitude (A) eingestellt werden. Die Flexibilität beim Definitionsprozess ist als gut einzuschätzen. Es stehen verschiedene Grundtypen von Komponentenfunktionen für die Approximation einer TF zur Verfügung. Dabei können mehrere Komponentenfunktionen zu komplexen 1D-TFs kombiniert werden. Durch den Einsatz von Gewichten erfolgt eine Abrundung von Kan-

ten des Funktionsgraphs sowie ein Verschmelzen der einzelnen Komponentenfunktionen. Die Auswirkungen dieser Gewichte auf das Visualisierungsergebnis sind jedoch nur schwer einzuschätzen, da in der Quellliteratur keine Untersuchungen darüber vorgenommen werden. Mit hoher Wahrscheinlichkeit wird eine weiche Definition der Opazitätsfunktion zu unscharfen Kanten in der Visualisierung führen. Darüber hinaus könnten an den Übergängen der Komponentenfunktionen unbeabsichtigte Farbinterpolationen auftreten. Eine Erweiterung für 2D-TFs ist relativ aufwändig, da bereits die 1D-Repräsentation recht komplex ist. Der Aufwand bei der Berechnung von LUTs ist als eher gering einzuschätzen, da vorwiegend lineare Interpolation genutzt werden kann. Die Auswertung der Polynome cfl_i, cfr_i und p_i wird einen Großteil der benötigten Rechenzeit in Anspruch nehmen.

3.1.3 Stückweise lineare Funktionen - ungewichtet

Stückweise lineare Funktionen werden Aufgrund ihrer geringen Anzahl an Parametern sehr häufig für die Definition von 1D-TFs verwendet. In [4] wird ein solcher Ansatz vorgestellt. TFs werden dabei aus „Peaks“ zusammengesetzt. Es können verschiedene Grundformen dieser Peaks realisiert werden. In Abbildung 3-5 sind die verschiedenen Grundtypen sowie ihre mathematische Beschreibung zusammengestellt. Sie verfügen über maximal drei Parameter. Das Zentrum c definiert die Position des Peaks auf der Datenachse, w die Breite des Peaks. Die Trapezform verfügt zusätzlich über den Parameter s, der die Steigung an den Flanken des Trapezes beeinflusst.

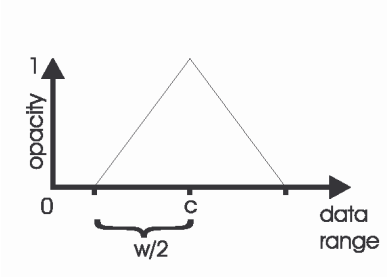
Trapez:



(3-7)

$$p(x)=\begin{cases}0 & 0\le x\le c-w/2\\ (x-c-w/2)/s & c-w/2<x\le c-w/2+s\\ 1 & c-w/2+s<x<c+w/2-s\\ 1-(x-c-w/2+s)/s & c+w/2-s<x\le c+w/2\\ 0 & c+w/2<x\le 1\end{cases}$$

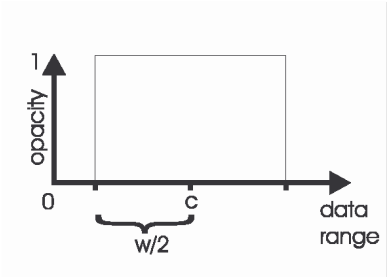
Hut:



(3-8)

$$p(x)=\begin{cases}0 & 0\le x\le c-w/2\\ (x-c-w/2)/(w/2) & c-w/2<x\le c\\ 1-(x-c)/(w/2) & c<x<c+w/2\\ 0 & c+w/2\le x\le 1\end{cases}$$

Rechteck:

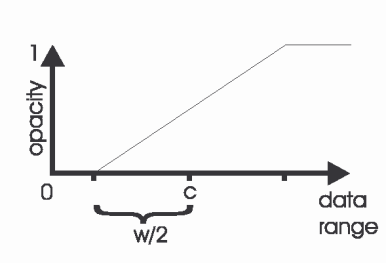


(3-9)

$$p(x)=\begin{cases}0 & 0\le x\le c-w/2\\ 1 & c-w/2<x<c+w/2\\ 0 & c+w/2\le x\le 1\end{cases}$$

Rampe:

(3-10)



$$p(x)=\begin{cases}0 & 0\le x\le c-w/2\\ (x-c-w/2)/w & c-w/2<x<c+w/2\\ 1 & c+w/2\le x\le 1\end{cases}$$

Abbildung 3-5: Grundtypen von Peaks.

Üblicherweise werden TFs durch nur einen Peak definiert. Die Interaktionsparameter sind deshalb sehr überschaubar. Es ist jedoch auch möglich, mehrere Peaks aneinander zu reihen und so komplexere Funktionen zu erzeugen.

Eine Opazitätsfunktion kann durch:

$$O(x)=\sum_{i=1}^{n} p_i(x) * A_i \qquad \textbf{(3-11)}$$
$$i \in N$$

definiert werden. A_i gibt dabei die Amplitude der Funktion und p_i den verwendeten Peak an. Eine Farbfunktion lässt sich durch:

$$C(x)=\begin{cases}color_i & p_i(x)\neq 0\\ black & sonst\end{cases} \qquad \textbf{(3-12)}$$

definieren. Dabei bezeichnet $color_i$ die Farbe des i'ten Peaks. Ein ähnlicher Ansatz wird in [36] vorgestellt.

Bewertung

Die geringe Anzahl an Parametern ermöglicht den Einsatz einfacher Interaktionstechniken. Für die Definition einer Rampe genügt es beispielsweise Mittelpunkt und Breite anzugeben. Dieses Konzept ist leicht verständlich, einfach anzuwenden und aus diesen Gründen auch sehr verbreitet. Positiv könnte sich deshalb auswirken, dass bei der Verwendung ähnlicher Konzepte auf Erfahrungen der Anwender zurückgegriffen werden kann und so mit kurzen Einarbeitungszeiten zu rechnen ist. Die Flexibilität beim Definitionsprozess ist als mittelmäßig einzuschätzen. Der Anwender ist bei der Definition von TFs auf wenige Grundtypen von Peaks eingeschränkt. In den meisten Fällen lassen sich damit jedoch gute Ergebnisse erzielen. Die Berechnung von LUTs kann vollständig mittels linearer Interpolation erfolgen und ist daher sehr schnell. Eine Erweiterung dieses Konzepts auf 2D ist leicht möglich.

3.1.4 Stückweise lineare Funktionen - manuelles Setzen von Stützstellen

Ein weiteres Konzept für die Definition stückweise linearer Funktionen besteht darin, Stützstellen der TF manuell festzulegen. Dies kann über die Positionierung von Punkten in einem 2D-Koordinatensystem geschehen. Auf diese Weise wird eine Zuordnung von Intensitätswerten zu Opazität und Grauwerten / Farbe vorgenommen. Die Definition von Farbfunktionen geschieht meist durch das Färben der Kontrollpunkte an den Stützstellen der Opazitätsfunktion oder durch ein Festlegen zusätzlicher Funktionen für die Farbkanäle. Abbildung 3-6 zeigt einen TF-Editor, in dem über vier Stützstellen eine Opazitätsfunktion festgelegt wurde.

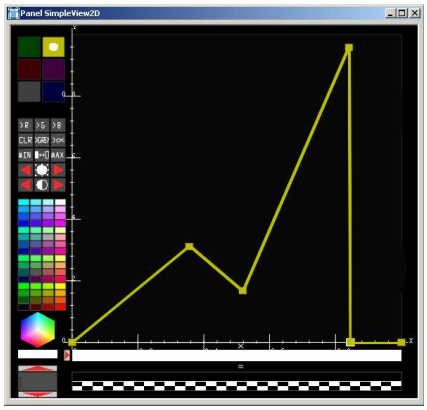


Abbildung 3-6: Editor für die Definition von TFs mit Hilfe von Stützstellen.

Bewertung

Jede Stützstelle der TF muss einzeln positioniert werden. Dafür ist ein hohes Maß an Erfahrung notwendig. Besonders wenn mehrere Kanäle der TF zu definieren sind, kann dies sehr zeitaufwändig sein. Positiv ist das hohe Maß an Flexibilität. Durch das Einfügen beliebig vieler Stützstellen lassen sich komplexe TFs erzeugen. LUTs können schnell berechnet werden, da auch hier lediglich eine Interpolation zwischen den Stützstellen erforderlich ist. Dieses Konzept lässt sich nur mit hohem Aufwand für 2D-TFs erweitern, da hierfür Stützstellen im dreidimensionalen Raum zu platzieren sind.

3.1.5 Parametrische kubische Kurven

Eine weitere Möglichkeit für die Repräsentation von 1D-TFs ist der Einsatz von parametrischen kubischen Kurven. Es stehen verschiedene Kurventypen zur Verfügung. Die bekanntesten sind Bézierkurven, Hermite-Splines und B-Splines [5, 25]. Diese Kurven können aus mehreren Elementen zusammengesetzt werden (stückweise Polynom-Kurve). Dabei ist zwischen den Kurvenelementen leicht C^1-Stetigkeit, bei B-Splines sogar C^2-Stetigkeit zu erreichen. Ein einzelnes Kurvenelement lässt sich mit Hilfe von vier Kontrollpunkten steuern. Ein Vorteil von Bézierkurven und Hermite-Splines ist, dass sich je ein Kontrollpunkt am Anfang und am Ende jedes Kurvenelementes befindet. Da die beiden Kontrollpunkte direkt auf der Kurve liegen, besteht eine gute Kontrollmöglichkeit über den Start- und Endpunkt jedes Kurvenelementes. Der Kurvenverlauf wird mit Hilfe der anderen zwei Kontrollpunkte festgelegt. Bei B-Splines ist das nicht der Fall. Hier befindet sich keiner der Kontrollpunkte auf der Kurve. Eine interaktive Definition gestaltet sich deshalb schwieriger.

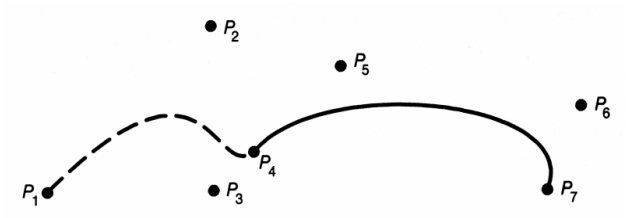


Abbildung 3-7: Bézierkurve, zwei Kurvenelemente verbunden im Kontrollpunkt P4. Quelle: [5]

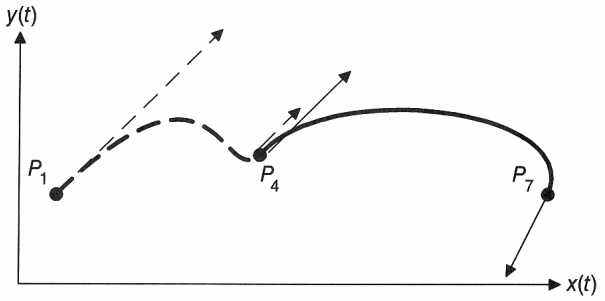


Abbildung 3-8: Hermitekurve, zwei Kurvenelemente verbunden im Kontrollpunkt P4. Quelle: [5]

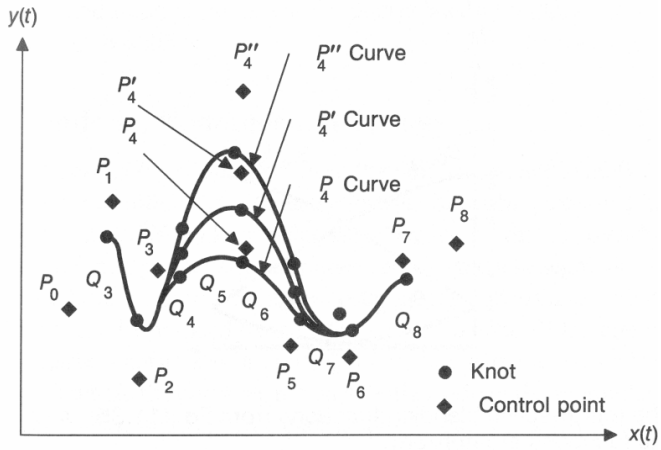


Abbildung 3-9: B-Sline, mit unterschiedlicher Positionierung des Kontrollpunktes P4. Quelle: [5]

Bewertung

Ein Nachteil dieser Repräsentationsformen ist ihre hohe Anzahl an Parametern. Außerdem sind für die Definition Kenntnisse über das Kurvenverhalten erforderlich. Besonders dann, wenn Kontrollpunkte nicht von der Kurve interpoliert werden, ist ihr Einfluss auf das Kurvenverhalten schwer abzuschätzen. Ein Vorteil liegt in einer hohen Flexibilität, die es erlaubt, sehr komplexe TFs zu erzeugen, was in den meisten Fällen jedoch nicht erforderlich ist. Das erklärt, weshalb diese Repräsentation in der Praxis kaum zur Anwendung kommt. Die Berechnung von LUTs ist mit vergleichsweise hohem Rechenaufwand verbunden, da kubische Polynome ausgewertet werden müssen. Die Kurven lassen sich leicht zu Flächen (biparametrische kubische Paches) zusammensetzten, eine Erweiterung für 2D-TFs ist also möglich. Das Positionieren der Kontrollpunkte könnte sich dabei jedoch als schwierig erweisen, weil

hierfür 3D-Interaktionskonzepte benötigt werden. Darüber hinaus ist eine Definition mit Hilfe von quadratischen Kurven möglich. Dabei sind im Vergleich zu kubischen Kurven Einschränkungen bei der Flexibilität hinzunehmen.

3.1.6 Vergleich der Repräsentationsformen

In Tabelle 3-1 wird eine abschließender Vergleich der Repräsentationsformen hinsichtlich der aufgestellten Anforderungen vorgenommen. Bewertet wird dabei die Flexibilität des Definitionsprozesses, die Anzahl der benötigten Parameter (möglichst wenige), der Aufwand bei der LUT-Berechnung sowie die Eignung für eine Erweiterung auf 2D.

	Stückweise lineare Funktionen - gewichtet	Stückweise lineare Funktionen - ungewichtet (Peaks)	Stückweise lineare Funktionen - Stützstellen	parametrisierte kubischen Kurven
Flexibilität	+	+	++	++
Anzahl Parameter	O	++	--	--
LUT-Berechnung	+	++	++	--
Erweiterung 2D	O	+	-	-

Tabelle 3-1: Vergleich von Repräsentationsformen für 1D-TFs.
(++ sehr gut, + gut, O ausreichend, - befriedigend, -- ungenügend)

Dabei tritt die Repräsentation der ungewichteten stückweise linearen Funktion besonders positiv hervor. Sie verfügt über wenige Parameter (2-3) und unterstützt aufgrund ihrer stückweisen Linearität eine effiziente Berechnung von LUTs. Darüber hinaus ist eine ausreichende Flexibilität des Definitionsprozesses gegeben. Auch eine Erweiterung auf 2D ist leicht möglich. Es ist zu beachten, dass diese Beurteilung ausschließlich auf den aufgestellten Anforderungen beruht. Für abweichende Anwendungsgebiete können sich durchaus andere Repräsentationsformen als nützlicher erweisen. Wenn beispielsweise eine hohe Flexibilität beim Definitionsprozess im Vordergrund steht, stellt die Verwendung von parametrisierten kubischen Kurven oder stückweise linearer Funktionen, die über Stützstellen definiert werden, sicherlich den besseren Weg dar.

3.2 Ableitung von Repräsentationen für 2D-TFs

In Abschnitt 3.2.1 und 3.2.2 werden zwei Repräsentationen für 2D-TFs vorgestellt. Die erste basiert auf einer Approximation mit Hilfe von 2D-Komponentenfunktionen, bei der zweiten werden 2D-TFs durch Interpolation zwischen 1D-TFs beschrieben.

3.2.1 2D-Komponentenfunktionen

Basierend auf der Definition ungewichteter stückweise linearer Funktionen, wie sie in Abschnitt 3.1.3 beschrieben wurde, wird im Folgenden eine Repräsentationsform für 2D-TFs abgeleitet.

Eine 2D-TF $TF(x,y)$ lässt sich als Summe von Komponentenfunktionen $CF_i(x,y)$ definieren:

$$TF(x,y) = \sum_{i=1}^{n} CF_i(x,y) \qquad \textbf{(3-13)}$$
$$i \in N$$

Die Komponentenfunktionen sind stückweise linear. Sie werden mit Hilfe von Stützstellen beschrieben, die in einem rectilinearen Gitter angeordnet sind (siehe Abbildung 3-11).

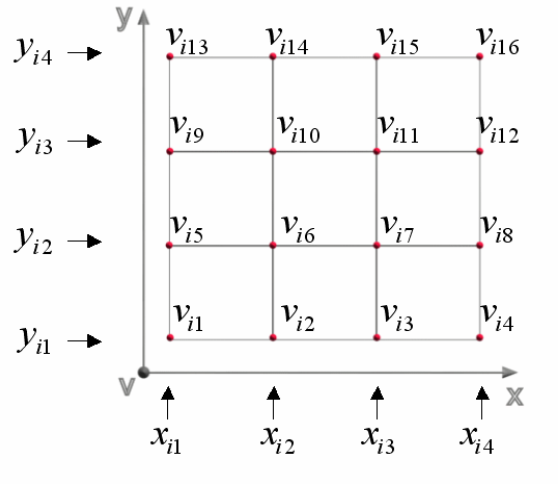


Abbildung 3-10: Parametrierung der Komponentenfunktionen.

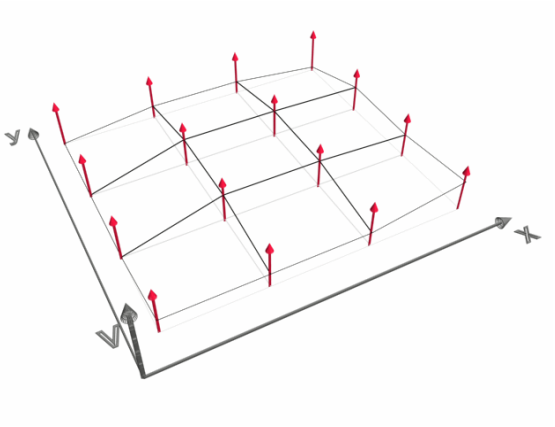


Abbildung 3-11: Anordnung der Stützstellen von Komponentenfunktionen in einem rectilinearen Gitter.

Eine formale Definition der Komponentenfunktionen wird festgelegt durch:

(3-14)

$$CF_i(x,y) = \begin{cases} 0 & y < y_{i1} \vee y > y_{i4} \vee x < x_{i1} \vee x > x_{i4} \\ bi(v_{i1}, v_{i2}, v_{i6}, v_{i5}, x, x_{i1}, x_{i2}, y, y_{i1}, y_{i2}) * A_i & x > x_{i1} \wedge x < x_{i2} \wedge y > y_{i1} \wedge y < y_{i2} \\ bi(v_{i2}, v_{i3}, v_{i7}, v_{i6}, x, x_{i2}, x_{i3}, y, y_{i1}, y_{i2}) * A_i & x > x_{i2} \wedge x < x_{i3} \wedge y > y_{i1} \wedge y < y_{i2} \\ \ldots & \end{cases}$$

$\forall A_i \in [0,1]$

$\forall x, y \in R$

$x_{i1},\ldots,x_{i4} \in R$

$y_{i1},\ldots,y_{i4} \in R$

$v_{i1},\ldots,v_{i16} \in [0,1]$

Die Komponentenfunktionen sind in Abhängigkeit von verschiedenen Parameter definiert (siehe Abbildung 3-10). $x_{i1},\ldots,x_{i4}$ und $y_{i1},\ldots,y_{i4}$ legen die Positionen der Gitterlinien fest, auf denen die Stützstellen angeordnet sind. Die Funktionswerte an den Stützstellen werden durch die Parameter $v_{i1},\ldots,v_{i16}$ repräsentiert. A_i definiert die Amplitude der Komponentenfunktionen.

Die Funktion *bi* berechnet eine bilineare Interpolation zwischen jeweils vier Stützstellen einer Komponentenfunktion. Sie kann definiert werden durch:

(3-15)

$$bi(v_0, v_1, v_2, v_3, x, x_{\min}, x_{\max}, y, y_{\min}, y_{\max}) = w_0 * v_0 + w_1 * v_1 + w_2 * v_2 + w_3 * v_3$$

mit

$$w_0 = (1-r)*(1-s)$$
$$w_1 = (1-r)*s$$
$$w_2 = r*s$$
$$w_3 = r*(1-s)$$

und

$$s = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$
$$r = \frac{y - y_{\min}}{y_{\max} - y_{\min}}$$

$v_0,...,v_3$ geben dabei die Werte an den Stützstellen der Komponentenfunktion an, zwischen denen mit Hilfe der Gewichte $w_0,...,w_3$ interpoliert wird. $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ definieren die Positionen der Gitterlinien, auf denen sich die Stützstellen befinden. (x , y) ist die Koordinate des zu interpolierenden Punktes. Der Graph einer Komponentenfunktion wird durch die Positionierung ihrer Gitterlinien (Stützstellen), die Definition von Funktionswerten an den Stützstellen sowie ihrer Amplitude bestimmt. Durch unterschiedliche Parametrisierung lassen sich verschiedene Grundtypen von Komponentenfunktionen erzeugen. In Abbildung 3-12 sind einige davon zusammengestellt, die sich besonders gut für die Definition von 2D-TFs eignen.

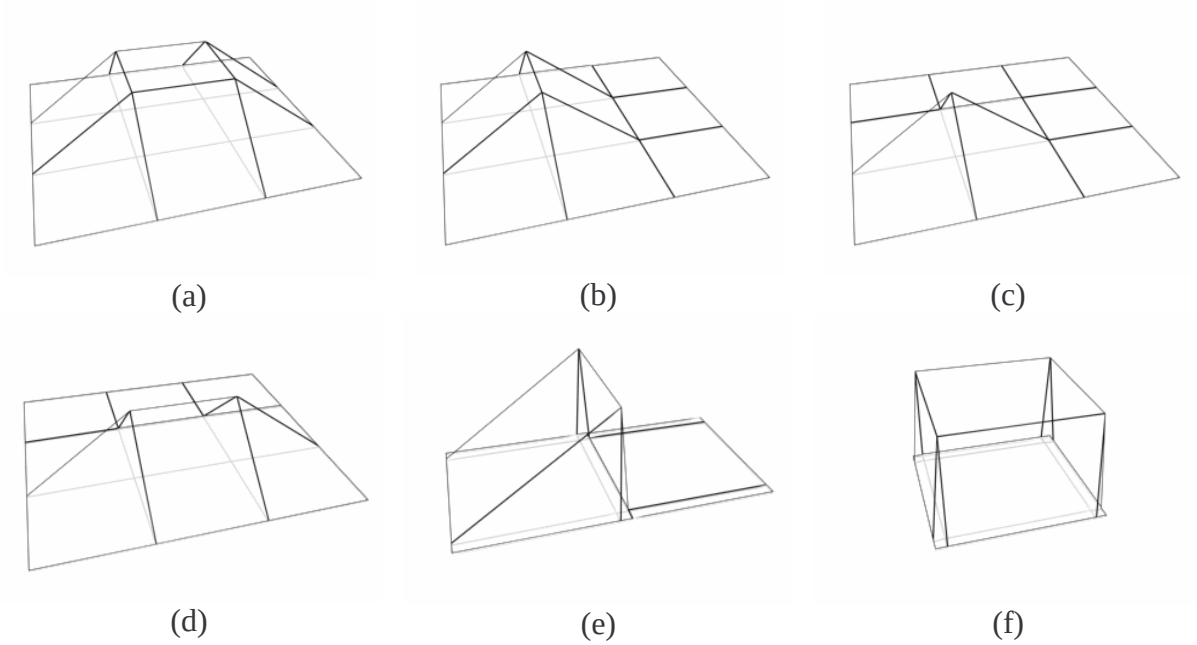


Abbildung 3-12: Grundtypen von 2D-Komponentenfunktionen.

Diese Grundtypen lassen sich durch Aneinanderfügen (Abbildung 3-13) und Überlagern (Abbildung 3-14) leicht zu komplexeren 2D-TFs kombinieren.

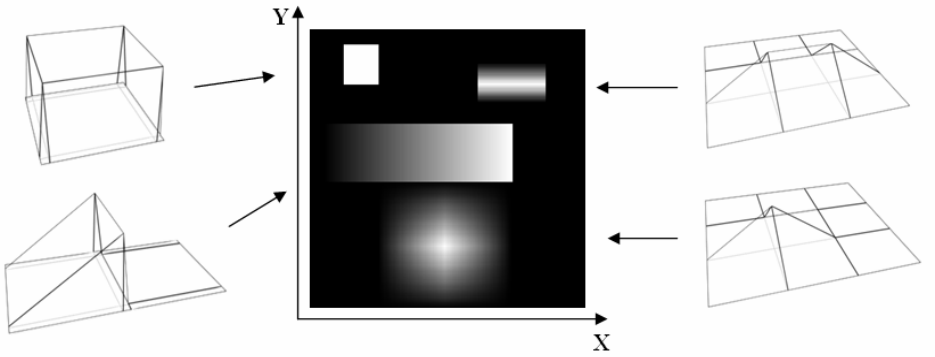


Abbildung 3-13: Kombination mehrerer Komponentenfunktionen durch Aneinanderfügen.

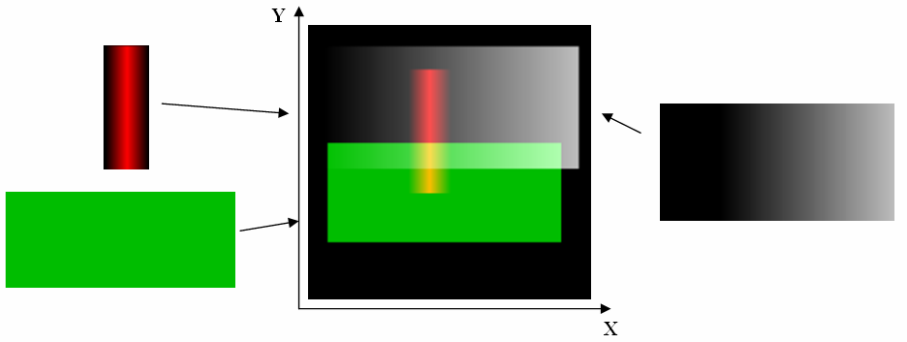


Abbildung 3-14: Kombination mehrerer Komponentenfunktionen durch additive Überlagerung. Darüber hinaus sind weitere Überlagerungsmodi, wie das Priorisieren der Komponentenfunktionen mit kleinem bzw. größerem Index oder einem Überblenden zwischen sich überlagernden Funktionen, denkbar. Wenn die Komponentenfunktionen über vier Kanäle (RGBA) definiert werden, ist der Einsatz von Farbe möglich. Dies erlaubt eine Differenzierung von Strukturen in der Visualisierung durch unterschiedliche Färbung.

Geeignete Interaktionsparameter

Um eine einfache interaktive Definition von TFs zu ermöglichen, ist es notwendig, deren Parameter zu reduzieren. Dabei ist darauf zu achten, dass die Flexibilität beim Definitionsprozess erhalten bleibt. Ausgangspunkte hierfür sind die Parameter der Komponentenfunktionen $x_{i1},...,x_{i4}, y_{i1},...,y_{i4}, v_{i1},...,v_{i16}, A_i$.

Zwei wesentliche Parameter bei der Definition 2D-TFs sind die Intervalle der abhängigen Variablen *x* und *y*. Ihnen kommt je nach verwendeten Bildeigenschaften unterschiedliche Bedeutung zu. Werden TFs, wie in dieser Arbeit beschrieben, in Abhängigkeit von der Bildintensität und von Distanzen zu einer Referenzfläche im Datensatz definiert, wird über das Intensitätsintervall der Gewebetyp und über das Distanzintervall der Abstand der darzustellenden Gewebe zur Referenzfläche festgelegt. Diese Parameter sind für die Definition von 2D-TFs essentiell. Um die Intervalle der abhängigen Variablen $[x_{min}, x_{max}]$ und $[y_{min}, y_{max}]$ auf

die Parameter der Komponentenfunktionen $x_{i1},...,x_{i4}$ und $y_{i1},...,y_{i4}$ abzubilden, wird eine Konstante ε eingeführt und folgendes definiert:

$$\begin{aligned} x_{i1} &= x_{\min} & y_{i1} &= y_{\min} \\ x_{i2} &= x_{\min} + \varepsilon & y_{i2} &= y_{\min} + \varepsilon \\ x_{i3} &= x_{\max} - \varepsilon & y_{i3} &= y_{\max} - \varepsilon \\ x_{i4} &= x_{\max} & y_{i4} &= y_{\max} \end{aligned} \qquad \textbf{(3-16)}$$

x_{i1}, x_{i4}, y_{i1} und y_{i4} liegen damit auf den Intervallgrenzen. x_{i2}, x_{i3}, y_{i2} und y_{i3} sind ausgehend von diesen Grenzen um den Betrag ε (sehr klein) in das Intervall hinein verschoben. Um die Parameter $x_{i1},...,x_{i4}$ und $y_{i1},...,y_{i4}$ festzulegen, ist lediglich die Definition des *x*- und *y*-Intervalls (dies kann z.B. einem Intensitätsintervall und einem Distanzintervall entsprechen) erforderlich.

Eine weitere wichtige Eigenschaft von TFs ist ihr Funktionsverlauf. Der Verlauf von Komponentenfunktionen wird durch die Parameter $v_{i1},...,v_{i16}$ (Funktionswerte an den Stützstellen) repräsentiert. Diese können am einfachsten durch die Auswahl eines Funktionstyps (siehe Abbildung 3-12) gesetzt werden. Mit Hilfe des Typs (e) ließe sich beispielsweise eine Rampe entlang der Intensitätswerte erzeugen, während Typ (f) konstante Werte innerhalb des Intensitäts- und Distanzintervalls definiert. Je nach Bedarf lassen sich beliebige Grundtypen festlegen.

Ein weiterer wesentlicher Parameter ist die Amplitude A_i. Diese kann in Abhängigkeit von der Dimension des Wertebereiches skalar sein oder auch einen Vektor bilden. Soll eine TF beispielsweise auf den RGBA-Farbraum abbilden, ist die Amplitude ein Quartupel welches jeweils einen Intensitätswert der Rot-, Grün-, Blau- und Alphakomponente des Farbraumes repräsentiert. Die Amplitude wird für jede Komponentenfunktion getrennt festgelegt.

Zusammenfassend ist zu sagen, dass die Definition von Komponentenfunktionen durch das Festlegen folgender Interaktionsparameter erfolgen kann:

- *Intervall der abhängigen Variable x*: Bei distanzabhängigen TFs ist dies ein Intensitätsintervall. Komponentenfunktionen können über verschiedenen Intensitätsintervallen aufgespannt werden. Dabei wird der Gewebetyp festgelegt, auf den die Komponentenfunktion Einfluss nimmt z.B.: Knochen, Muskeln, Fett, Haut etc.
- *Intervall der abhängigen Variable y*: Bei distanzabhängigen TFs ist dies ein Distanzintervall. Dabei wird der Distanzbereich festgelegt, auf den eine Komponentenfunktion Einfluss nimmt, beispielsweise auf Gewebe, dass sich in einem Abstand von 5 bis 10 mm von einer ausgewiesenen Referenzfläche befindet.
- *Funktionstyp*: Der Funktionstyp definiert, wie sich die Komponentenfunktion innerhalb der festgelegten Intervalle auswirkt. Es stehen verschiedene Grundtypen zur Auswahl.
- *Amplitude (Farbe, Transparenz)*: Die Amplitude legt die maximale Intensität der Komponentenfunktion für alle Dimensionen ihres Wertebereiches (Farbraum) fest. Meist werden in diesem Zusammenhang RGBA-Werte verwendet.

Alle Parameter, der in diesem Abschnitt beschriebenen Repräsentation, können daraus abgeleitet werden. Dabei ist jedoch anzumerken, dass eine Reduktion der Parameter immer mit Einschränkungen beim Definitionsprozess verbunden ist. Falls diese für einzelne Anwendungen zu groß erscheinen, sind geeignetere Interaktionsparameter zu wählen. Für die meisten Anwendungen werden sich die angegebenen Interaktionsparameter jedoch als sinnvoll und ausreichend erweisen.

Vorschläge für die Präsentation der Interaktionsparameter

Wie lassen sich die definierten Parameter in einer Benutzeroberfläche darstellen? - Im Folgenden werden hierfür mögliche Konzepte diskutiert.

Die Intervalle der Variablen *x* und *y*, in denen eine Komponentenfunktion definiert ist, lassen sich gut mit Hilfe von Slidern einstellen. Der minimalen bzw. maximalen Position eines Sliders wird hierfür der minimale bzw. maximale Werte der zugeordneten Bildeigenschaft zugewiesen. Für die Definition eines Intervalls sind zwei Slider erforderlich. Einer kennzeichnet dabei den Beginn, der andere das Ende des Intervalls (siehe Abbildung 3-15).

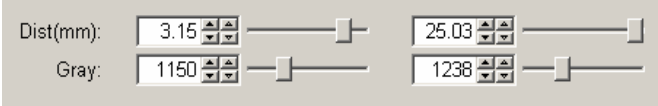


Abbildung 3-15: Einstellen des Definitionsbereiches der Komponentenfunktionen mittels Slidern. In diesem Beispiel wird für die Komponentenfunktion das Distanzintervall [3.15, 25.03] und das Intensitätsintervall [1150, 1238] festgelegt.

Eine weitere Möglichkeit diese Intervalle einzustellen, besteht in einer Assoziation der Intervallgrenzen mit den Koordinaten der Eckpunkte eines Rechtecks. Das Rechteck wird hierzu im zweidimensionalen Argumentraum der Komponentenfunktion (*x*,*y*) aufgespannt (siehe Abbildung 3-16). An den Eckpunkten befinden sind Kontrollpunkte, die sich mit der Maus verschieben lassen. Für eine besseren Orientierung ist es sinnvoll im Hintergrund ein 2D-Histogramm des zu visualisierenden Datensatzes einzublenden. Mit Hilfe des Histogramms können Gewebe, die sich auf Grund ihrer Eigenschaften (*x*,*y*) von anderen unterscheiden, differenziert werden.

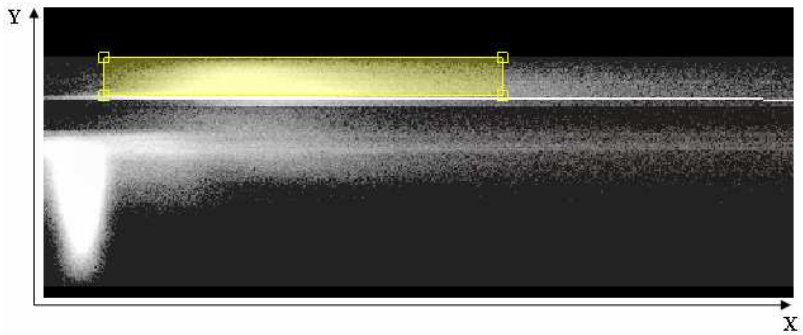


Abbildung 3-16: Einstellen des Definitionsbereiches der Komponentenfunktionen über die Koordinaten eines aufgespannten Rechtecks.

Die Auswahl einer Amplitude (Farbe, Transparenz) kann mit Hilfe von Farbpaletten erfolgen (Abbildung 3-17). Hierbei können vordefinierte Farben verwendet oder neue festgelegt werden. Dabei ist darauf zu achten, dass die verwendete Palette Transparenzwerte unterstützt. Ist dies nicht der Fall, muss das Festlegen der Transparenz in Form eines skalaren Wertes zwischen 0 und 1 getrennt von der Farbdefinition erfolgen. Für die Auswahl des Funktionstyps bieten sich Buttons oder Listenfelder an. Es ist hierbei sinnvoll den Funktionsverlauf skizzenhaft anzudeuten (Abbildung 3-18), aber auch rein textuelle Beschreibungen sind möglich.

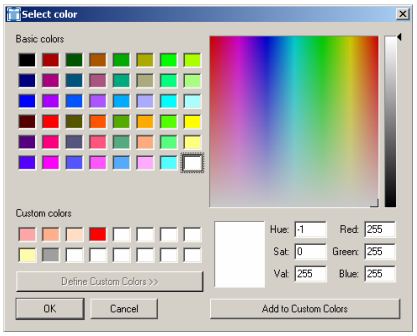


Abbildung 3-17: Auswahl von Farben mittels Farbtabellen.

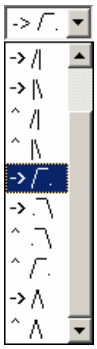


Abbildung 3-18: Auswahl des Funktionstyps über ein Listenfeld.

Für die Auswahl des Funktionsverlaufs sollten folgende Grundtypen bereitgestellt werden:

#	-	Funktion Konstant entlang der Grauwertachse (G), sowie konstant entlang der Distanzachse (D), siehe Abbildung 3-12 (f)
> /|	-	Funktion linear ansteigend in G, konstant in D, siehe Abbildung 3-12 (e)
> /‾.	-	Funktion innerhalb des definierten Intervalls linear steigend in G, danach konstant, konstant in D
> /\	-	Funktion steigend und danach fallend in G, konstant in D, siehe Abbildung 3-12 (b)

Darüber hinaus können erweiterte Funktionstypen angeboten werden:

×	-	Funktion steigend und danach fallend in G und D, siehe Abbildung 3-12 (c)
> |\	-	Funktion linear fallend in G, konstant in D
^ /|	-	Funktion konstant in G, linear steigend in D
^ |\	-	Funktion konstant in G, linear fallend in D
> .‾\	-	Funktion innerhalb des definierten Intervalls konstant in G, danach linear fallend in G, konstant in D
^ .‾\	-	Funktion innerhalb des definierten Intervalls konstant in D, danach linear fallend in D, konstant in G
^ /‾.	-	Funktion innerhalb des definierten Intervalls steigend in D, danach konstant in D, konstant in G
^ /\	-	Funktion steigend und danach fallend in D, konstant in G, siehe Abbildung 3-12 (d)

Beurteilung der Repräsentation

2D-Komponentenfunktionen zeichnen sich durch eine relativ geringe Anzahl an Parametern aus. Dennoch ist eine hohe Flexibilität des Definitionsprozesses sichergestellt. TFs werden als Summe von Komponentenfunktionen betrachtet. Dabei findet eine starke Abstraktion statt, da TFs durch Parameter eines Repräsentationsmodells dargestellt werden. Das zugrunde liegende Konzept ähnelt der Definition von 1D-TFs unter Verwendung von Rampen. Es stellt damit die Erweiterung eines bewährten Ansatzes dar.

3.2.2 Interpolation zwischen 1D-TFs

Ein weiterer Ansatz für den Entwurf von 2D-TFs basiert auf der Interpolation zwischen 1D-TFs. Definitionsmöglichkeiten für 1D-TFs wurden in Abschnitt 3.1 vorgestellt. Die zugrunde liegende Idee ist, mehrere dieser 1D-TFs miteinander zu 2D-TFs zu kombinieren. Dies kann durch Interpolation geschehen (siehe Abbildung 3-19). Ausgangspunkt hierfür ist eine Menge stückweise linearer 1D-TFs.

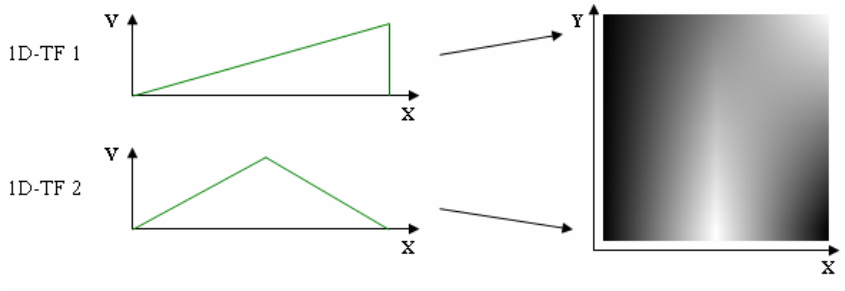


Abbildung 3-19: Kombination mehrerer stückweise linearer 1D-TFs mittels Interpolation.

Angleichen von Stützstellen

Für die Interpolation ist es vorteilhaft, wenn die 1D-TFs an denselben Positionen über Stützstellen verfügen. Um dies zu gewährleisten, sind ihnen die Stützstellen der jeweils anderen Funktionen hinzuzufügen. Die Funktionswerte an den eingefügten Stützstellen lassen sich linear aus den bereits vorhandenen interpolieren. Außerdem ist jeder TF ein y-Wert zuzuordnen. Abbildung 3-20 (links) zeigt dies am Beispiel zweier 1D-TFs.

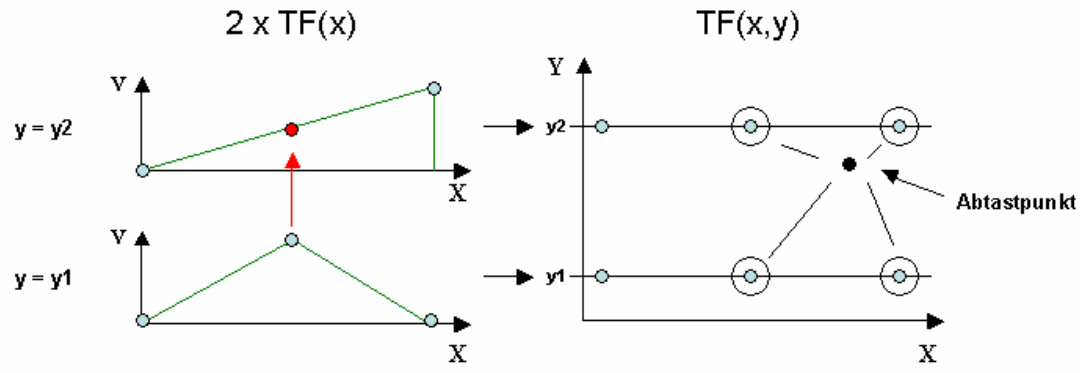


Abbildung 3-20: Im linken Bereich der Abbildung ist das Angleichen der Stützstellen zweier 1D-TFs dargestellt. Der rote Punkt kennzeichnet dabei die eingefügte Stützstelle. Der rechte Bereich zeigt die Interpolation eines Abtastpunktes aus den Stützstellen der beiden einschließenden 1D-TFs. Der Einfachheit halber werden in diesem Beispiel lediglich zwei 1D-TFs verwendet. In der Praxis würden für eine sinnvolle Definition meist mehr benötigt werden.

Berechnung der Interpolation

Die Berechnung eines Funktionswertes der 2D-TF ist sehr einfach. Hierfür werden zunächst durch einen Vergleich der y-Werte der 1D-TFs die beiden 1D-TFs bestimmt, die sich in y-Richtung direkt oberhalb bzw. unterhalb des Abtastpunktes befinden. Anschließend werden durch einen Vergleich der x-Werte die Stützstellen der beiden 1D-TFs ermittelt, die links und rechts neben dem Abtastpunkt liegen. Der Funktionswert am Abtastpunkt kann bilinear aus diesen vier Stützstellen interpoliert werden (Abbildung 3-20, rechts). Falls sich der Abtastpunkt direkt auf einer Stützstelle befindet, entfällt diese Interpolation. Liegt der Abtastpunkt oberhalb der 1D-TF mit dem maximalen y-Wert kann seine Koordinate auf diesen y-Wert verschoben werden. Analog hierzu kann bei einem unterschreiten des minimalen y-Wertes vorgegangen werden. Dadurch wird eine Extrapolation der 1D-TFs außerhalb des eingeschlossenen y-Intervalls erreicht.

Interaktionsparameter

Um 1D-TFs zu definieren, bietet sich der Einsatz von ungewichteten stückweise linearen Funktionen an (siehe Anschnitt 3.1.3). Bei dieser Repräsentationsform sind lediglich zwei bis drei Parameter festzulegen, was eine recht einfache Definition ermöglicht. Die Parameter können beispielsweise mit Hilfe von Slidern eingestellt werden. Da mehrere 1D-TFs benötigt werden, ist die Anzahl der zu definierenden Parameter recht groß. Der Aufwand bei der Definition legitimiert sich jedoch durch ein hohes Maß an Flexibilität. Die Anzahl der Parameter kann auf Kosten der Flexibilität reduziert werden, indem z.B. Gruppen von vordefinierten 1D-TFs angeboten werden. Das Vorgehen hierbei ist vom jeweiligen Visualisierungsziel abhängig.

Beurteilung der Repräsentation

Diese Repräsentation basiert auf der Kombination von 1D-TFs, für die bereits einige Definitionskonzepte zur Verfügung stehen. Die verwendeten 1D-TFs können beliebig komplex sein, außerdem ist ihre Anzahl unbegrenzt. Auf diese Weise kann eine hohe Flexibilität gewährleistet werden. Da mehrere 1D-TFs erforderlich sind, ergibt sich eine hohe Anzahl an Parametern. Ein weiterer Nachteil liegt darin, dass die Interpolation zwischen den 1D-TFs mitunter nur schwer vom Anwender nachzuvollziehen ist. Dies gilt in besonderem Maße, wenn die TFs auf Farbe abbilden, also zwischen Farbwerten interpoliert wird. Es ist deshalb vorteilhaft, während des Definitionsprozesses Unterstützung, beispielsweise durch eine Darstellung der aktuellen 2D-LUT, anzubieten.

3.3 Vergleich der entwickelten Repräsentationsformen

Resultierend aus den unterschiedlichen Parametern der beiden Repräsentationsformen ergeben sich verschiedene Anwendungsgebiete. Sollen unterschiedliche Distanzen im Datensatz mit verschiedenen 1D-TFs dargestellt werden, wobei zwischen diesen Distanzen ein fließender Übergang der Abbildungseigenschaften von Interesse ist, empfiehlt es sich eine Interpolation zwischen 1D-TFs zu nutzten. 1D-TFs können mit Hilfe intuitiver Parameter (Einsatz von Grauwertrampen) definiert werden. Ein Nachteil besteht darin, dass mit der Anzahl der benötigten 1D-TFs der Interaktionsaufwand für die Definition steigt. Um beispielsweise ein Distanzintervall mit konstanten Abbildungseigenschafen zu visualisieren (Gewebe außerhalb

dieses Intervalls sollen ausgeblendet werden), werden vier 1D-TFs benötigt. Wesentlich eleganter lässt sich dieses Visualisierungsziel mit Hilfe von 2D-Komponentenfunktionen erreichen, da in diesem Fall nur eine Komponentenfunktion benötigt wird. Hierfür sind entsprechend weniger Parameter festzulegen. Eine Entscheidung über den Einsatz einer der beiden Repräsentationsformen sollte also in Abhängigkeit vom jeweiligen Visualisierungsziel getroffen werden.

3.4 Implementierung

Bei der Suche nach einem geeigneten Modell für die Implementierung der in den Abschnitten 3.2.1 und 3.2.2 vorgestellten Repräsentationsformen erwiesen sich rectilineare Gitter als vorteilhaft. Dieser Gittertyp beschränkt sich auf achsenparallele Gitterlinien mit variablen Abständen untereinander. Die Achsenparallelität ermöglicht eine effiziente Interpolation zwischen den Gitterknoten. Dies ist im Hinblick auf das Erzeugen von LUTs sehr nützlich. Einschränkungen für die Definition von Knoten ergeben sich aus den achsenparallelen Gitterlinien nicht, da nicht definierte Gitterknoten leicht aus definierten interpoliert bzw. extrapoliert werden können. Darüber hinaus wird durch die variablen Abstände zwischen den Gitterlinien die erforderliche Flexibilität beim Setzten von Gitterknoten gewährleistet.

Ein solches Gitter lässt sich gut für die Implementierung von 2D-TFs nutzten. Die Knoten des Gitters werden hierfür mit den Stützstellen der darzustellen TF assoziiert. Das Einfügen von Knoten in ein Gitter entspricht damit dem Einfügen von Stützstellen in einer 2D-TF. Die Knoten / Gitterlinien können intern als Felder verwaltete werden. Es ist sinnvoll, eine Sortierung nach den Koordinaten vorzunehmen, um ein Einfügen, Suchen und Löschen in O(log n) zu ermöglichen.

Initialisierung von TFs

Um ein TF zu initialisieren, werden dem Gitter vier Knoten hinzugefügt. Deren Gitterkoordinaten ergeben sich aus dem Definitionsbereich der zu repräsentierenden TF. Die Werte der Knoten (Funktionswerte der TF) werden auf „0“ gesetzt. Für die Initialisierung einer TF *TF(x,y)* mit dem Definitionsbereich $[x_{min},x_{max}]$ und $[y_{min},y_{max}]$ sind dem Gitter die Knoten $(x_{min}, y_{min}, 0)$, $(x_{max}, y_{min}, 0)$, $(x_{min}, y_{max}, 0)$, $(x_{max}, y_{max}, 0)$ hinzuzufügen.

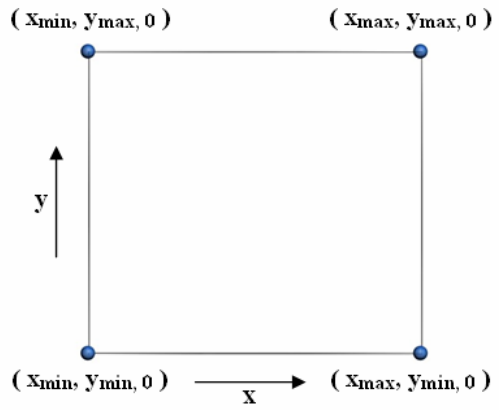


Abbildung 3-21: Initialisierung des Gitters über vier Knoten, die den Definitionsbereich der TF festlegen.

Einfügen von Stützstellen

Eine so initialisierte TF wird durch 4 Gitterknoten (Stützstellen) repräsentiert. Ihre Funktionswerte sind über den gesamten Definitionsbereich 0. Um TFs mit Funktionswerten ungleich 0 zu erzeugen, ist das Einfügen zusätzlicher Stützstellen erforderlich. Diese können beliebig innerhalb des Definitionsbereiches verteilt werden. Hierzu werden dem Gitter weitere Knoten hinzugefügt. Abbildung 3-22 stellt das Einfügen eines Knotens in ein etwas komplexeres Gitter dar. Die bereits vorhandenen Knoten sind blau, der neu eingefügte rot gekennzeichnet. Das so erzeugte Gitter ist nicht rectilinear, lässt sich jedoch durch das Einfügen zusätzlicher Knoten (grün) leicht in ein rectilineares überführen. Die Koordinaten dieser Knoten können aus den Koordinaten des neu eingefügten Knotens und aus den Positionen der Zeilen und Spalten des Gittes abgeleitet werden. Die Funktionswerte an den Knoten sind linear aus den bereits vorhandenen Knoten zu interpolieren. Dieser Schritt wird nach dem Einfügen jedes neuen Knotens durchgeführt. Wenn viele Knoten einzufügen sind, ist dieses Vorgehen jedoch ineffizient. Hierfür bietet sich eine andere Methode an. Nach dem Einfügen der Knoten werden zunächst die x-Koordinaten aller Spalten und die y-Koordinaten aller Zeilen des Gitters ermittelt. Anschließend wird für alle Permutationen der Koordinaten (Schnittpunkte der Gitterlinien) geprüft, ob ein Knoten an dieser Position existiert. Falls nicht, wird er aus den Umgebenden linear interpoliert und dem Gitter hinzugefügt.

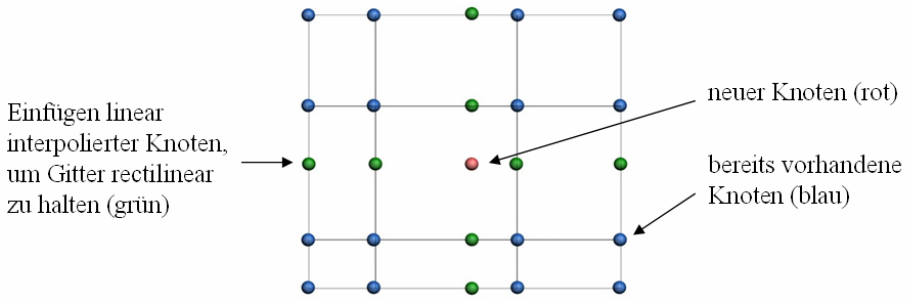


Abbildung 3-22: Einfügen weiterer Knoten (Stützstellen).

Ermitteln von Funktionswerten

Da das verwendete Gitter rectilinear ist, können die Funktionswerte an beliebigen Koordinaten bilinear aus den umgebenden Knoten interpoliert werden. Hierfür ist es erforderlich, die Knoten um den Abtastpunkt herum zu ermitteln. Dies kann durch einen Vergleich der Koordinate des Abtastpunktes mit denen der Gitterknoten erfolgen.

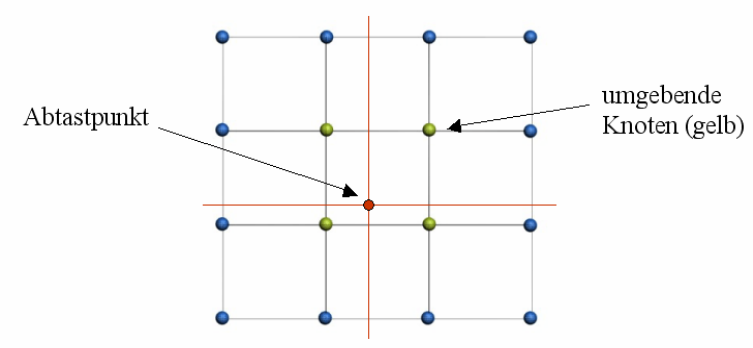


Abbildung 3-23: Ermitteln von Funktionswerten durch Interpolation.

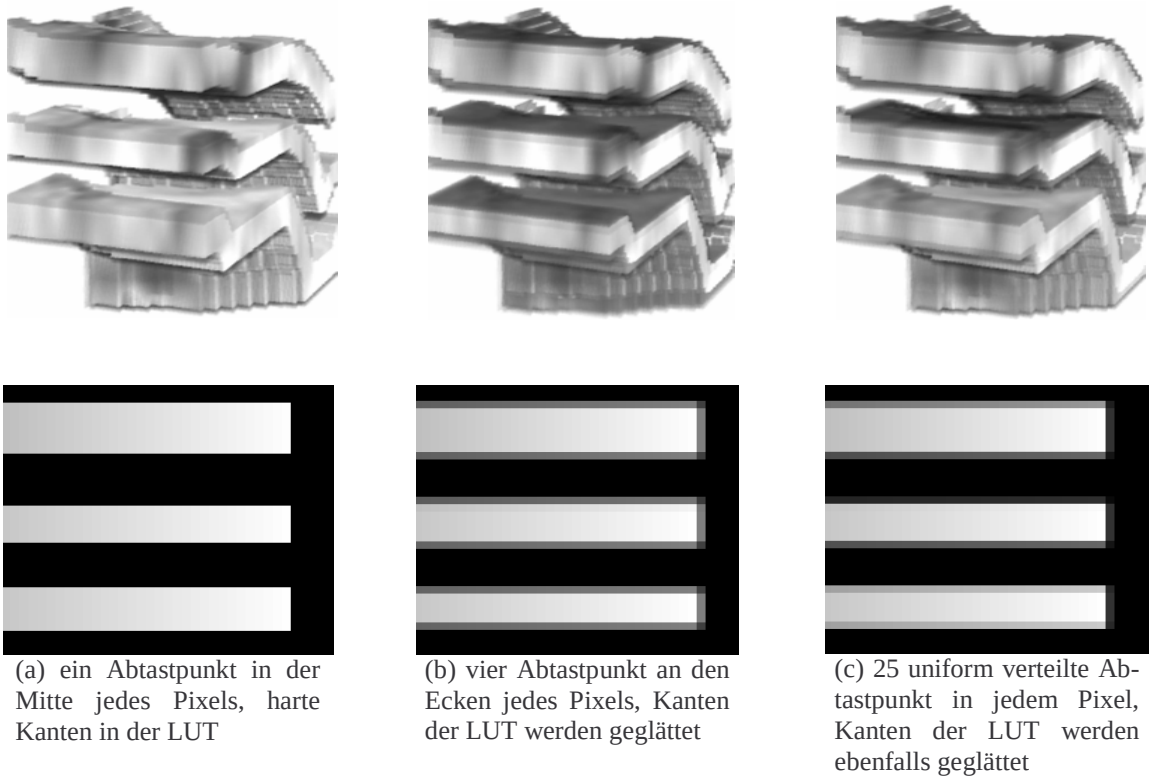
TFs mit mehrdimensionalem Wertebereich

Üblicherweise kommen TFs zum Einsatz, die auf einen mehrdimensionalen Wertebereich abbilden. Ein Beispiel wäre die Abbildung von Intensitätswerten auf Farbe und Transparenz. Dies bedeutet, dass die Knoten des Gitters keinen skalaren Wert, sondern eine Wertevektor zu repräsentieren haben. Für eine Abtastung des Gitters ergibt sich daraus, dass für jeden Abtastpunkt mehrere Werte zu interpolieren sind. Dies ist sehr effizient möglich, da die für die Interpolation der Knoten benötigten Gewichte (siehe Formel (3-15) nur einmal berechnet werden müssen.

Bei der Definition von TFs kann es erforderlich sein, die Farbfunktion getrennt von der Opazitätsfunktion festzulegen. Für beide Funktionen sind dann unterschiedliche Stützstellen und damit verschiedene Gitter erforderlich.

Erzeugen von 2D-Lookup-Tabellen

Das vorgestellte Gittermodell eignet sich gut für die Implementierung von TFs, wird in dieser Form jedoch nicht von Volumenrenderern unterstützt. Es muss deshalb in ein allgemeines Format umgewandelt werden. Ein häufig verwendetes Konzept, welches sich in diesem Zusammenhang anbietet, sind LUTs. 2D-TFs können durch 2D-LUTs repräsentiert und so von Volumenrenderern, die dies unterstützen, für die Visualisierung von Volumendatensätzen verwendet werden. Im Folgenden wird die Umwandlung des vorgestellten Gittermodells in 2D-LUTs beschrieben. Diese Umwandlung kann durch eine Abtastung des Gitters erfolgen. Hierfür ist zunächst ein Raster von Abtastpunkten über das Gitter zu legen, welches dem Pixelraster der zu erzeugenden LUT entspricht (2D-LUTs können als 2D-Bild aufgefasst werden). Dabei ist darauf zu achten, dass die Abtastpunkte korrekt über dem Gitter positioniert werden. Anschließend sind die Werte an den Abtastpunkten zu bestimmen (siehe Abschnitt 3.4). Für die Berechnung eines Pixels der LUT können mehrere Abtastpunkte verwendet werden (Überabtastung). Der Wert des Pixels ergibt sich dann aus deren Mittelwert. Die Abtastpunkte können regelmäßig oder willkürlich innerhalb eines Pixels verteilt werden. Bei einer Überabtastung ist zu beachten, dass an harten Kanten in der LUT deutliche Glättungseffekten auftreten. In der resultierenden Visualisierung äußert sich dies durch ein Verlaufen von Strukturgrenzen. Abbildung 3-24 verdeutlicht diesen Zusammenhang.



(a) ein Abtastpunkt in der Mitte jedes Pixels, harte Kanten in der LUT

(b) vier Abtastpunkt an den Ecken jedes Pixels, Kanten der LUT werden geglättet

(c) 25 uniform verteilte Abtastpunkt in jedem Pixel, Kanten der LUT werden ebenfalls geglättet

Abbildung 3-24: Kanten in 2D-LUTs bei unterschiedlichen Abtastraten und die resultierenden Volumenvisualisierungen am Beispiel von drei Kortexschichten (MRT). In der unteren Reihe der Abbildung werden Ausschnitte von LUTs dargestellt, die mit unterschiedlichen Abtastraten erzeugt wurden. In Abbildung (a) sind die Kanten scharf, in Abbildung (b) und (c) geglättet. Die aus den LUTs (b) und (c) erzeugten Visualisierungen (oberhalb) weisen weiche Übergänge an Strukturgrenzen auf.

Die Performanz bei der LUT-Berechnung ist gut (liegt im interaktionsfähigen Bereich), da bilinear zwischen Stützstellen der TF (Knoten des Gittermodells) interpoliert wird. Die Anzahl der Interpolationen wird dabei von der Pixelanzahl der zu erzeugenden LUT bestimmt. Für jedes Pixel wird das Gittermodell ein bzw. mehrmals abgetastet. Für die LUT-Berechnung lassen sich verschiedene Beschleunigungsverfahren implementieren. Hierbei kann die einfache Struktur von 2D-LUTs ausgenutzt werden. 2D-LUTs enthalten viele rechteckige homogene Bereiche, in denen es nicht erforderlich ist, jedes Pixel abzutasten. Vielmehr genügt es diese Bereiche einmal abzutasten und mit der ermittelten Intensität / Farbe zu füllen. Die entscheidende Aufgabe liegt hier in einer Zerlegung der LUTs in möglichst wenig homogene Bereiche. Dies kann durch eine Auswertung des Gittermodells erfolgen. Durch den Einsatz derartiger Beschleunigungen lässt sich der Berechnungsaufwand, in Abhängigkeit von der Anzahl und Größe der homogenen Bereiche in der LUT, um über 90% reduzieren.

3.5 Zusammenfassung

Ziel dieses Kapitels ist es, Repräsentationen für 2D-TFs bereitzustellen, die sich für die interaktive Definition von DTFs eignen. Zu diesem Zweck wurde zunächst eine Evaluierung von Repräsentationen für 1D-TFs vorgenommen (Abschnitt 3.1). Dabei stellte sich die Repräsentation der ungewichteten stückweise linearen Funktionen, die mittels „Peaks“ definiert wird, als besonders geeignet heraus. Im Anschluss wurde auf Basis dieses Ansatzes eine Repräsentation für 2D-TFs, die 2D-Komponetenfunktionen abgeleitet (Abschnitt 3.2.1). Darüber hin-

aus wurde eine zweite Repräsentationsform, die auf der Interpolation zwischen 1D-TFs basiert, vorgestellt (Abschnitt 3.2.2). Für beide Repräsentationsformen wurden geeignete Interaktionskonzepte sowie Anregungen für deren Implementierung und deren Einsatz diskutiert. Das gesetzte Ziel, geeignete Repräsentationen für die interaktive Definition von DTFs bereitzustellen, konnte erreicht werden.

Im nächsten Kapitel wird ein Entwurf von DTFs diskutiert. Dabei kommen die entwickelten Repräsentationsformen zum Einsatz. In diesem Zusammenhang wird näher auf die Unterstützungsmöglichkeiten bei der Definition der Repräsentationsparameter eingegangen.

4 Entwurf distanzabhängiger Transferfunktionen

Ein Großteil der Methoden für die Definition von TFs verfolgen rein intensitätsabhängige Ansätze. Die Definitionsmöglichkeiten sind hierbei sehr eingeschränkt. Durch die zusätzliche Verwendung abgeleiteter Bildeigenschaften wird ein höhere Flexibilität erreicht. In Abschnitt 2.5.4 wurde auf einige Arbeiten eingegangen, die sich mit mehrdimensionalen TFs beschäftigen. Im Mittelpunkt dieser Arbeiten steht häufig die Darstellung von Kanten. Über ihre Kanten lassen sich Strukturen in Datensätzen gut visualisieren. Ein neuer Ansatz beruht auf einer distanzabhängigen Definition von TFs (DTFs). Für die Definition wird hierfür zusätzlich zur Intensitätsdimension eine zweite Dimension verwendet, die in Abhängigkeit von der Distanz zu einer Referenzfläche im Datensatz definiert wird. Diese Referenzflächen können von Gewebeoberflächen gebildet oder auf andere Weise aus der vorliegenden Anatomie abgeleitet werden. Ziel ist es, eine distanzabhängige Steuerung von TFs zu erreichen. Mögliche Anwendungen liegen beispielsweise in Schichtdarstellungen von Strukturen in Bezug zu ihren Oberflächen. Außerdem könnte mit einer distanzabhängigen Färbung das Abschätzen von Abständen zwischen Strukturen erleichtert werden. Darüber hinaus ließe sich mit Hilfe von distanzabhängigen Opazitäten ein gezieltes Ausblenden von Strukturen erreichen, die in der Visualisierung nicht benötigt werden oder Strukturen von höherem Interesse verdecken.

Dieses Kapitel ist wie folgt gegliedert. Zu Beginn werden zwei verwandte Arbeiten vorgestellt. Außerdem werden einige Grundlagen distanzabhängiger TFs vermittelt. Eine zentrale Rolle spielt dabei die Generierung von DistanceMaps und in diesem Zusammenhang das Segmentieren bzw. Erzeugen geeigneter Referenzflächen, beides wird diskutiert. Im Anschluss wird die Definition von DTFs unter Anwendung der vorgestellten Repräsentationsformen (2D-Komponentenfunktionen und Interpolation zwischen 1D-TFs) erläutert. Darüber hinaus wird auf den Renderingprozess unter Verwendung von 3D-Texturen und DTFs, auf Abhängigkeiten der Bildqualität sowie auf die Verwendung von Presets eingegangen.

4.1 Verwandte Arbeiten

In [39] wurden bereits Untersuchungen zu DTFs vorgenommen. Zusätzlich zur Intensitätsdimension wird eine zweite Datendimension eingesetzt und in Abhängigkeit von der Distanz der Voxel zu einem festgelegten Punkt (kann vom Anwender angegeben werden) definiert. Die so bereitgestellten Distanzinformationen werden für die Definition von TFs genutzt. Damit besteht die Möglichkeit, Strukturen in Abhängigkeit vom Abstand zu diesem Punkt zu färben oder auszublenden. Als Anwendungen hierfür werden beispielsweise das lokale Entfernen unerwünschter Strukturen sowie das Clipping von Volumen beschrieben. Die Anwendungsmöglichkeiten sind dabei durch den Umstand begrenzt, dass ausschließlich der Distanzbezug zu einem konkreten Punkt berücksichtigt wird.

Ein weiterer distanzabhängiger Ansatz wird in [51] vorgestellt. Hierbei wird die Distanz zur virtuellen Kamera (Betrachterposition) ausgewertet. TFs werden in Abhängigkeit von diesem Abstand definiert. Als Parameter für die Definition dienen drei Distanzwerte, mit deren Hilfe zwei Distanzintervalle festgelegt werden, die als S-Band und T-Band bezeichnet werden. Das S-Band definiert ein Distanzintervall, indem alle Voxel transparent dargestellt werden. Allen Voxeln, deren Distanz innerhalb des T-Bands liegt, wird mit steigenden Distanz eine steigende Opazität zugeordnet (Abbildung 4-1).

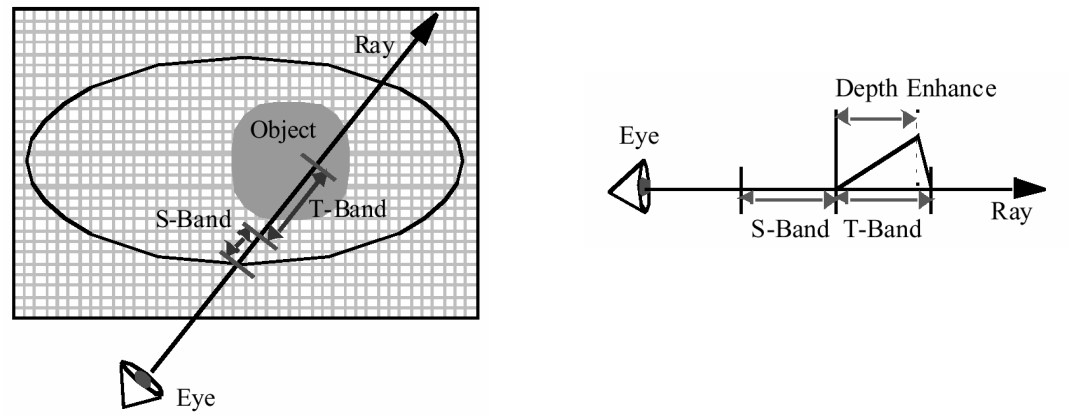


Abbildung 4-1: Confocal Volume Rendering. Quelle [51]

Mit Hilfe dieses Verfahrens können Strukturen in Abhängigkeit ihres Abstandes zum Befrachter dargestellt werden. Möglichen Anwendungen hierfür liegen beispielsweise in einer Unterstützung bei der Exploration von Datensätzen.

4.2 Grundlagen distanzabhängiger TFs

Für die Anwendung von DTFs ist es zunächst erforderlich, Referenzflächen zu definieren. Diese können durch Untermengen der Voxel eines Datensatzes gebildet werden (Segmentierung). Es ließen sich auch komplexere Funktionen einsetzen, die die Voxel eines Datensatzes in Abhängigkeit von ihren Koordinaten Referenzflächen zuordnen. Als Referenzflächen können außerdem geometrische Formen wie Ebenen, Punkte oder Würfel verwendet werden. Im nächsten Schritt werden die euklidischen Distanzen aller Voxel zu der Referenzfläche ermittelt. Diese können in einem zweiten Datensatz, der als DistanceMap bezeichnet wird, abgelegt und später vom Volumenrenderer verwendet werden. Die Distanzberechnung kann außerdem auch auf Basis von Referenzflächenmasken direkt durch den Volumenrenderer erfolgen. Die Masken haben die Größe des Datensatzes und legen für jedes Voxel seine Zugehörigkeit zur Referenzfläche fest. Um sie interpretieren zu können, ist eine Anpassung des verwendeten Renderers erforderlich. Die Distanzberechnung kann auf dieselbe Weise erfolgen, wie es für DistanceMaps in Abschnitt 4.3 beschrieben wird. Problematisch ist hierbei, dass die Berechnung für jede Visualisierung neu ausgeführt werden muss. Bei einem externen Zuführen von DistanceMaps ist die Distanzberechnung nur einmal erforderlich, da die DistanceMaps gespeichert und für spätere Visualisierungen wieder verwendet werden können. Über die Bereitstellung von DistanceMaps hinaus ist es notwendig, DTFs festzulegen, die beschreiben, wie sich Intensität und Distanz von Voxeln auf deren Grauwert bzw. Farbe und Opazität auswirken. DTFs definieren eine intensitäts- und distanzabhängige Farb- und Opazitätsfunktion für den Visualisierungsprozess. Sie sind zweidimensional und können mit den in Kapitel 3 entwickelten Repräsentationsmodellen erzeugt werden.

4.2.1 Distanzabhängige Steuerung der Farbfunktion

Im nachfolgenden Beispiel (Abbildung 4-2) wurde die erste Schicht des Datensatzes als Referenzfläche definiert. Die Distanzen werden in Bezug zu dieser Referenzfläche berechnet, sind also direkt auf der Referenzfläche 0 und steigen entlang der z-Achse des Datensatzes linear an. Darüber hinaus wurde eine DTF definiert, die im Bereich der Knochen, also von etwa 0 bis 1500 HU, einen kontinuierlichen Farbverlauf entlang der Distanzachse sowie opake Strukturen realisiert.

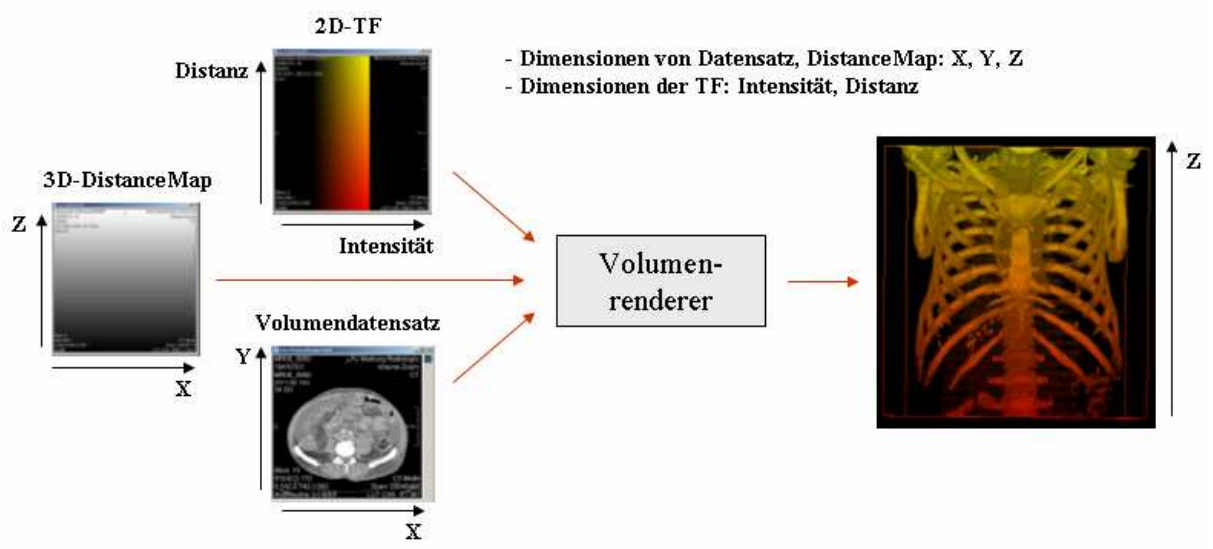


Abbildung 4-2: Beispiel für die Anwendung von DTFs mit variablem Farbverlauf.

Im resultierenden DVR ist sehr gut die variierende Färbung der Knochen entlang der z-Achse zu erkennen. Da die TF für Intensitäten außerhalb des [0HU, 1500HU] Intervalls mit einer Opazität von 0 definiert wurde, werden alle nicht-knöchernen Gewebe ausgeblendet.

4.2.2 Distanzabhängige Steuerung der Opazitätsfunktion

Ein zweites Beispiel verwendet dieselbe DistanceMap jedoch eine andere DTF. Diese DTF hat ebenfalls außerhalb des Intensitätsintervalls [0HU, 1500HU] eine Opazität von 0. Außerdem wurden innerhalb des Intensitätsintervalls variierende Opazitäten festgelegt. Voxel die sich in der Nähe der Referenzfläche befinden werden opak dargestellt. Danach folgen drei Distanzintervalle für die abwechselnd vollständig transparente / opake Voxel definiert wurden. Anschließend wurde ein mit steigender Distanz kontinuierlich zunehmender Opazitätsverlauf realisiert.

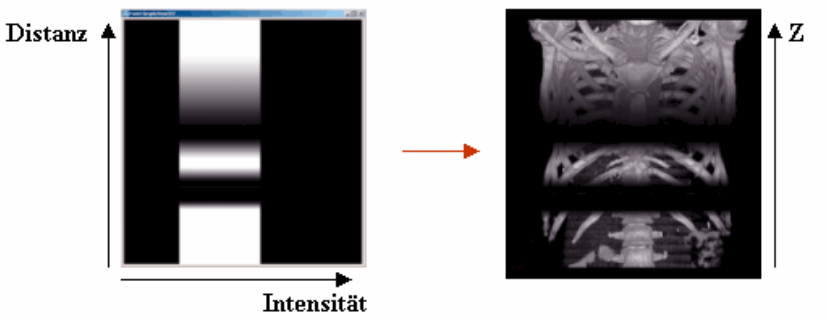


Abbildung 4-3: Beispiel für die Anwendung DTFs mit variablem Opazitätsverlauf.

4.3 Erzeugen von DistanceMaps

Damit DTFs angewendet werden können, müssen Distanzinformationen bereitgestellt werden. Dies kann in Form von DistanceMaps geschehen. DistanceMaps sind Volumendatensätze, in

denen Distanzen von Voxeln zu Referenzflächen abgelegt sind. Sie lassen sich durch die Anwendung von Distanztransformationen auf Referenzflächen berechnen, bei denen jedem Voxel sein euklidischer Abstand zur Referenzfläche zugeordnet wird. Als Referenzflächen werden in diesem Zusammenhang anatomische Flächen im Datensatz bzw. abgeleitete oder frei gewählte Flächen bezeichnet. Deren Bereitstellung wird in Abschnitt 4.4 diskutiert. Für die Erzeugung von DistanceMaps sind Distanztransformationen auf den Referenzflächen anzuwenden. Dabei wird jedem Voxel des Datensatzes sein Abstand zur Referenzfläche zugeordnet. Der euklidische Abstand $D(p)$ eines Voxels p zur Referenzfläche RF wird definiert durch:

$$D(p) = \min\left\{ \sqrt{(p_x - r_x)^2 + (p_y - r_y)^2 + (p_z - r_z)^2} \right\} \qquad \textbf{(4-1)}$$

$$r \in RF\ ,\ D(r) = 0$$

Eine exakte Berechnung der euklidischen Distanz ist aufwändig. Ein solches Verfahren wird in Abschnitt 4.3.3 vorgestellt. Darüber hinaus existieren Verfahren die die euklidische Distanz approximieren. Diese arbeiten schneller aber auch ungenauer. Zwei dieser Verfahren werden in Abschnitt 4.3.1 und 4.3.2 beschrieben.

4.3.1 Chamfer-Metrik

Eine sehr einfache Methode für die Berechnung von Distanzen ist die Chamfer-Transformation. Die Distanz zwischen zwei Pixeln wird hierbei durch die Schritte (im Pixelraster) bestimmt, die benötigt werden um von einem Pixel zum anderen zu gelangen. Dabei wird zwischen horizontalen, vertikalen und diagonalen Schritten unterschieden.

Die Distanz zwischen zwei Pixeln ist bestimmt durch:

$$D = m_1 + \sqrt{2} * m_2 \qquad \textbf{(4-2)}$$

wobei m_1 die Summe der horizontalen und vertikalen Schritte und m_2 die Anzahl der diagonalen Schritte angibt. Horizontalen und vertikalen Schritten wird also eine Länge von 1, diagonalen eine Länge von $\sqrt{2}$ zugewiesen. In Abbildung 4-4 ist das Ergebnis einer Chamfer-Transformation auf einem 2D-Bild (ein Vordergrundpixel) dargestellt.

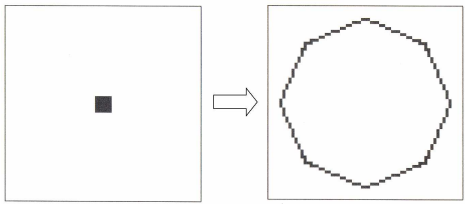


Abbildung 4-4: 2D-Chamfer-Tansformation. Quelle: [52]

Das Verfahren lässt sich leicht dahingehend anpassen, dass ganzzahlige Distanzen berechnet werden, wodurch der Speicherbedarf für Distanzbilder gering gehalten werden kann. Diese Metrik ist sehr ungenau und deshalb für die Berechnung von DistanceMaps, wie sie zur Anwendung von DTFs benötigt werden, ungeeignet.

4.3.2 Distanzberechnung mit Hilfe von lokalen Distanzen

Ein weiter Algorithmus für die Berechnung von Distanzen wird in [10] beschrieben. Dieser Algorithmus bestimmt Distanztransformationen durch die Auswertung von lokalen Distanzen. Für die Berechnung der lokalen Distanzen können verschiedne Kernel eingesetzt werden. Diese werden im Folgenden beschrieben.

City-Block-Abstand

Der City-Block-Abstand approximiert den euklidischen Abstand am schlechtesten. Die Berechnung ist aufgrund der geringen Anzahl an benötigten Operationen sehr schnell. Die lokale Distanz wird lediglich für eine 6'er-Nachbarschaft ermittelt.

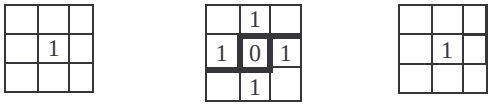


Abbildung 4-5: City-Block-Abstand.

Schachbrett-Abstand

Der Schachbrett-Abstand stellt eine Erweiterung des City-Block-Abstands dar. Anstatt der 6'er Nachbarschaft wird ein 18'er-Nachbarschaft verwendet.

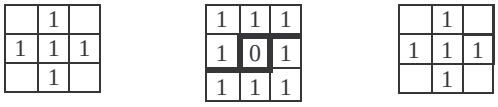


Abbildung 4-6: Schachbrett-Abstand.

Euklidischer-Abstand (3×3×3-Matrix)

Deutlich bessere Ergebnisse sind zu erzielen, indem den diagonal angeordneten Voxeln ihre tatsächliche euklidische Distanz zugeordnet wird. Eine sehr einfache Variante unterscheidet sich vom Schachbrett-Abstand lediglich darin, dass den diagonalen Voxeln Distanzen von $\sqrt{2}$ zugeordnet werden.

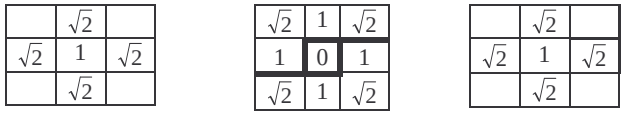


Abbildung 4-7: Euklidischer-Abstand (3×3×3).

Euklidischer-Abstand (5×5×5-Matrix)

Die Genauigkeit der euklidischen Approximation lässt sich durch eine Vergrößerung des verwendeten Kernels weiter erhöhen. Zu beachten ist jedoch der steigende Berechnungsaufwand (liegt bei 5×5×5-Kernel bereits um Faktor 10 höher als bei 3×3×3-Kernel).

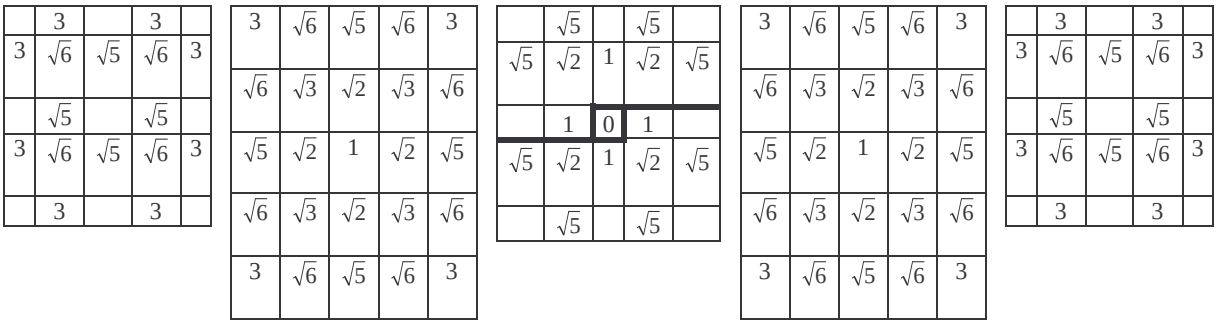


Abbildung 4-8: Euklidischer-Abstand (5×5×5).

Vor der Anwendung eines dieser Kernel wird zunächst das Distanzvolumen mit maximalen Distanzen initialisiert. Alle Voxel auf der Referenzfläche erhalten eine Distanz von 0. Die Distanzberechnung läuft in zwei Schritten ab. Zunächst wird das Volumen von vorne nach hinten durchlaufen. Dabei werden nur die Felder des Kernels angewendet, die sich oberhalb und links neben dem zu berechnenden Voxel befinden (in der Kerneldarstellung durch eine Linie von den anderen getrennt). Die Werte des Kernels werden mit den Distanzen der überlagerten Voxel addiert. Die resultierende Distanz ergibt sich aus dem Minimum aus dem vorherigem Voxelwert und der kleinsten Summe. Anschließend wird das Distanzvolumen rückwärts durchlaufen. Dabei werden alle Felder des Kernels angewendet, die unterhalb und rechts neben dem zu berechnenden Voxel liegen. Auf diese Weise kann durch die Berechnung lokaler Distanzen die gesamte DistanceMap ermittelt werden.

4.3.3 Exakte Berechnung der euklidischen Distanz

Darüber hinaus ist eine exakte Berechnung der euklidischen Distanz möglich. Ein Verfahren hierfür wird in [52] vorgestellt. Es läuft in drei Schritten ab, bei denen jeweils eine Dimension des Datensatzes ausgewertet wird.

Im ersten Schritt wird das dreidimensionale binäre Eingansbild $v_{b,r,c}$ (bestehend aus Vordergrund - und Hintergrundvoxeln) mit den Ausdehnungen n_b, n_r, n_c Pixel für Pixel in ein Zwischenbild $f_{b,r,c}$ transformiert. Hierbei wird der Wert jedes Hintergrundvoxels durch seinen minimalen quadratischen Abstand zum nächstgelegenen Vordergrundvoxel ersetzt.

$$\left\{v_{b,r,c} \in \{0,1\} \mid 0 \le b < n_b, 0 \le r < n_r, 0 \le c < n_c\right\}$$

$$f_{b,r,c} = \min\left\{(c-j)^2, v_{b,r,j} = 0,\ 0 \le j \le n_c\right\} \qquad \textbf{(4-3)}$$

Der zweite Schritt läuft ähnlich ab. Dabei wird eine andere Dimension des Datensatzes betrachtet.

$$g_{b,r,c} = \min\left\{f_{b,i,c} + (r-i)^2 \mid 0 \le i \le n_r\right\} \qquad \textbf{(4-4)}$$

Im letzten Schritt wird analog zum zweiten die dritte Dimension ausgewertet.

$$h_{b,r,c} = \min\left\{g_{k,r,c} + (b-k)^2 \mid 0 \le k \le n_b\right\} \qquad \textbf{(4-5)}$$

Das Ergebnisbild $h_{b,r,c}$, repräsentiert die exakte euklidische Distanz zu den Vordergrundvoxeln des Eingangsbildes $v_{b,r,c}$. Für die Berechung von DistanceMaps wurde auf diesen Algorithmus zurückgegriffen, da Approximationen der euklidischen Distanz durch Chamfer-Transformation oder lokale Distanzen keine genauen Ergebnisse liefern.

Erzeugen von positiven und negativen Distanzwerten

Falls die Referenzflächen Volumen einschließen, wie es beispielsweise bei Organoberflächen geschieht, ist es nicht möglich, zwischen Distanzen innerhalb des eingeschlossenen Volumens und Distanzwerten außerhalb zu unterscheiden. Darum ist es sinnvoll, die DistanceMap nach ihrer Berechnung mit einer Maske des segmentieren Volumens zu multiplizieren. Alle Voxel, die sich außerhalb des eingeschlossenen Volumens befinden, erhalten dabei ein negatives Vorzeichen. Je nach Lage der Voxel zur Referenzfläche sind also sowohl positive als auch negative Distanzen möglich.

Abbilden von Distanzen auf ganzzahlige Werte

Die Distanzen werden zunächst in Float-Genauigkeit berechnet. Um eine gute Performanz bei der Visualisierung zu gewährleisten, erfolgt eine Abbildung der Werte auf Signed Integer. Auf diese Weise kann eine Halbierung des anfallenden Datenvolumens erreicht werden (für die Speicherung eins Float-Wertes sind üblicherweise 4Byte, für einen Int16 nur 2 erforderlich). Damit eine Genauigkeit bis zur zweiten Nachkommastelle gewährleistet werden kann, sind die Distanzwerte vor ihrer Konvertierung mit 100 zu multipliziert. Auf diese Weise können Distanzen von -327,68 bis +327,68 mm mit einer Genauigkeit von 0.01 mm durch Integer-Werte aufgelöst werden. Für die in Kapitel 6 vorgestellten Anwendung ist dies ausreichend.

4.4 Erzeugen von Referenzflächen am Beispiel medizinischer Datensätze

Als Referenzflächen werden Untermengen der Voxel eines Volumendatensatzes bezeichnet, die zur Distanzberechnung herangezogen werden. Sie lassen sich nach der Art ihrer Generierung in drei Gruppen einteilen:

- anatomische Referenzflächen z.B.: Organoberflächen, Schädel
- aus Anatomie abgeleitete Referenzflächen z.B.: Fläche über Rippen aufgespannt
- von der Anatomie unabhängige Referenzflächen z.B.: Ebene, Kugel, Zylinder etc.

Für jede dieser Gruppen sind andere Berechnungsstrategien erforderlich.

4.4.1 Ermittlung anatomischer Referenzflächen

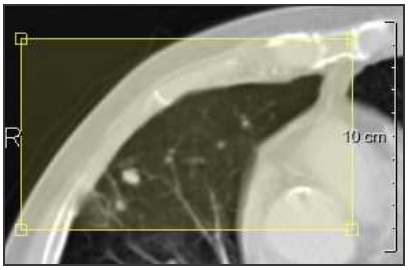
Bei der Ermittlung anatomischer Referenzflächen besteht das Ziel darin, Subvolumina der Datensätze, wie beispielsweise einzelne Organe, zu segmentieren und ihre Konturen zu berechnen. Derartige Konturen eignen sich sehr gut als Referenzflächen. Sie lassen sich durch eine Verknüpfung von Bildverarbeitungsalgorithmen bestimmen. Je nach Art der vorliegenden Gewebestrukturen können hierfür verschiedene Algorithmen sinnvoll sein. Als nützlich haben sich in diesem Zusammenhang RegionGrowing, Wasserscheidentransformation, morphologische Operatoren sowie verschiedene Filter erwiesen [22]. Im Folgenden werden Berechnungsmöglichkeiten von Referenzflächen am Beispiel der Lungenoberfläche, Kortexoberfläche sowie des Rückenmarks vorgestellt.

Lungenoberfläche (basierend auf CT-Aufnahmen)

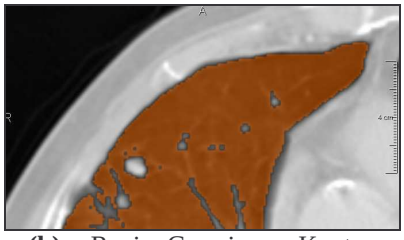
Um die Lungenoberfläche zu ermitteln, sind mehrere Schritte erforderlich:

- *Bildausschnitt anpassen*: Zunächst sollte der verwendete Datensatz auf die interessierenden Bereiche reduziert werden. Dies kann geschehen, indem er an Ebenen parallel der Koordinatenebenen geclippt wird. Auf diese Weise lässt sich die Anzahl der enthaltenen Voxel oft erheblich reduzieren. Dies ist im Hinblick auf ein späteres Volumenrendering aus Performanzgründen sehr sinnvoll (Abbildung 4-9, a).
- *Grobe Segmentierung der Lunge*: Um die Oberfläche der Lunge zu berechnen, muss die Lunge zunächst segmentiert werden. Dies ist in grober Form mit RegionGrowing-Algorithmen möglich. Hierfür wird ein Startvoxel innerhalb der Lunge sowie ein Schwellwert, der der Intensität der Lungenoberfläche entspricht, festgelegt. Ausgehend vom Startvoxel werden nun alle Voxel, deren Intensität unterhalb des Schwellwertes liegt, dem segmentierten Volumen hinzugefügt. Da sich die Intensität innerhalb der Lunge (Luft, Blutgefäße) recht stark von der außerhalb (Muskel, Knochen, andere Organe) unterscheidet, ist so eine gute Segmentierung möglich. Probleme treten in Regionen auf, die eine hohe Intensität aufweisen, aber trotzdem segmentiert werden sollen. Dies betrifft beispielsweise Blutgefäße mit großem Durchmesser, Bronchien aber auch Tumore. Sie erzeugen Löcher in der Segmentierung, die geschlossen werden müssen (Abbildung 4-9, b).
- *Schließen von Löchern*: Durch den Einsatz morphologischer Operatoren können die meisten dieser Löcher geschlossen werden. Mit einem Closing (Dilatation gefolgt von einer Erosion) lassen sich gute Ergebnisse erzielen. Durch die Dilatation wächst das segmentierte Volumen etwas an. Dabei werden kleinere Löcher geschlossen. Die anschießende Erosion reduziert das Volumen wieder annähernd auf seine Ausgangsgröße. Kernelgrößen von 3×3×3 und 5×5×5 stellen dabei einen guten Kompromiss zwischen hoher Performanz und einem vollständigem Schließen der Löcher dar (Abbildung 4-9, c).
- *Ermitteln der Kontur (Referenzfläche)*: Danach wird die Kontur des segmentierten Volumens bestimmt. Sie wird von den Voxeln gebildet, die sowohl an Voxel innerhalb als auch an Voxel außerhalb des segmentierten Volumens angrenzen. Diese Kontur ist eine Approximation der Lungenoberfläche und kann als Referenzfläche verwendet werden (Abbildung 4-9, d).

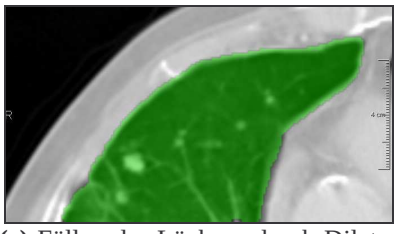
Die folgende Abbildung stellt die einzelnen Schritte graphisch dar:



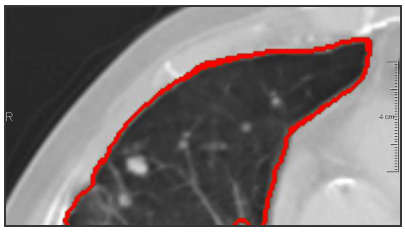
(a) Zuschneiden des Datensatzes.



(b) RegionGrowing, Knoten sowie einige Gefäße werden nicht segmentiert.



(c) Füllen der Löchern durch Dilatation und Erosion. (Kernelgröße = 3×3×3)



(d) Ermitteln der Kontur (3D).

Abbildung 4-9: Berechnung einer Referenzfläche basierend auf einem Teil des linken Lungenflügels.

Das Ergebnis ist eine Maske, die die Voxel auf der Oberfläche der Lunge kennzeichnet. Diese kann als Grundlage für eine Distanzberechnung dienen.

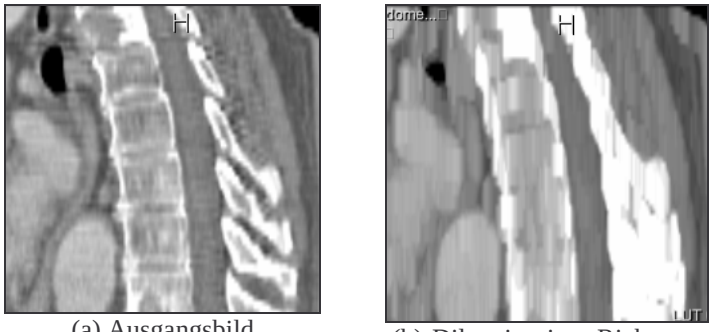
Kortexoberfläche (basierend auf MRT-Aufnahmen)

Die Berechnung der Kortexoberfläche ähnelt in vielen Punkten der Berechnung der Lungenoberfläche. Allein das verwendete Segmentierungsverfahren ist ein anderes. Zu Beginn wird der Datensatz auf den interessierenden Bereich zugeschnitten. Anschließend ist der Kortex zu segmentieren. Hiefür ist ein RegionGrowing ungeeignet, da es an vielen Bereichen, beispielsweise am Ansatz des Rückenmarks, auslaufen würde. Außerdem ist das Verfahren nur bedingt für MRT-Aufnahmen geeignet. Die für diese Aufnahmen typischen Bildinhomogenitäten wirken sich negativ auf das Regionenwachstum aus. Durch sie ist es unmöglich, mit einem einheitlichen Schwellwert den Kortex zu segmentieren. Wesentlich besser geeignet ist die Wasserscheidentransformation [23]. Dabei werden Intensitätssenken im Gradientenbild betrachtet, die „aufgefüllt“, also dem segmentierten Volumen hinzugefügt werden können. Dies kann interaktiv geschehen. Um gute Ergebnisse zu erzielen ist ein recht großer Interaktionsaufwand erforderlich. Dieser hängt stark von der Qualität des Bildmaterials sowie den verwendeten Aufnahmeparametern ab. Aus einer T1-gewichteten MRT-Aufnahme lässt sich der Kortex in akzeptabler Genauigkeit in etwa 1-5 min segmentieren. Besonders gute Ergebnisse sind mit diesem Verfahren bei T1-gewichteten MRT-Aufnahmen zu erzielen. Das so segmentierte Volumen kann Unebenheiten auf der Oberfläche aufweisen. Diese sind vorwiegend auf die bereits angesprochenen Bildinhomogenitäten zurückzuführen. Eine Glättung kann wiederum durch Closing erfolgen. Im Anschluss ist die Kontur des segmentierten Volumens zu bestimmen. War die Segmentierung hinreichend genau, approximiert die berechnete Kontur die Kortexoberfläche sehr gut.

Abbildung 4-10: Segmentierung der Kortexoberfläche aus MRT-Aufnahmen.

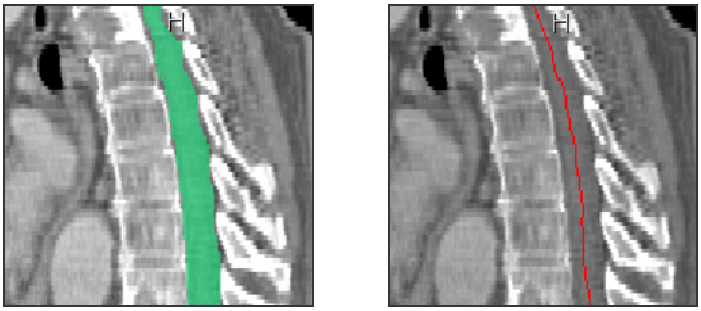
Rückenmark (basierend auf CT-Aufnahmen)

Für die Segmentierung des Rückenmarks ist wiederum RegionGrowing gut geeignet. Als Startvoxel ist hierbei ein Voxel des Rückenmarks auszuwählen. Der Schwellwert sollte auf die Intensität der Bandscheiben gesetzt werden. Dabei ist zu beobachten, dass es zwischen den Wirbeln zum Auslaufen des selektierten Volumens kommen kann. Dies lässt sich durch eine „Verschmierung“ des Datensatzes in Richtung der Wirbelsäule verhindern. Hierfür eignet sich beispielsweise eine gerichtete Dilatation (Kernel: 1×1×7). Auf diese Weise kann ein Verschmelzen der einzelnen Wirbelkörper herbeigeführt und ein Auslaufen verhindert werden. Dabei ist darauf zu achten, dass der Rückenmarkskanal erhalten bleibt. Auf dem so selektierten Rückenmark ist anschließend eine Skelettierung durchzuführen. Dabei wird das segmentierte Volumen auf seine Mittellinie reduziert. Wenn die Segmentierung starke Unebenheiten auf der Oberfläche aufweist (besonderst schlecht wirken sich teilweise von restlichen Volumen abgetrennte Bereiche aus), kann es zur Bildung von Seitenästen am Skelett kommen. Der Einfluss dieser Seitenäste auf das Ergebnis einer späteren Distanzberechnung ist nur gering, weil sich die Äste auf Grund der umgebenden Wirbel nicht weit von der Mittellinie des Rückenmarks entfernen können, dennoch sollten sie vermieden werden. Durch Closing oder Glättung des segmentierten Rückenmarks kann die Bildung von Seitenästen bei der Skelettierung nahezu vollständig verhindert werden. Die Parameter der Skelettierung, wie maximale Länge des Skeletts oder Toleranzen bei der Berechnung, können einmal eingestellt für beliebige Datensätze beibehalten werden. Die Angabe von Start und Endpunkt des zu erzeugenden Skelettes ist nicht erforderlich. Die auf diese Weise berechnete Mittellinie approximiert den Verlauf des Rückenmarks sehr gut. Sie ist deshalb als Ausgangspunkt einer Distanzberechnung interessant.



(a) Ausgangsbild

(b) Dilatation in z-Richtung, Kernel = [1,1,7]



(c) Segmentierung mit RegionGrowing, Intensität StartVoxel = 46HU, Schwelle ± 3%

(d) Skelettierung

Abbildung 4-11: Berechnung einer Mittellinie des Rückenmarkkanals.

4.4.2 Modellierung anatomisch abgeleiteter Referenzflächen

Das Bereitstellen anatomisch abgeleiteter Referenzflächen ist recht aufwändig. Im Mittelpunkt steht hier die Generierung eines Modells der Referenzfläche auf Basis anatomischer Gegebenheiten. Derartige Referenzflächen können gut durch polygonale Netze oder Patches [5] repräsentiert werden. Beide Modelle gewährleisten eine hohe Flexibilität beim Formen von Referenzflächen. Außerdem ließen sich deren Gitterabstände adaptiv anpassen. Einfache Referenzflächen könnten bereits mit Netzen von 4^2 oder 5^2 - Kontrollpunkten definiert werden. Die Positionierung der Kontrollpunkte wird aus der vorliegenden Anatomie abgeleitet. Mögliche Anwendungen sind beispielsweise das Aufspannen von Referenzflächen durch die Rippen oder die Handknochen. Voraussetzung ist eine Segmentierung der interessierenden Strukturen. Hierfür können verschiedene Methoden angewendet werden. Für die Segmentierung von Knochen eignen sich z.B. spezielle Bone-Removal-Verfahren [24].

4.4.3 Erzeugen anatomisch unabhängiger Referenzflächen

Anatomisch unabhängige Referenzflächen können beispielsweise mit 3D-Zeichentools erstellt, oder durch Funktionen angenähert werden. Die Anzahl der möglichen Formen ist nahezu unbegrenzt. Als sinnvoll könnten sich Ebenen, Kugeln, Quader oder auch Punkte erweisen. Durch Überlagerung von Volumendatensätzen mit diesen Geometrien können Masken der getroffenen Voxel erzeugt werden. Diese Masken lassen sich als Ausgangpunkt für Distanzberechnungen nutzten.

4.4.4 Zusammenfassung

Je nach Visualisierungsziel können unterschiedliche Referenzflächen zu Einsatz kommen, für die wiederum verschiedene Berechnungsstrategien erforderlich sind. Für die Ermittlung anatomischer Referenzflächen ist die Segmentierung von Strukturen im Datensatz erforderlich. Hierfür können Segmentierungsverfahren wie RegionGrowing und Wasserscheidentransformation eingesetzt werden. Die Bereitstellung anatomisch abgeleiteter Referenzflächen gestaltet sich wesentlich aufwändiger, da Modelle der Referenzflächen aus den Datensätzen abzuleiten sind. Methoden hierfür wurden an dieser Stelle nicht weiter untersucht. Darüber hinaus besteht die Möglichkeit anatomisch unabhängige Referenzflächen einzusetzen. Diese können beispielsweise mit 3D-Zeichentools erzeugt werden.

Durch die Vereinigung ihrer Voxelmengen lassen sich beliebig viele Referenzflächen miteinander kombinieren. Deren Lage untereinander ist dabei unerheblich. Sie können sich durchstoßen, berühren oder getrennt von einander positioniert sein. Für eine spätere Distanzberechnung ist es nicht erforderlich, dass die resultierende Voxelmenge eine zusammenhängende Struktur bildet.

4.5 Definition distanzabhängiger TFs

In diesem Abschnitt wird eine Definition von DTFs anhand der in Abschnitt 3.2.1 und 3.2.2 entwickelten Repräsentationen für 2D-TFs erläutert. Ziel ist es, durch eine geeignete Parametrierung TFs mit den gewünschten Eigenschaften bereitzustellen. Dieser Prozess läuft iterativ ab. Nach jeder Änderung der Interaktionsparameter wird eine Volumenvisualisierung mit der aktuellen TF erzeugt. Die Parameter können auf diese Weise so lange angepasst werden, bis das Visualisierungsergebnis den Erwartungen entspricht. Dabei kann dem Anwender über die Volumenvisualisierung hinaus zusätzliches Feedback zur Verfügung gestellt werden.

Zum Erzeugen einfacher TFs wird teilweise auf datenbasierte Verfahren wie Histogrammanalysen oder Kantendetektion zurückgegriffen [3, 26, 27, 28, 29]. Ein solches Vorgehen ist jedoch für die Definition von DTFs nicht sinnvoll. Die verschiedenen Repräsentationsparameter sind hier derart spezifisch, dass ihre Einstellung nicht aus den Bildeigenschaften sondern lediglich aus dem Visualisierungsziel abgeleitet werden kann. Es können beispielsweise aus demselben Datensatz sehr unterschiedliche Visualisierungen erzeugt werde. Aus diesem Grund geschieht die Definition ausschließlich interaktiv.

4.5.1 Definition mit Hilfe von 2D-Komponentenfunktionen

Wie in Abschnitt 3.2.1 beschrieben, können 2D-TFs durch Komponentenfunktionen repräsentiert werden. Die Komponentenfunktionen verfügen über die Parameter:

- Intensitätsintervall,
- Distanzintervall,
- Typ der Komponentenfunktion,
- Amplitude der Komponentenfunktion (Grauwert / Farbe, Opazität).

Deren Einstellung lässt sich wie folgt aus dem Visualisierungsziel ableiten:

Intensitätsintervall: Durch die Auswahl eines Intensitätsintervalls werden die darzustellenden Gewebetypen festgelegt. Für CT-Datensätze ist dies recht einfach, da sie standardisiert sind und so eine Zuordnung von Intensitätsintervallen und Gewebetypen möglich ist. Zu beachten ist dabei, dass sich die Intensitätsintervalle nahezu aller Gewebetypen mit anderen überschneiden (Abbildung 4-12). Eine eindeutige Auswahl eines bestimmten Gewebes ist deshalb nicht möglich. Vielmehr kann jedoch ein Großteil der unerwünschten Gewebe aus der Visualisierung ausgeschlossen werden.

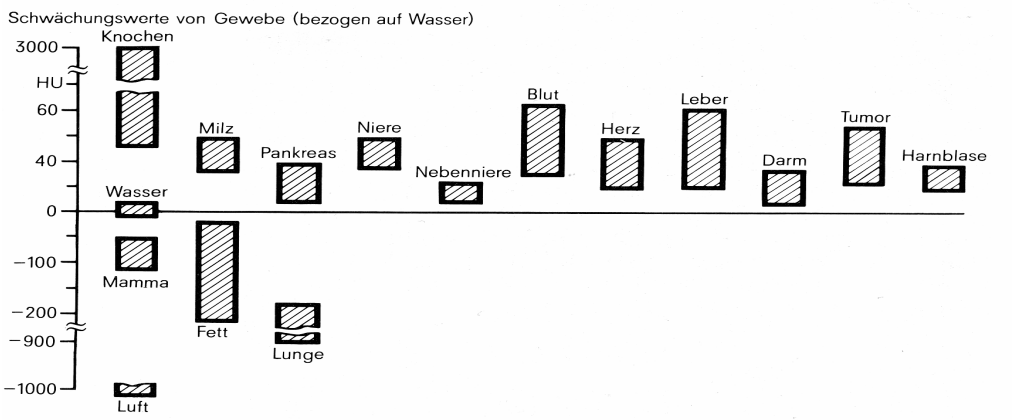


Abbildung 4-12: HU-Intervalle verschiedener Gewebetypen in CT-Aufnahmen. Quelle: [6]

Bei MRT-Aufnahmen sind die Gewebeintensitäten von den Aufnahmeparametern abhängig (Abbildung 2-1, Abbildung 2-2). Eine allgemeine Zuordnung zwischen Bildintensität und Gewebetypen ist deshalb nicht möglich. Die Intensitäten können jedoch leicht Schichtbildern entnommen werden. Zu beachten ist dabei, dass die Gewebeintensitäten in MRT-Aufnahmen Schwankungen unterworfen sind (Abbildung 4-22). Intensitätsintervalle sind dementsprechend groß zu wählen. Eine nützliche Unterstützung bei der Einstellung des Intensitätsintervalls kann durch eine Schichtbilddarstellung des Datensatzes erfolgen, in der alle Voxel, die im aktuellen Intensitätsintervall liegen, eingefärbt werden. Abbildung 4-13 zeigt ein solches Schichtbild.

Distanzintervall: Das Distanzintervall legt fest, innerhalb welcher Distanzen zur Referenzfläche Gewebe dargestellt werden. Die Einstellung dieses Intervalls kann ebenfalls durch eine Hervorhebung der betreffenden Voxel in Schichtbildern unterstützt werden (Abbildung 4-14). Im Allgemeinen ist die Einstellung unproblematisch, da das zugrunde liegende Konzept, Gewebe innerhalb eines bestimmten Abstandes zu einer Fläche zu visualisieren, mental leicht nachvollzogen werden kann.

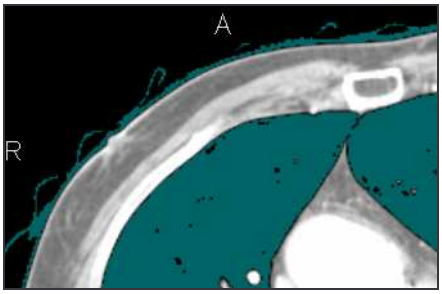


Abbildung 4-13: Darstellung des aktuellen Intensitätsintervalls. In diesem Beispiel wurde ein Intervall von [-980 HU, -420 HU] gewählt. Alle Voxel, deren Intensitäten innerhalb dieses Intervalls liegen, die also von der aktuellen Komponentenfunktion beeinflusst werden, sind mit einer Maske überlagert.

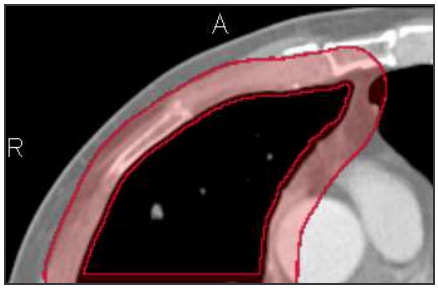


Abbildung 4-14: Darstellung des aktuellen Distanzintervalls. In diesem Beispiel wurde ausgehend von der Lungenoberfläche ein Intervall von [-14 mm, 1 mm] eingestellt. Analog zur linken Abbildung werden alle Voxel deren Distanzen im definierten Intervall liegt mit einer Maske überlagert. Die Grenzen des Intervalls sind durch opake Linien gekennzeichnet.

Eine weitere Möglichkeit für die Festlegung des Intensitäts- und Distanzintervalls besteht in einer Assoziation der Intervalle mit den Koordinaten der Eckpunkte eines Rechteckes. Dieses kann interaktiv mit Hilfe der Maus über einem 2D-Histogramm (Intensität, Distanz) aufgespannt werden (siehe Abschnitt 3.2.1, Vorschläge für die Präsentation der Interaktionsparameter).

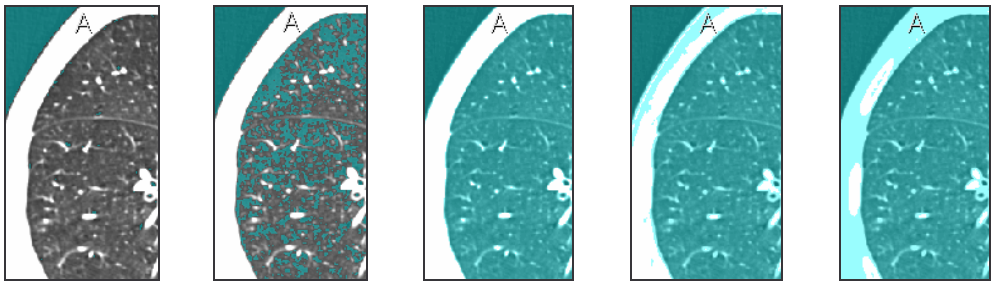
Typ der Komponentenfunktion: Durch die Auswahl des Funktionstyps wird eine Entscheidung darüber getroffen, wie Voxel, deren Intensität und Distanz innerhalb der gewählten Intervalle liegt, dargestellt werden (Farb- und Opazitätsverlauf entlang der Intensitäts- und Distanzachse). Es stehen verschiedene Funktionsgrundtypen zur Auswahl (Abbildung 3-12). Sehr häufig werden in diesem Zusammenhang Funktionen benötigt, die einen linear steigenden bzw. fallenden Farb- / Opazitätsverlauf realisieren. Sie werden auch als Rampen bezeichnet. Ihr Funktionsverlauf kann symbolisch durch (¯_; _/¯)angedeutet werden. Mit ihrer Hilfe ist es beispielsweise möglich, innerhalb eines bestimmten Distanzintervalls eine kontinuierlich steigende oder fallende Opazität zu definieren. Diese Funktionstypen können auch für die Definition von Farbverläufen eingesetzt werden. Um beispielsweise einen steigenden Farbverlauf zu realisieren, werden die Werte des Farbtupels (RGB) kontinuierlich von (0,0,0) auf die gewünschte Farbe z.B. rot (1,0,0) erhöht. Darüber hinaus sind auch Funktionen nützlich, deren Graph sich innerhalb eines bestimmten Bereichs konstant verhält (_|¯|_). Mit ihnen kann eine konstante Farbe / Opazität entlang der Intensitäts- bzw. Distanzachse definiert werden. Darüber hinaus könnten sind weitere Funktionstypen, wie Zeltfunktionen: (_/_) oder Sägezahnfunktionen: (_/|_ ; _|_) als nützlich erweisen. Oft genügt es, identische Funktionstypen für Farb- und Opazitätsverlauf festzulegen. In einigen Fällen kann es jedoch auch sinnvoll sein, unterschiedliche Funktionstypen anzugeben. Dies ist z.B. dann notwendig, wenn ein konstanter Farbverlauf bei steigender Opazität realisiert werden soll.

Amplitude der Komponentenfunktion: Die Angabe einer Amplitude ist sowohl für die Opazitätsfunktion (Alpha-Kanal) als auch für die Farbfunktion (RGB-Kanäle) erforderlich. Damit wird die maximale Opazität bzw. Farbe einer Komponentenfunktion festgelegt. Die Amplitude der Opazitätsfunktion wird durch einen skalaren Wert [0,1], die Farbe durch ein Werte-Tupel ([0,1];[0,1];[0,1]) repräsentiert. Ist z.B. ein maximale Opazität von 0.5 und ein Funktionstyp eingestellt, der konstante Werte definiert, werden alle Gewebe im betreffenden Intervall mit einer Opazität von 0.5 dargestellt. Für die Auswahl von Farben bieten sich Farbpaletten an. Wenn die verwendete Farbpalette RGBA-Farben unterstützt, kann das Festlegen von Farbe und Opazität in einem Schritt erfolgen.

Beispiel für die Definition mit 2D-Komponentenfunktionen

Im Folgenden wird anhand eins Beispiels die Definition von DTFs mit Hilfe von 2D-Komponentenfunktionen beschrieben. Die Aufgabe besteht in der Visualisierung einer 5 mm dicken opaken Schicht innerhalb der Lunge. Diese Schicht soll einen konstanten Abstand von 10 mm von der Lungenoberfläche aufweisen.

Die Aufgabe kann durch die Definition einer Komponentenfunktion gelöst werden. Zunächst ist deren Intensitätsintervall auszuwählen. Da Lungengewebe dargestellt werden soll, ist es so einzustellen, dass die Intensitäten aller Lungenvoxel abgedeckt werden. Ein Feedback kann über die Färbung der im Intervall befindlichen Voxel bereitgestellt werden.



(a) [-1024,-974] **(b)** [-1024,-894] **(c)** [-1024,-424] **(d)** [-1024,6] **(e)** [-1024,106]

Abbildung 4-15: Auswahl eines Intensitätsintervalls. Die Intensitäten sind in HU angegeben.

Weil lediglich eine Schicht innerhalb der Lunge darzustellen ist, haben Gewebe außerhalb der Lunge keinen Einfluss auf die Visualisierung. Ihre Opazität wird durch das Festlegen des Distanzintervalls auf 0 gesetzt. Aus diesem Grund ist es unproblematisch, wenn das Intensitätsintervall in den Bereich von Fettgewebe oder Knochen reicht. Wichtig ist jedoch, dass alle Intensitäten der Lunge abgedeckt werden. Es können also die Intervalle c, d und e aus Abbildung 4-15 verwendet werden. Die Intervalle a und b sind ungeeignet.

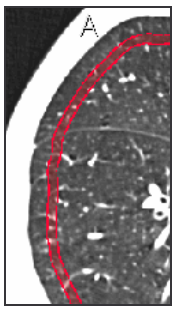


Abbildung 4-16: Darstellung des gewählten Distanzintervalls in Schichtbildern.

Das Festlegen des Distanzintervalls ist sehr einfach. Aus der Aufgabenstellung ergibt sich das Intervall [10 mm, 15 mm]. Eine Kontrolle ist wiederum durch die Färbung der betreffenden Voxel möglich.

Weiterhin sind die Funktionstypen für Opazität und Farbe zu bestimmen. Da die darzustellende Schicht eine einheitliche Opazität erhalten soll, ist eine Opazitätsfunktion festzulegen, die über Intensitäten und Distanzen konstant verläuft (_|‾|_). Für die Farbfunktion gilt das nicht. Um in der Visualisierung Strukturen zu erkennen, ist eine intensitätsabhängige Definition erforderlich. Hierfür eignen sich ansteigende Rampen (_/‾) besonderst gut, da sie, entlang der

Intensitätsachse definiert, geringen Intensitäten dunkle und hohen Intensitäten helle Farbwerte zuordnen und dabei eine kontinuierlichen Farbverlauf realisieren.

Abschließend sind die Amplituden von Farb- und Opazitätsfunktion festzulegen. Um die Strukturen vollständig Opak darzustellen, ist für die Opazitätsfunktion 1 anzugeben. Die Amplitude der Farbfunktion wird durch die Auswahl einer Farbe festgelegt. In diesem Beispiel wird die Farbe verwendet. Durch die ausgewählte Farbrampe (_/¯) ergibt sich entlang der Intensitätsachse innerhalb des gewählten Intervalls die in Abbildung 4-17 dargestellte Farbverteilung. Farbwerte außerhalb des Intensitätsintervalls werden konstant extrapoliert.

Abbildung 4-17: Beispiel einer Farbverteilung entlang der Intensitätswerte.

Abbildung 4-18 zeigt die resultierende 2D-Lut. Sie ist für den gesamten Intensitätsbereich des Ausgangsbildes sowie für alle Distanzen der berechneten DistanceMap definiert. Werte ungleich [0,0,0,0] (RGBA) liegen ausschließlich für Distanzen innerhalb des festgelegten Distanzintervalls vor. Zwischen -1024 und -424HU existiert ein kontinuierlicher Farbverlauf, für Intensitäten >-424HU ist eine konstante Farbe definiert.

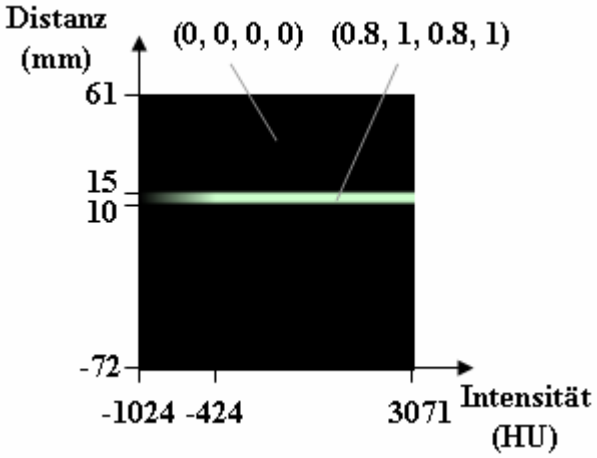


Abbildung 4-18: Darstellung der resultierenden 2D-Lut.

Abbildung 4-19 zeigt Ergebnisse einer Volumenvisualisierung unter Verwendung der erzeugten LUT (Abbildung 4-18). Der Interaktionsaufwand für die Definition der LUT war gering. Es wurden lediglich fünf Parameter eingestellt.

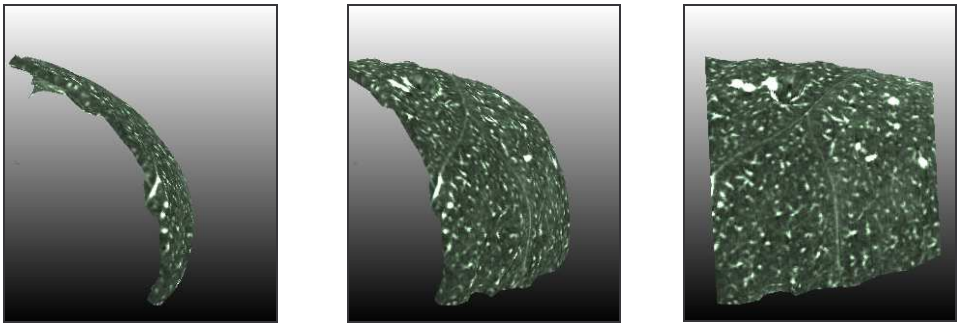


Abbildung 4-19: Volumenvisualisierung unter Verwendung der erzeugten LUT (Abbildung 4-18).

4.5.2 Definition durch Interpolation zwischen 1D-TFs

Darüber hinaus ist die Definition von 2D-TFs durch Interpolation zwischen 1D-TFs möglich (Abschnitt 3.2.2). Ein wesentlicher Unterschied zu 2D-Komponentenfunktionen besteht darin, dass 1D-TFs lediglich intensitätsabhängig für konkrete Distanzen definiert werden. Die 1D-TFs lassen sich hierfür wie folgt parametrisieren:

- Intensitätsintervall
- Distanz
- Funktionstyp (Abbildung 3-5)
- Amplitude (Farbe, Opazität)

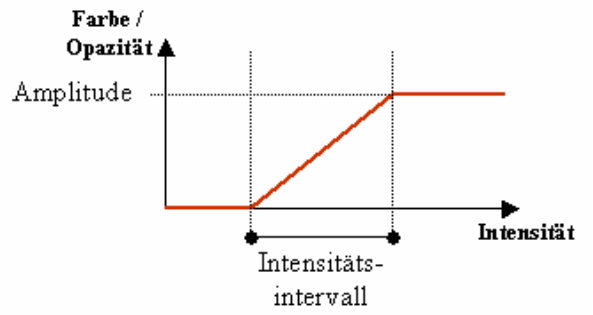


Abbildung 4-20: Geeignete Parameter für 1D-TFs.

Das Einstellen dieser Parameter kann auf dieselbe Weise geschehen, wie es bereits für 2D-Komponentenfuntionen beschrieben wurde. Eine Ausnahme ist der Distanzparameter. Weil 1D-TFs ausschließlich in Abhängigkeit von der Bildintensität festgelegt werden, muss ihnen im Nachhinein ein Distanzwert zugewiesen werden. 1D-TFs sind für alle Distanzwerte festzulegen, an denen sich die Farb- / Opazitätseigenschaften der zu erzeugenden 2D-TF ändern soll.

Beispiel für die Definition mit 1D-TFs

Dem nachfolgenden Beispiel liegt dieselbe Aufgabenstellung zugrunde, wie sie zur Veranschaulichung der Definition mit Hilfe von 2D-Komponentenfunktionen (Abschnitt 4.5.1) gestellt wurde: Visualisierung einer 5 mm dicken opaken Schicht innerhalb der Lunge. Diese Schicht soll einen konstanten Abstand von 10 mm von der Lungenoberfläche aufweisen.

Der Distanzraum teilt sich in diesem Beispiel in 3 Bereiche auf. Für den Distanzbereich unterhalb von +10 mm sind alle Strukturen transparent darzustellen. Im Bereich von +10 mm bis +15 mm sollen Strukturen vollständig opak und mit einem intensitätsabhängigen Farbverlauf visualisiert werden. Strukturen ab einer Distanz von +15 mm sind wieder transparent darzustellen.

Die Definition kann durch vier 1D-TFs erfolgen, die an den Grenzen dieser Distanzbereiche festgelegt werden:

- *10 mm*: Hier ist eine 1D-TF zu definieren, die für den gesamten Intensitätsbereich eine Opazität von 0 festlegt. Das Intensitätsintervall ist also auf [-1024HU, 3071HU], die Opazität auf 0 und die Farbe auf (0,0,0) zu setzen. Außerdem ist für Opazitäts- und Farbfunktion ein Funktionstyp mit konstantem Verlauf zu wählen (_|‾|_).
- *10 mm + 0.001 mm*: für diese Distanz wird eine 1D-TF benötigt, die innerhalb des Intensitätsbereichs von Lungengewebe eine konstante Opazität von 1 und ein kontinuierlichen Farbverlauf realisiert. Demnach ist ein Intensitätsintervall von [-1024, -424],

eine konstante Opazitätsfunktion (_|‾|_), eine steigende Farbfunktion (_/‾) eine Opazität von 1 und die gewünschte Farbe auszuwählen.
- *15 mm – 0.001 mm*: wie zweite 1D-TF (10 mm + 0.001 mm)
- *15 mm*: wie erste 1D-TF (10 mm)

Die resultierende LUT ist mit der durch 2D-Komponentenfunktionen definierten identisch. Der Interaktionsaufwand ist jedoch als höher einzuschätzen. Anstatt einer Komponentenfunktion sind vier 1D-TFs festzulegen, von denen jede einzelne über 5 Parameter verfügt. Der höhere Interaktionsaufwand ist jedoch in Fällen, in denen hohe Flexibilität erforderlich ist, gerechtfertig. Durch eine geschickte Voreinstellung der Parameter kann der Interaktionsaufwand reduziert werden.

4.6 DVR unter Verwendung von DTFs und 3D-Texturmapping

Für die direkte Visualisierung von Volumendaten stehen verschiedenen Hard- und Softwarelösungen zur Verfügung (siehe Abschnitt 2.4). In dieser Arbeit wird das 3D-Textur-Verfahren verwendet. Mit Hilfe dieses Verfahrens ist das Erzeugen qualitativ hochwertiger Visualisierungen bei akzeptablen Kosten und guter Performanz möglich. Benötig wird hierfür eine geeignete Grafikkarte (muss 3D-Texturing unterstützen) sowie eine Software (Volumenrenderer), um diese anzusprechen. Die Berechnung der Visualisierung wird von der Grafikkarte ausgeführt. Der Volumenrenderer versorgt sie hierfür mit allen benötigten Daten und Steuerbefehlen.

Für eine Visualisierung mit DTFs müssen der Grafikhardware folgende Daten zur Verfügung gestellt werden:

- *Intensitätsdatensatz*: Dies kann ein üblicher CT- oder MRT-Datensatz sein. Dieser ist in Voxel unterteilt, wobei jedem Voxel ein Intensitätswert zugeordnet ist.
- *DistanceMap*: DistanceMaps sind Volumendatensätze, die die Distanz jedes Voxels des zugeordneten Intensitätsdatensatzes zu einer Referenzfläche enthalten.
- *Distanzabhängige 2D-LUT*: Dies sind Lookup-Tabellen mit deren Hilfe Voxeln in Abhängigkeit ihres Intensitäts- und Distanzwertes RGBA-Farbwerte zugewiesen werden können. Ein Pixel der LUT definiert dabei die Farbe (RGBA), mit der Voxel des Intensitätsdatensatzes, die in dem dem Pixel zugeordneten Intensitätswert- und Distanzintervall liegen, dargestellt werden.
- *Proxygeometrie*: Da aktuelle Grafikhardware nicht in der Lage ist Volumendaten direkt darzustellen, wird eine Proxygeometrie verwendet, auf die das Volumen projiziert wird. Diese Proxygeometrie wird vom Volumenrenderer erzeugt.
- *Steuerbefehle*

Der Intensitätsdatensatz wird von der Grafikkarte in bildebenen parallelen Schichten neu abgetastet. Dies geschieht mittels trilinearer Interpolation. Auf den neu interpolierten Intensitätswerten wird anschließend unter Verwendung des Distanzdatensatzes die 2D-LUT angewendet. Jedem Voxel wird dabei eine Farbe und Opazität zugewiesen, mit der er anschließend dargestellt wird. Hierfür ist es erforderlich, dass die LUT die Wertebereiche der beiden Bilddatensätze (Intensität, Distanz) vollständig abdeckt.

4.6.1 Einflüsse auf die Bildqualität

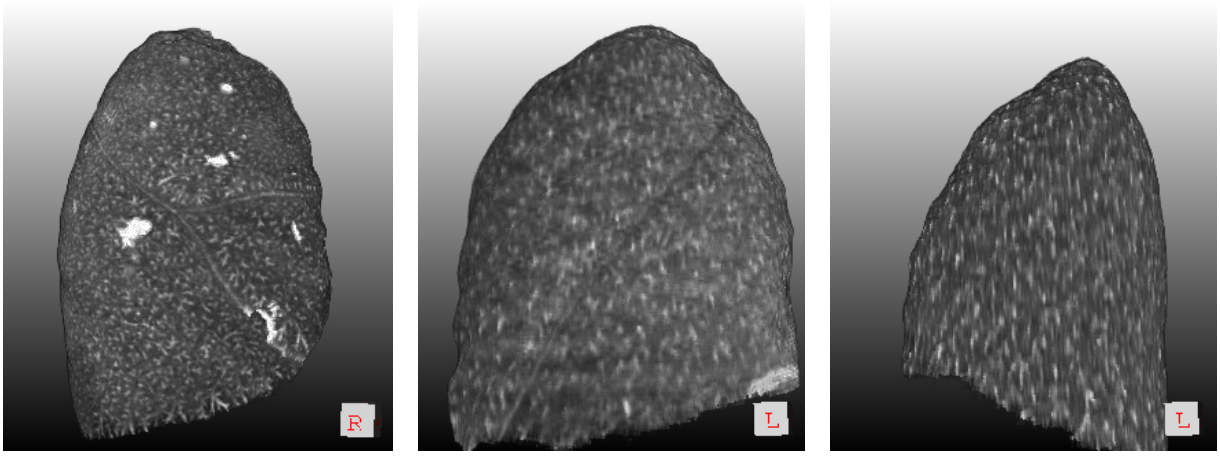
Die Bildqualität, die dabei erreicht wird, ist von zahlreichen Faktoren abhängig:

Korrekte Berechnung der DistanceMaps: Falls die Segmentierung / Berechnung der Referenzflächen fehlerhaft ist, können keine korrekten DistanceMaps erzeugt werden. Fehler in der DistanceMap beeinflussen das Visualisierungsergebnis negativ. Wenn beispielsweise die Segmentierung einer Organoberfläche, die als Referenzfläche dienen soll, fehlerhaft ist, weil zu viel bzw. zu wenig Gewebe segmentiert wurde, kann keine zur Organoberfläche äquidistante Schichtdarstellung erfolgen. Dies äußert sich in Form von Ausbuchtungen in den dargestellten Schichten. Solche Fehler können durch eine möglichst exakte Ermittlung der Referenzflächen minimiert werden.

Auflösung der LUTs: Es können LUTs mit verschiedenen Auflösungen zum Einsatz kommen. Je höher die Auflösung der verwendeten LUT desto differenzierter kann die Definition der TFs erfolgen (siehe Abschnitt 2.5, Lookup-Tabellen). Die maximale Auflösung wird durch die für LUTs unterstützte Bittiefe der Grafikkarte bestimmt.

Bittiefen: Grafikkarten unterstützten verschiedene Bittiefen für 3D-Texturen und LUTs. Übliche Werte liegen im Bereich von 8, 10, 11 und 12 Bit. Je höher die unterstützten Bittiefen, desto genauer können Intensitäts- / Distanzwerte differenziert werden. Eine Bittiefe von 8 lässt lediglich eine Differenzierung von 256 verschiedenen Werten zu, während mit 12 Bit bereits 4096 Werte unterschieden werden können. In dieser Arbeit wurde ein GeForce Quadro verwendet. Diese unterstützt lediglich eine Bittiefe von 8. Dies bereitete mitunter Probleme bei der Differenzierung geringer Intensitätsunterschiede in CT-Aufnahmen, da Unterschiede kleiner 16HU (4096HU/256) nicht mehr durch 8-Bit-LUTs aufgelöst werden konnten. Es besteht jedoch die Möglichkeit den Intensitätsbereich des Bildmaterials auf die Intensitäten der interessierenden Gewebe zu reduzieren und so eine bessere Auflösung der LUTs zu erreichen. Für die Volumenvisualisierung von CT-Aufnahmen ist eine Bittiefe von 12 ideal, für MRT-Datensätzen genügen 8 bzw. 10 Bit.

Auflösung des Bildmaterials: Die Auflösung des verwendeten Intensitätsdatensatzes hat erheblichen Einfluss auf die Renderingqualität. Gute Ergebnisse sind bei Voxelgrößen ab ca. 1×1×2.5 mm zu erzielen. Die besten Ergebnisse wurden mit hoch aufgelösten CT-Datensätzen bei Voxelgrößen unter 0.8×0.8×0.8 mm erreicht. Ein Problem ergibt sich aus dem Umstand, dass CT- und im besonderen Maße MRT-Datensätze oft einen recht großen Schichtabstand aufweisen. Bei Schichtabständen größer 4 mm sind kaum akzeptable Volumenvisualisierungen möglich. In Abbildung 4-21 sind Ergebnisse mit unterschiedlich aufgelösten Datensätzen zusammengestellt.



CT: 0.617×0.617×0.8 mm (a) | CT: 0.820×0.820×2 mm (b) | CT: 0.742×0.742×4 mm (c)

Abbildung 4-21: Visualisierungen verschieden aufgelöster Bilddatensätze.

Abbildung (a) zeichnet sich durch hohe Schärfe aus. Selbst kleine Gefäße, welche als helle Gebiete auf der Oberfläche sichtbar sind, können gut differenziert werden. Im Gegensatz hierzu ist Abbildung (c) sehr verschwommen. Die Unterschiede resultieren aus der um Faktor 5 verschiedenen Auflösung des Bildmaterials entlang der z-Achse.

Parametrierung des Volumenrenderers: Volumenrenderer verfügen meist über eine Vielzahl von Parametern, mit denen sich Einfluss auf die Visualisierung nehmen lässt. Hierzu gehören Parameter für die Einstellung von Beleuchtung, Abtastraten, Renderingverfahren etc. Geeignete Einstellungen lassen sich oft sehr schnell durch einige Tests ermitteln.

4.7 Presets

Im klinischen Einsatz ist die manuelle Definition einer größeren Anzahl von Parametern inakzeptabel. Außerdem ist es aufwändig die Parameter von TFs für jede Visualisierung neu festzulegen. Günstiger ist es, eine Bibliothek von TFs anzulegen, aus der je nach Visualisierungsziel die passende geladen werden kann. Dafür ist es erforderlich, TFs bereit zu stellen, die ohne bzw. mit sehr geringem Adaptionsaufwand auf unterschiedliche Datensätze angewendet werden können. Derartige TFs werden im Folgenden als Presets bezeichnet. Presets lassen sich auf ein konkretes Visualisierungsziel hin optimieren und können ausschließlich für dieses verwendet werden.

4.7.1 Anwendung von Presets - Theoretische Betrachtung möglicher Probleme

Presets werden im Falle von DTFs in Abhängigkeit von Intensitätswerten sowie von Distanzen zu Referenzflächen im Datensatz definiert. Damit sie auf unterschiedliche Datensätze angewendet werden können, müssen zwei Bedingungen erfüllt sein:

- Bei der Definition von DTF werden durch die Auswahl von Intensitätsintervallen die darzustellenden Gewebetypen festgelegt. Eine wesentliche Voraussetzung für die Verwendung von Presets ist deshalb, dass in allen Datensätzen, die mit einem Preset visualisiert werden sollen, die enthaltenden Gewebetypen annähernd denselben Intensitätsintervallen zugeordnet sind bzw. eine dahingehende Anpassung erfolgen kann.

- Eine weitere Bedingung ergibt sich aus den verwendeten Referenzflächen. Diese werden in den meisten Fällen aus den Bilddaten segmentiert bzw. abgeleitet. Da DTFs in Abhängigkeit von der Distanz zu diesen Referenzflächen definiert werden, ist es wichtig, dass deren Segmentierung / Ableitung für verschiedene Datensätze vergleichbare Ergebnisse liefert.

Mögliche Probleme bei der Berechnung von Referenzflächen

Abweichende Anatomie: Eine abweichende Anatomie führt zur Berechnung unterschiedlicher Referenzflächen / DistanceMaps. Da Presets distanzabhängig definiert werden, können größere Abweichungen der Anatomie Anpassungen der Presets erforderlich machen. Probleme könnte es beispielsweise bei der Definition von Presets geben, die sowohl für Datensätze von Kindern als auch für Datensätze von Erwachsenen Verwendung finden sollen, da hier mit stark variierenden Distanzen, beispielsweise zwischen Organen, zu rechnen ist. Eine einfache Lösung besteht darin, unterschiedliche Presets zu definieren. Komplexere Lösungen könnten eine alters- / größenabhängige Verschiebung aller kritischen Distanzen des Preset durchführen. Weitere anatomische Unterschiede können durch krankhafte Veränderungen wie beispielsweise Tumore entstehen. Dies sollte bei der Definition von Presets Beachtung finden. Geschieht das nicht, ist die Darstellung dieser Veränderungen nicht explizit festgelegt. Im schlechtesten Fall könnte ein Ausblenden erfolgen, weil die Opazitätsfunktion für das betreffende Intensitätsintervall mit 0 definiert wurde.

Unterschiedliche Algorithmen: Die Verwendung unterschiedlicher Algorithmen kann zu verschiedenen Ergebnissen führen. Referenzflächen / DistanceMaps sollten deshalb für alle Datensätze mit denselben Algorithmen berechnet werden. Auf diese Weise wird eine Reproduzierbarkeit der Ergebnisse gewährleistet, was besonders im medizinischen Bereich eine wesentliche Anforderung darstellt.

Probleme durch abweichende Gewebeintensitäten

Kontrastmittel: Ein weiterer Aspekt, der in diesem Zusammenhang nicht zu vernachlässigen ist, ist die Gabe von Kontrastmittel. Durch Kontrastmittel erhöhen sich die Intensitätswerte bestimmter Gewebetypen. Dies muss bei der Definition von Presets beachtet werden. Da die Verteilung von Kontrastmittel im Körper gut vorhersagbar ist, können Presets leicht dahingehend angepasst werden. In den meisten Fällen genügt eine Vergrößerung der betroffenen Intensitätsintervalle.

Bildinhomogenitäten bei MRT-Aufnahmen: Ohne Kontrastmittelgabe sind die Intensitätsbereiche bei CT-Aufnahmen für selbe Gewebetypen nahezu identisch. Dies gilt jedoch nicht für MRT-Aufnahmen. Diese sind von lokalen / globalen Bildinhomogenitäten durchzogen, die aus den physikalischen Eigenschaften des Messsystems resultieren und nicht vermieden werden können.

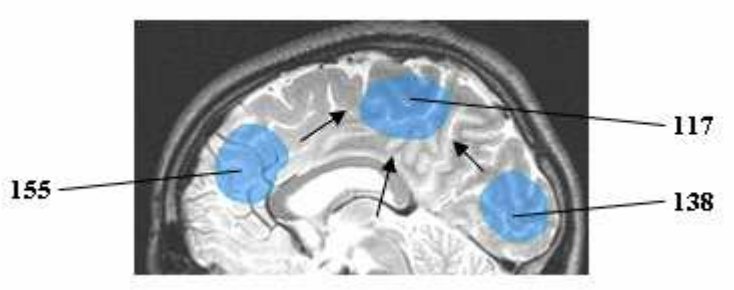


Abbildung 4-22: Bildinhomogenitäten bei MRT-Aufnahmen. Die mittlere Intensität nimmt in Richtung der Pfeile ab. Für die eingefärbten Regionen wurden Mittelwerte berechnet.

Durch einen Ausgleich der Intensitätsschwankungen kann das Problem reduziert, jedoch nicht behoben werden. Derartige Korrekturverfahren sind Gegenstand aktueller Forschung und standen für diese Arbeit nicht zur Verfügung.

4.7.2 Anwendung von Presets - Untersuchungen verschiedener Datensätze

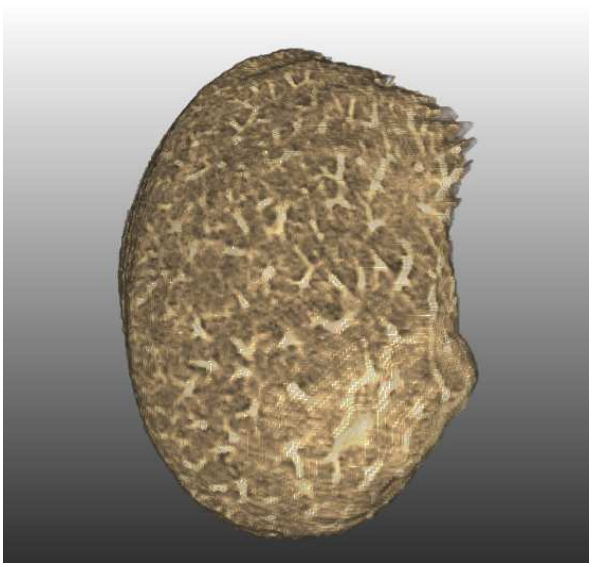
In diesem Abschnitt wird die Möglichkeit untersucht, allgemeingültige Presets zu definieren, d.h. vergleichbare Ergebnisse mit unterschiedlichen Datensätzen zu erzielen. Hierzu wird in drei Versuchen auf verschiedene Aspekte eingegangen, die bei der Suche nach geeigneten Presets relevant sind.

Versuch I (CT, Thorax, kontrastiert: Schichtvisualisierung der Lunge)

Ziel dieses Versuches ist es, die Eignung von Presets für die Visualisierung von CT-Datensätzen sowie die Auswirkung von Kontrastmittel zu untersuchen. Zu diesem Zweck werden vier zufällig gewählte kontrastierte Thoraxdatensätze ausgewertet. Dabei kommt ein Preset zum Einsatz, der die opake Darstellung einer zur Lungenoberfläche äquidistanten Schicht des Lungengewebes ermöglicht. Die Datensätze werden zunächst auf dieselbe anatomische Region zugeschnitten. Anschließend werden für alle Datensätze Referenzflächen (Lungenoberfläche) und DistanceMaps berechnet. Die Visualisierung erfolgt unter Verwendung desselben Presets.

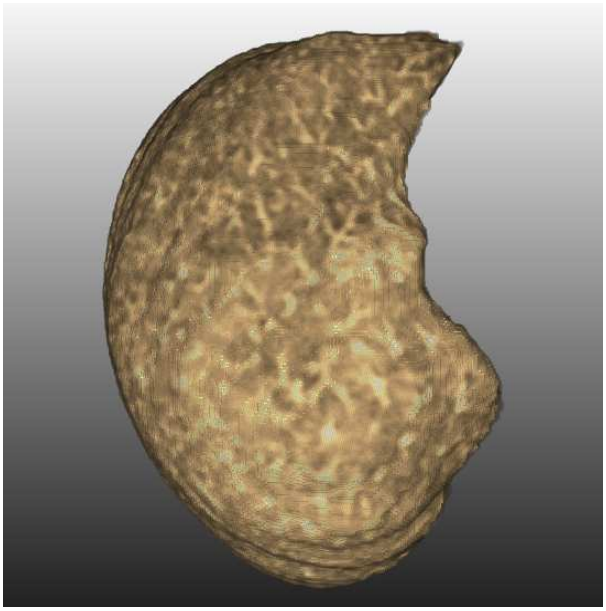
Auf Grund der Standardisierung von CT-Aufnahmen ist es leicht möglich, ein einheitliches Intensitätsintervall für einen bestimmten Gewebetyp zu ermitteln, was die Definition von Presets vereinfacht. In diesem Versuch ist es deshalb leicht, das darzustellende Intensitätsintervall festzulegen. Es beträgt für Lungengewebe –900 bis –200HU. Für die Farbfunktion wird ein linearer Anstieg innerhalb dieses Intervalls definiert. Die Opazitätsfunktion ist für die gestellte Visualisierungsaufgabe ausschließlich distanzabhängig zu definieren. Die Opazität eines Voxels ergibt sich aus seiner Distanz zu der Referenzfläche. Sein Intensitätswert hat in diesem Fall keinen Einfluss auf die Opazität. Aus diesem Grund machen sich Unterschiede in der Kontrastmittelverteilung lediglich durch Farbdifferenzen in der Visualisierung bemerkbar. Diese sind jedoch gering und beeinträchtigen kaum die Qualität der erzeugten Bilder. Ausschlaggebend ist eine korrekte Ermittlung der Referenzflächen. An Stellen, wo Gefäße aus der Lunge führen, können Ausbuchtungen der Referenzfläche entstehen. Dies sollte so weit wie möglich vermieden werden, da sie die Ergebnisse der Distanzberechnung verfälschen.

Ein Vergleich der Visualisierungsergebnisse muss vor dem Hintergrund erfolgen, dass die Datensätze Unterschiede in der Anatomie aufweisen, sowie verschieden aufgelöst sind. Durch einen optischen Vergleich der erzeugten Bilder kann die Eignung des festgelegten Presets für die gestellte Visualisierungsaufgabe bestätigt werden (Abbildung 4-23). Außerdem ist anzunehmen, dass der Preset auch für andere CT-Thorax-Datensätze geeignet ist.

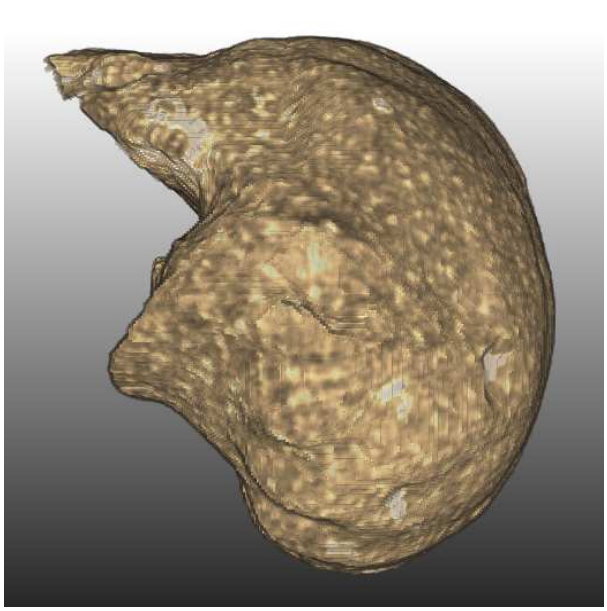


Datensatz I: CT, 0.742×0.742×4 mm

Datensatz II: CT, 0.820×0.820×2 mm



Datensatz III: CT, 0.682×0.682×4 mm



Datensatz IV: CT, 0.617×0.617×0.8 mm

Abbildung 4-23: Visualisierung einer 5 mm unter der Lungenoberfläche liegenden opaken Schicht. Es werden vier verschiedene CT-Datensätze unter Anwendung desselben Presets dargestellt. Alle Gewebe mit einer Distanz < 5 mm zur Lungenoberfläche werden dabei unter Verwendung einer distanzabhängigen Opazitätsfunktion ausgeblendet.

Für die Definition von Presets können folgende Erkenntnisse zusammengefasst werden:

- CT-Datensätze lassen sich aufgrund ihrer Standardisierung sehr gut unter Verwendung von Presets visualisieren.
- Eine rein distanzabhängige Definition der Opazitätsfunktion macht Presets stabil gegen Intensitätsschwankungen. Opazitätsfunktionen lassen sich immer dann unabhängig von Intensitätswerten definieren, wenn flächige Ansichten von Strukturen erwünscht sind. Die dazustellenden „Schnittflächen“ haben dabei einen konstanten Abstand zur verwendeten Referenzfläche.
- Je ähnlicher sich die zu visualisierenden Datensätze hinsichtlich Anatomie und Intensitätsbereichen der enthaltenden Strukturen sind, desto einfacher ist das Festlegen gemeinsamer Presets.

Versuch II (CT, Thorax, kontrastiert: Visualisierung von Lungengefäßen)

Ziel ist eine Bestätigung der Eignung von CT-Datensätzen für die Visualisierung mit Presets, sowie eine Untersuchung von intensitätsabhängigen Opazitätsfunktionen. In diesem Versuch werden dieselben Datensätze wie in Versuch I verwendet. Als Referenzfläche dient ebenfalls die Lungenoberfläche. Der einzige Unterschied liegt darin, dass ein anderer Preset zur Anwendung kommt. Dieser soll die Gefäße innerhalb der Lunge visualisieren.

Hierfür erhalten alle Voxel außerhalb der Lunge (Distanz < 0 mm) eine Opazität von 0. Damit werden Gewebe, die nicht zur Lunge gehören, ausgeblendet. Strukturen innerhalb der Lunge (Distanz >= 0 mm) werden mit Hilfe einer intensitätsabhängigen Farb- und Opazitätsfunktion dargestellt. Das Intensitätsintervall von Farb- und Opazitätsfunktion wurde dabei so gewählt, dass die Lungengefäße sichtbar sind [-800 HU, -400 HU]. Die intensitätsabhängige Definition der Opazitätsfunktion stellt sich, trotz des verwendeten Kontrastmittels und der damit verbundenen Intensitätsschwankungen, als unproblematisch heraus, da die Intensitätsintervalle der Gefäße durch das Kontrastmittel kaum bzw. nur geringfügig beeinflusst wurden. So konnte leicht ein einheitliches Intensitätsintervall gefunden werden. Alle Voxel mit einem Intensitätswert außerhalb dieses Intervalls werden transparent dargestellt. Auf diese Weise konnten die Gefäße gut dargestellt werden. Ein gemeinsamer Preset ließ sich sehr schnell finden.

Auch hier sind die Ergebnisse bei allen getesteten Datensätzen ähnlich (Abbildung 4-24).

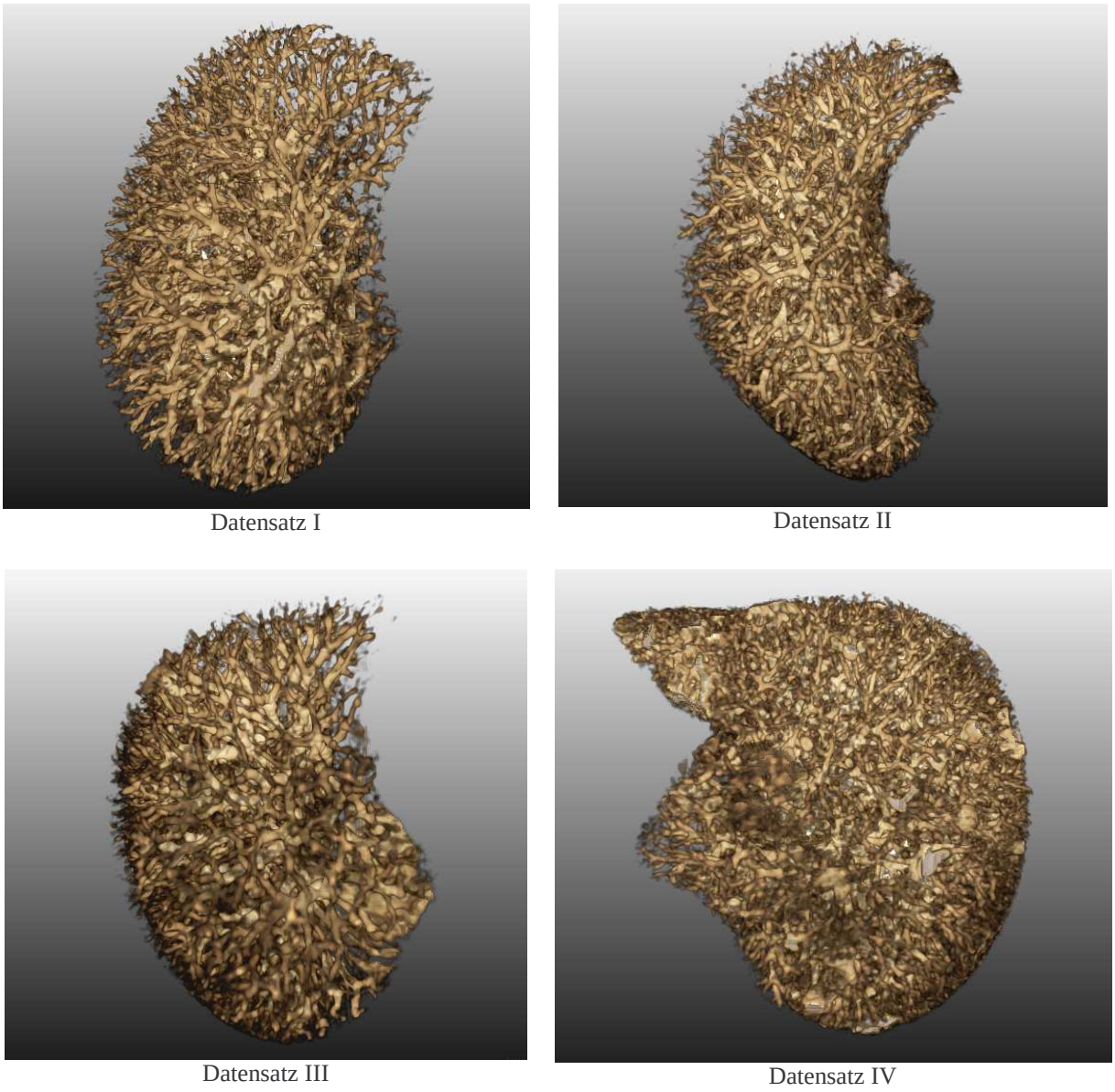


Abbildung 4-24: Visualisierung von Lungengefäßen. Hierbei wird eine intensitäts- und distanzabhängige Opazitätsfunktion verwendet, die sowohl alle Gewebe die sich außerhalb der Lunge befinden sowie Gewebe deren Intensität nicht der von Lungengefäßen entspricht ausgeblendet.

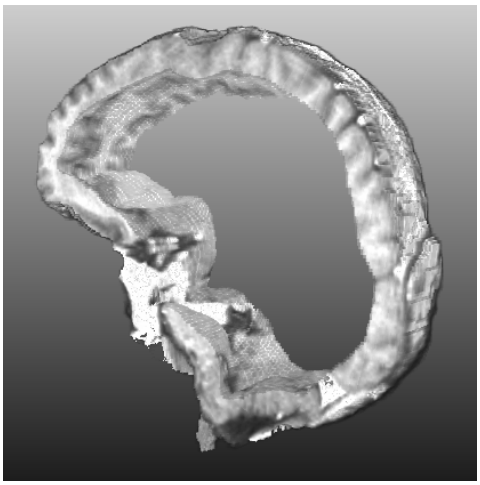
Zusammenfassend ist festzustellen:

- Die Aussage aus Versuch I, dass eine gute Verwendbarkeit von Presets für CT-Daten besteht, kann unterstrichen werden.
- Eine intensitätsabhängige Steuerung der Opazitätsfunktion ist solange unproblematisch, wie keine stärkeren unvorhergesehenen Intensitätsschwankungen innerhalb des darzustellenden Gewebes auftreten. Diese lässt auf mögliche Probleme bei der Visualisierung von MRT-Aufnahmen schließen. Im nächsten Versuch wird eine dahingehende Untersuchung durchgeführt.

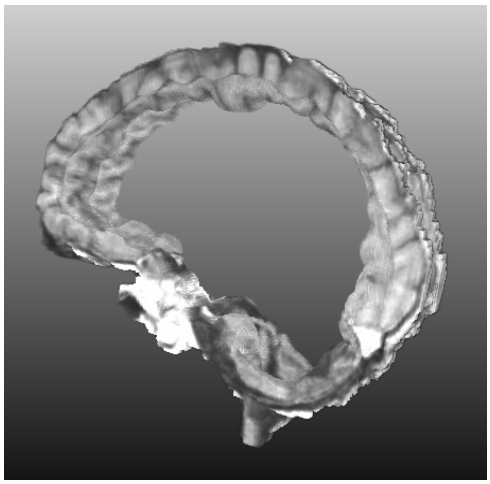
Versuch III (MRT-Datensätze, Kopf: Schichtvisualisierung des Kortex)

Ziel dieses Versuchs ist es, die Eignung von Presets für das DVR von MRT-Aufnahmen zu untersuchen. Die Definition von Presets stellt hier eine besondere Herausforderung dar, da mit stärkeren Schwankungen von Gewebeintensitäten zu rechnen ist. Das Visualisierungsziel besteht darin, eine 5 mm starke Schicht des Kortex darzustellen.

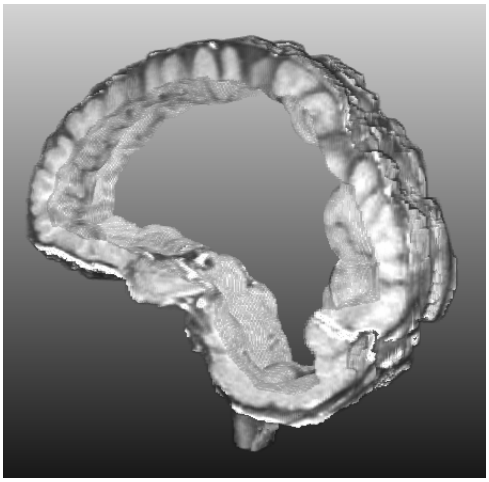
Die Opazitätsfunktion ist hierfür rein distanzabhängig zu definieren. Gewebe außerhalb des gewünschten Distanzintervalls werden ausgeblendet. Die Intensität eines Voxels hat damit keinen Einfluss auf seine Transparenz. Dies ist in anbetracht unvorhersagbarer Intensitätswerte (MRT-Datensätze) ein erheblicher Vorteil. Für die Farbfunktion ist eine intensitätsabhängige Definition kaum zu vermeiden, da eine realistische Färbung nur in Abhängigkeit von den Intensitätswerten erfolgen kann. Hierfür ist gegebenenfalls eine Anpassung der Intensitätsintervalle des Presets erforderlich. Dies kann sowohl interaktiv als auch (semi-)automatisch durch eine Histogrammanalyse geschehen. In diesem Versuch wurden drei MRT-Datensätze visualisiert. Das Intensitätsintervall des verwendeten Presets [0, 190] musste dabei für einen Datensatz (III) auf [0, 500] vergrößert werden. Darüber hinaus war keine weitere Anpassung des Presets erforderlich. Die Visualisierungsergebnisse sind in Abbildung 4-25 dargestellt. Eine Allgemeingültigkeit des verwendeten Presets kann, unter der Voraussetzung der Anpassung des angesprochenen Intensitätsintervalls, angenommen werden.



Datensatz I: MR, 0.977×0.977×2.5 mm



Datensatz II: MR, 1×1×2 mm



Datensatz III: MR, 0.977×0.977×2.5 mm

Abbildung 4-25: Visualisierung einer 5 mm starken opaken Schicht des Kortex. Hierfür kommt eine rein distanzabhängige Opazitätsfunktion zum Einsatz.

Zusammenfassend ist festzustellen:

- Die Verwendung von Presets ist auch bei der Visualisierung von MRT-Datensätzen möglich. Dabei ist jedoch Adaptionsaufwand für die Anpassung von Intensitätsintervallen in Kauf zunehmen.
- Eine intensitätsabhängige Definition der Opazitätsfunktion ist in den meisten Fällen nicht sinnvoll. Stärkere Intensitätsschwankungen, aufgrund von Inhomogenitäten oder unterschiedlichen Aufnahmeparametern, lassen oft keine vorhersagbaren Ergebnisse zu.

4.7.3 Parametrisierte Presets

Presets können Parameter enthalten, die während der Visualisierung interaktiv manipulierbar sind. Bei der Arbeit mit DTFs haben sich zwei Parameter als besonderst nützlich erwiesen. Diese werden im Folgenden erläutert.

Verschiebung ausgewählter Distanzwerte

Ein häufiges Ziel distanzabhängiger Visualisierung ist die Darstellung von Gewebeschichten, die sich in einem bestimmten Abstand zu Referenzflächen befinden. Von Interesse kann dabei eine interaktive Manipulation dieses Abstandes sein. Hierfür sind die betreffenden Distanzwerte im Preset festzulegen. Diesen wird ein vom Anwender einstellbarer Offset aufaddiert. Auf diese Weise kann der Schichtabstand während der Visualisierung beliebig angepasst werden. Abbildung 4-26 zeigt diese am Beispiel des menschlichen Kortex.

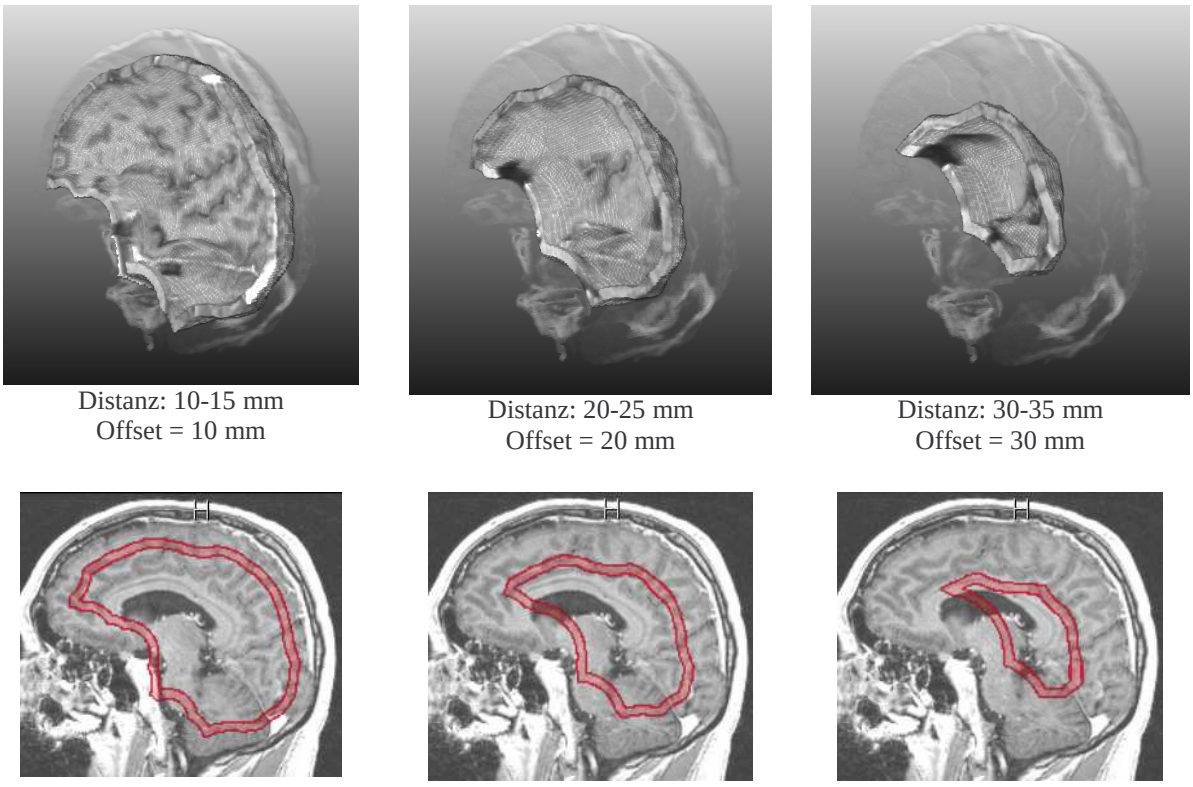


Abbildung 4-26: Interaktive Manipulation von Schichtabständen.

Dargestellt wird eine 5 mm starke Gewebeschicht sowie die semitransparente Kortexoberfläche. Der Abstand der Gewebeschicht zur Kortexoberfläche wird durch die Manipulation von Distanzwerten des verwendeten Presets verändert. Da für jeden Distanzwert der Einfluss des Offsets getrennt festgelegt wird, ist es möglich eine gezielte Verschiebung ausgewählter Distanzwerte zu erreichen. Dadurch kann eine Verschiebung der Gewebeschicht erfolgen, ohne dass dabei die Darstellung der Kortexoberfläche beeinflusst wird.

Verschiebung aller Distanzwerte

Durch die Verwendung unterschiedlicher Algorithmen oder Parametrierungen bei der Berechnung von Referenzflächen / DistanceMaps kann es vorkommen, dass die DistanceMaps um einen konstanten Wert verschoben sind. Falls dies nicht bereits bei ihrer Generierung korrigiert wurde, ist es möglich einen entsprechenden Ausgleich im Preset vorzunehmen. Hierfür ist allen Distanzwerten des Presets ein konstanter Wert hinzu zu addieren. In Abbildung 4-27 ist dieses Vorgehen illustriert.

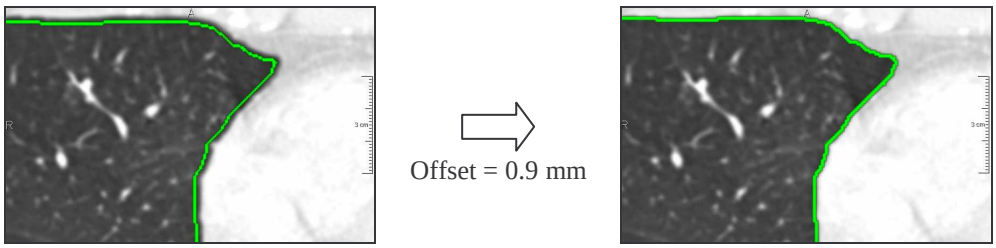


Abbildung 4-27: Korrektur einer DistanceMap durch konstante Verschiebung aller Distanzwerte im Preset. Die 0 mm-Distanzen sind jeweils durch eine grüne Linie gekennzeichnet.

Hierfür ist keine Neuberechnung der DistanceMaps erforderlich. Eine Anpassung kann während der Visualisierung durch Einstellen des Offset-Parameters erfolgen. Auf diese Weise ist eine einfache Korrektur verschobener Distanzwerte möglich.

4.7.4 Auswahl von Presets

Die Auswahl von Presets kann durch Icons unterstützt werden. Diese ermöglichen eine schnelle visuelle Einschätzung der Preseteigenschaften. Um Eigenschaften anzugeben, die nicht oder nur sehr schlecht aus Icons abzuleiten sind, sind zusätzliche textuelle Beschreibungen sinnvoll. Icons können leicht durch Volumenvisualisierung mit dem entsprechenden Preset erzeugt und nach Visualisierungszielen gruppiert für eine Auswahl präsentiert werden.

4.8 Zusammenfassung

In diesem Kapitel wurde ein distanzabhängiger Ansatz für die Definition von TFs vorgestellt. Zunächst wurde auf die Grundlagen von DTFs eingegangen. Anschließend wurde das Bereitstellen von DistanceMaps, die Definition von DTFs sowie die Möglichkeit der Verwendung von Presets erläutert. Darüber hinaus wurde der Rederingprozess unter Verwendung von 3D-Texturen erläutert sowie Faktoren diskutiert, die Einfluss auf die Bildqualität der erstellten Visualisierungen nehmen.

Durch die Einbeziehung von Distanzinformationen in den Visualisierungsprozess konnte die Flexibilität bei der Definition von TFs erhöht werden. Eine distanzabhängige Steuerung von Farb- und Transparenzwerten eröffnet eine ganze Reihe neuer Visualisierungsmöglichkeiten, von denen einige in Kapitel 6 näher diskutiert werden. Der vorgestellte Ansatz ist sowohl für die Darstellung von CT- als auch für MRT-Datensätze geeignet und bietet damit Spielraum für verschiedene Anwendungen.

5 Implementierung eines Editors zum Entwurf von DTFs

Im Zuge dieser Diplomarbeit wurde ein Editor zum Entwurf von DTFs basierend auf der medizinischen Bildverarbeitungssoftware MeVisLab (http://www.mevislab.de/) implementiert. Im nachfolgenden Abschnitt werden zunächst einige Grundlagen von MeVisLab vorgestellt. Anschließend werden Anforderungen an einen Editor für die Erzeugung von DTFs diskutiert. MeVisLab stellt bereits einen Großteil der für diesen Editor benötigten Funktionalität, wie das Einlesen, Verarbeiten und Darstellen von Bilddaten zur Verfügung. Darüber hinaus ist es erforderlich, zusätzliche Klassen / Module bereitzustellen, worauf entsprechend eingegangen wird. Abschließend wird die Realisierung des Editors diskutiert.

5.1 Grundlagen von MeVisLab

MeVisLab wurde / wird von den MeVis gGmbH, einem Aninstitut der Universität Bremen, für die Analyse medizinischer Bilddaten entwickelt. Es besteht aus einer Reihe von Bildverarbeitungs- und Visualisierungsmodulen sowie einer Benutzeroberfläche. Die Bildverarbeitungsmodule sind von der ML (MeVis Image Processing Library, C++) abgeleitet und implementieren eine Vielzahl von Algorithmen für die Aufbereitung und Analyse von Bilddaten. Dies beinhaltet beispielsweise verschiedene Filter, Segmentierungs -und Registrierungsverfahren sowie Bildtransformationen. Mit Hilfe von OpenInventor-Modulen werden dem zusätzliche Funktionalitäten wie die 2D- und 3D-Darstellung und die Interaktion mit Bilddaten hinzugefügt. Die verschiedenen Module verfügen über Connectoren, mit deren Hilfe sie sich unter einer Benutzeroberfläche zu komplexen Netzwerken und Macro-Modulen verknüpfen lassen. Abbildung 5-1 verdeutlicht dieses Vorgehen am Beispiel einer einfachen Bildverarbeitungsaufgabe. Ein Volumendatensatz soll in den Arbeitsspeicher geladen, gaußgefiltert und danach in Form von Schichtbildern dargestellt werden. Hierfür werden drei Module benötigt. Das *ImgLoad*-Modul lädt zunächst den Volumendatensatz und übergibt ihn über seinen Ausgang an den Eingang des *Convolution*-Moduls. Vom *Convolution*-Modul wird ein Gaußfilter angewendet. Der so manipulierte Datensatz wird anschließend über den Ausgang des *Convolution*-Moduls an den Eingang eines Viewers (*View2D*-Modul) weitergegeben und dargestellt. Die meisten Module verfügen über ein Panel, in dem verschiedene Parametrisierungen vorgenommen werden können. So ist es beispielsweise für das *ImgLoad*-Modul erforderlich, den Pfad des zu ladenden Datensatzes anzugeben. Im Panel des *Convolution*-Moduls kann unter anderem der verwendete Filtertyp und die Größe der Filtermaske ausgewählt werden. Das Panel des Viewers visualisiert den Datensatz.

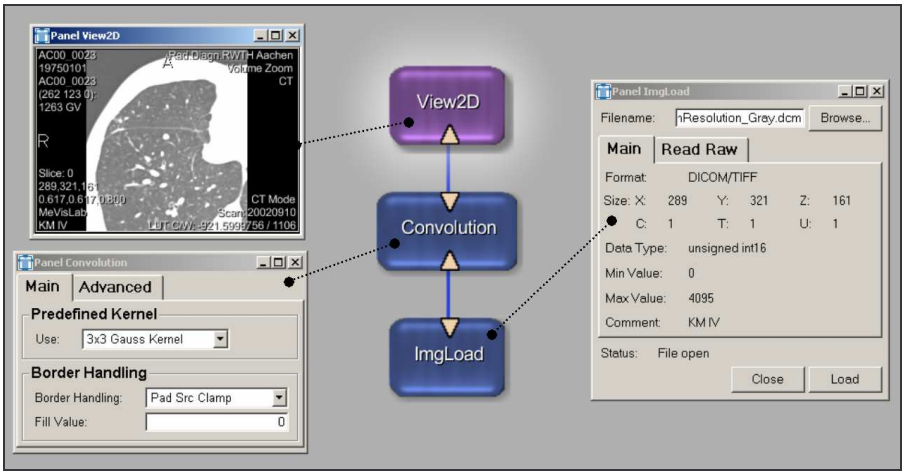


Abbildung 5-1: MeVisLab - Verknüpfung von Modulen zu Netzwerken. Die Module können über ihre Panel parametrisiert werden.

Viele Bildverarbeitungsaufgaben lassen sich durch das Verknüpfen vorhandener Module lösen. Durch das Ableiten von ML- bzw. Inventorklassen können bedarfsabhängig neue Module implementiert werden.

5.2 Anforderungen an einen Editor für DTFs

Die Hauptanforderung an einen DTFs-Editor besteht in der Bereitstellung von DTFs für die Visualisierung von CT- und MRT-Datensätzen. Im Vordergrund steht hierbei eine hohe Flexibilität beim Definitionsprozess sowie eine einfache Bedienbarkeit. Die erforderliche Benutzerinteraktion ist hierfür auf ein möglichst geringes Maß zu beschränken. Dem Anwender sollte es schnell und ohne größeren Lernaufwand möglich sein, ein bestimmtes Visualisierungsziel zu erreichen. Um eine schnelle Auswahl von Voreinstellungen zu gewährleisten, sollte darüber hinaus der Einsatz von Presets ermöglicht werden.

Zusammenfassend werden folgende Anforderungen an einen Editor für DTFs gestellt:

- Definition von DTFs durch Interpolation zwischen 1D-TFs (Abschnitt 3.2.2)
- Definition von DTFs mit Hilfe von 2D-Komponentenfunktionen (Abschnitt 3.2.1)
- Erzeugen, Speichern und Laden von Presets (Abschnitt 4.7)
- Bereitstellung verschiedener Hilfswidgets zur Unterstützung der Definition von Repräsentationsparametern: 2D-Histogramm, Grauwertmaske, Distanzmaske, Maske der aktuellen LUT
- Erzeugen von LUTs mit beliebiger Größe
- Darstellen der aktuellen LUT

5.3 Zusätzlich erforderliche Module / Klassen

Für die Realisierung des Editors als MeVisLab-Macro-Modul ist die Implementierung zusätzlicher Module / Klassen erforderlich. Diese werden im Folgenden kurz erläutert.

5.3.1 Bimesh4-Klasse

Mit der *Bimesh4*-Klasse wurde die erforderliche Funktionalität für die Generierung von 2D-LUTs zur Verfügung gestellt. Mit Hilfe dieser Klasse können Repräsentationen für 2D-TFs (2D-Komponentenfunktionen, Interpolation zwischen 1D-TFs) erzeugt und in 2D-LUTs umgewandelt werden. Wichtige Methoden dieser Klasse sind beispielsweise:

```
void init(float yMin, float yMax, float xMin, float xMax);
```
Initialisieren einer LUT-Repräsentation über die Intervalle der abhängigen Variablen *x* und *y* (bei DTFs Intensität und Distanz).

```
bool add2DRamp( int typeRGB, int typeA, float yMin, ...);
```
Fügt der aktuellen LUT-Repräsentation eine 2D-Komponentenfunktion hinzu.

```
bool add1DRampByConstY( int typeRGB, int typeA, float y, ...);
```
Fügt der aktuellen LUT-Repräsentation eine 1D-Funktion (TF) hinzu.

```
void interpolateMissingNodesOverRows();
void interpolateMissingNodesOverColumns();
```
Methoden, um eine beliebige LUT-Repräsentation in eine rectilineare zu überführen.

```
void getInterpolatedValue( float y, float x, float* valueRGBA);
```
Tastet die aktuelle LUT-Repräsentation an einer konkreten Koordinate (*x*,*y* à Intensität, Distanz) ab.

```
void getImg(float* imgRange, int* imgBufferRange, int useBuffer, ...);
```
Erzeugt eine 2D-LUT mit der gewünschten Größe.

Die *Bimesh4*-Klasse verfügt darüber hinaus über weitere Methoden, auf die an dieser Stelle nicht weiter eingegangen werden soll. Da diese Klasse unabhängig von der ML implementiert wurde, kann sie auch in anderen C++ Projekten genutzt werden.

5.3.2 JointHist-Modul

Außerdem wurde das bereits vorhandene *JointHist*-Modul überarbeitet und erweitert. Dieses Modul dient der Berechnung von 2D-Histogrammen. Es verfügt über zwei Eingänge, an denen Volumendatensätze angeschlossen werden können. Aus diesen Datensätzen werden ein 2D-Histogramm sowie zwei 1D-Histogramme berechnet und an den Ausgängen zur Verfügung gestellt (siehe Abbildung 5-2).

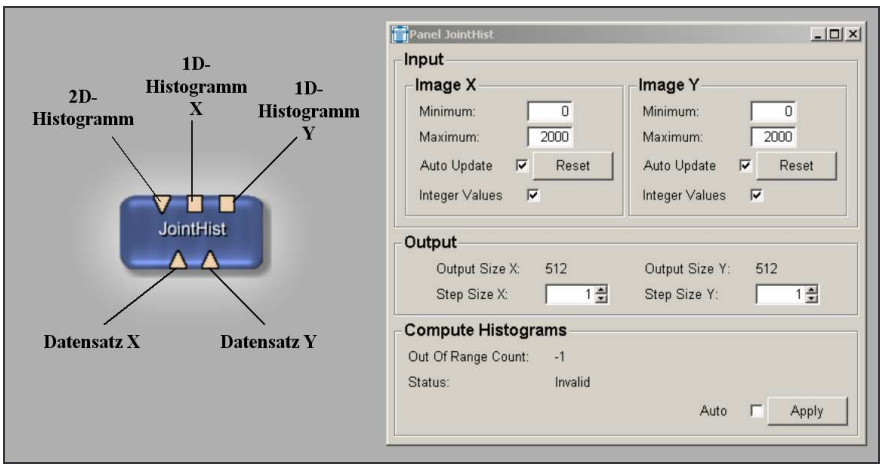


Abbildung 5-2: Das *JointHist*-Module und das zugehörige Panel.

Im Modul-Panel können einige Einstellungen vorgenommen werden. Dazu gehören die Angabe der auszuwertenden Werteintervalle beider Datensätze sowie die Größe der zu erzeugenden Histogramme.

5.3.3 JointHistMask-Modul

Das *JointHistMask*-Modul dient dazu auf Basis von 2D-LUTs oder Werteintervallen Masken aus zwei Datensätzen zu erzeugen. Die Datensätze müssen hierzu dieselbe Größe aufweisen. Eine so erzeugte Maske repräsentiert die von der angeschlossenen 2D-LUT bzw. den ausgewählten Intervallen abgedeckten Voxel. Ein Voxel wird also genau dann der Maske zugerechnet, wenn seine beiden skalaren Werte (aus den Datensätzen) innerhalb beider Intervalle liegen bzw. von der angeschlossenen 2D-LUT auf Werte ungleich (0,0,0,0) abgebildet werden. Das Modul verfügt über drei Eingänge, an denen zwei Datensätze sowie eine 2D-LUT bereitgestellt werden können. Es berechnet die beschriebene Maske und stellt sie am Modulausgang zur Verfügung (siehe Abbildung 5-3).

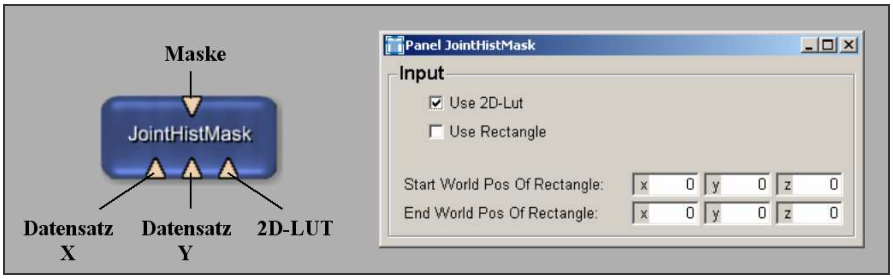


Abbildung 5-3: Das *JointHistMask*-Modul und das zugehörige Panel.

Im Panel des Moduls kann eine Auswahl des gewünschten Modus' (2D-LUT, Intervalle) erfolgen. Die Intervallgrenzen werden mit den Eckkoordinaten eines Rechteckes assoziiert, welche sich von außen in das Modul-Panel hinein verknüpfen lassen.

5.3.4 DistLutEditor-Modul

Das DistLutEditor-Modul stellt den eigentlichen Kern des Editor-Macros dar. Von diesem Modul wird die *Bimesh4*-Klasse instanziiert. Darüber hinaus werden alle Interaktionsparameter, die für die Parametrisierung von LUTs erforderlich sind, bereitgestellt und überwacht. Das Modul verfügt über vier Eingänge für das Anschließen von Grauwertdatensatz (CT- oder MRT-Aufnahme), DistanceMap, 2D-Histogramm (basierend auf Grauwert- und Distanzinformationen) sowie für das Laden von Presets (Base-Objekte). Darüber hinaus steht ein Ausgang für LUTs und ein Ausgang zum Speichern von Presets zur Verfügung (Abbildung 5-4). Im Panel des Moduls können zahlreiche LUT-Parameter eingestellt werden (Abbildung 5-5). Auf diese wird in Abschnitt 5.4 näher eingegangen.

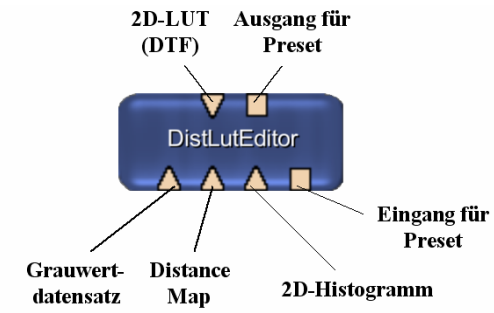


Abbildung 5-4: Ein- und Ausgänge des *DistLutEditor*-Moduls.

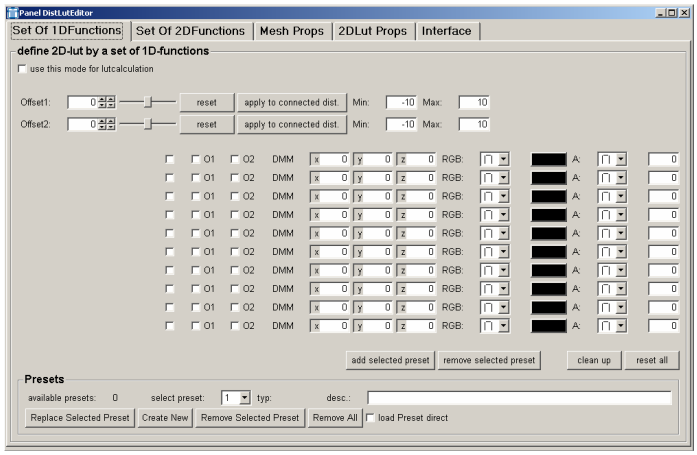


Abbildung 5-5: Panel des *DistLutEditor*-Moduls.

Für den Betrieb des *DistLutEditor*-Moduls ist ein Netzwerk aus mehreren Modulen anzulegen. Ein solches Netzwerk ist in Abbildung 5-6 dargestellt. Es wird im Folgenden kurz erläutert.

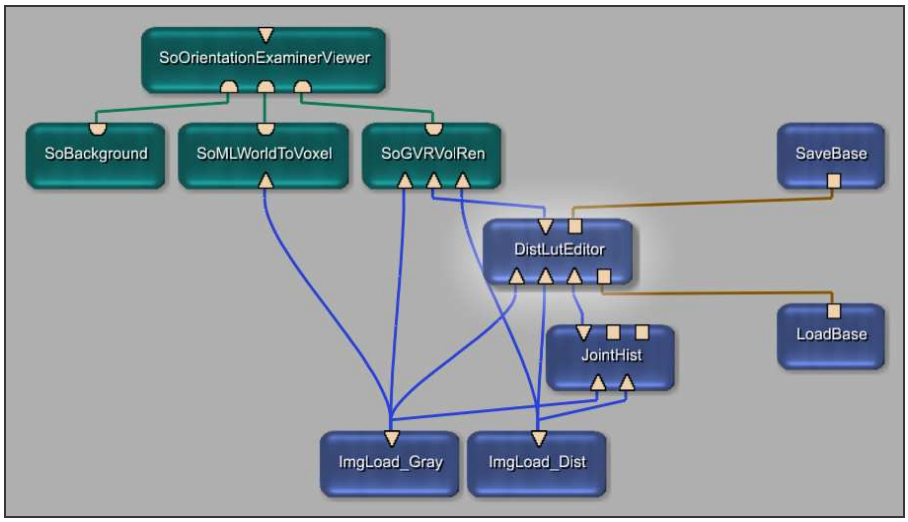


Abbildung 5-6: Beispielnetzwerk für den Betrieb des DistLutEditor-Modules.

Die Module *ImgLoad_Gray* und *ImgLoad_Dist* laden einen Grauwert- und Distanzdatensatz. Diese werden dem *JointHist*-, dem *SoGVRVolRen*- (Volumenrenderer) und dem *DistLutEditor*-Modul zur Verfügung gestellt. Das Modul *JointHist* berechnet ein 2D-Histogramm beider Datensätze und reicht dieses an das *DistLutEditor*-Modul weiter. Das Laden und Speichern von Presets erfolgt mit Hilfe der Module *LoadBase* und *SaveBase*. Die vom DistLutEditor berechnete 2D-LUT wird an den Volumenrenderer weitergegeben. Dieser erzeugt ein DVR, welches vom *SoOrientationExaminerViewer*-Modul dargestellt wird. Das Modul *SoML-WorldToVoxel* skaliert die Volumendarstellung auf die korrekten Seitenverhältnisse. Das Modul *SoBackground* fügt der Darstellung einen Hintergrund hinzu.

Das *DistLutEditor*-Modul stellt bereits grundlegende Funktionalität für die Definition von DTFs zur Verfügung. Es existieren jedoch keinerlei Hilfswidgets, um das Einstellen von Parametern zu unterstützen. Außerdem kann das Modul nur in einem komplexeren Netzwerk betrieben werden, da Datensätze, 2D-Histogramm, Presets und Viewer von außen anzuschließen sind. Eine elegantere Implementierung kann durch die Definition eines Macros bereitgestellt werden. Hierauf wird im nächsten Abschnitt eingegangen.

5.4 Das DistLutEditor-Macro

Der DTFs-Editor wurde außerdem als MeVisLab-Macro-Modul implementiert. Damit ist der Vorteil verbunden, dass auf die Funktionalität anderer MeVisLab-Module aufgebaut werden kann. Die darüber hinaus erforderliche Funktionalität wurde mit denen in Abschnitt 5.3 vorgestellten Modulen / Klassen bereitgestellt.

Das *DistLutEditorMacro* verfügt über zwei Eingänge für Grauwertdatensätze und Distance-Maps sowie einen Ausgang für 2D-LUTs (Abbildung 5-7).

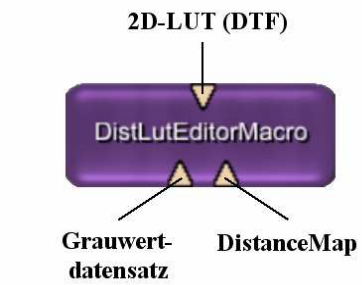


Abbildung 5-7: Anschlüsse des *DistLutEditorMacro*-Moduls.

5.4.1 Aufbau des DistLutEditorMacro-Moduls

Das *DistLutEditorMacro*-Modul wurde aus einer Vielzahl von MeVisLab-Modulen zusammengesetzt. Der Aufbau des Macros ist in Anhang A dargestellt. Zum besseren Verständnis sind die einzelnen Module zu Blöcken zusammengefasst, deren Funktion im Folgenden kurz erklärt wird.

Block 1: In diesem Block werden die Datensätze (Grauwertdatensatz und DistanceMap) geladen. Aus dem Pfad und Dateinamen des Grauwertdatensatzes wird dabei der Pfad und Dateiname der DistanceMap bestimmt.

Block 2: Block2 stellt den Kern des Macros dar. Hier befindet sich das *DistLutEditor*-Modul sowie weitere Module zum Laden und Speichern von Presets.

Block 3: Dieser Block enthält den Volumenrenderer sowie anderer Module, die für die Darstellung des DVRs benötigt werden.

Block 4: Durch diesen Block werden Preset-Vorschaubildern bereitgestellt. Die Vorschaubilder werden aus dem aktuellen DVR erzeugt.

Block 5: Block 5 verwaltet die Darstellung von Preset-Vorschaubildern. Die Vorschaubilder werden zusammen mit der zugehörigen Presetliste geladen und in einem Viewer präsentiert.

Block 6: Dieser Block dient der Darstellung von 2D-Histogammen sowie Interaktionselementen, die für das Einstellen von LUT-Parametern benötigt werden.

Block 7: Block 7 stellt einen Schichtbild-Viewer zur Verfügung in dem Schichten des aktuellen Grauwertdatensatzes angezeigt werden. Diese Schichten lassen sich mit verschiedenen Masken überlagern.

Block 8: Dieser Block dient der Darstellung der aktuellen LUT. Die LUT kann sowohl mit als auch ohne Alpha-Kanal angezeigt werden.

5.4.2 Funktion des DistLutEditorMacro-Moduls

Im Folgenden wird die Funktion des *DistLutEditorMacro*-Moduls erläutert. Eine Ansicht der Benutzeroberfläche ist in Abbildung 5-8 dargestellt.

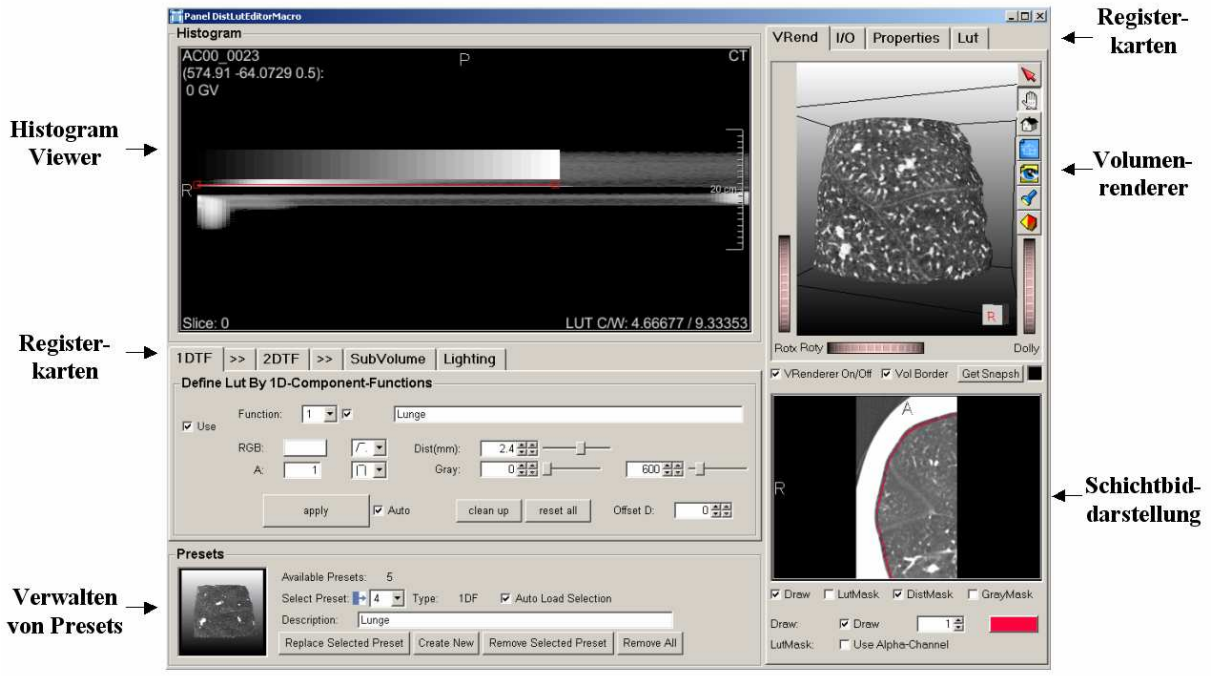


Abbildung 5-8: Benutzeroberfläche des DistLutEditorMacros.

Unter der Oberfläche wurden ein HistogramViewer für die Darstellung von 2D-Histogrammen, ein Volumenrenderer, ein Viewer für Schichtbilder, ein Viewer zum Darstel-

len der aktuellen LUT (unter Registerkarte LUT) sowie zehn Registerkarten, unter denen zahlreiche Einstellungen, beispielsweise für die Definition von TFs vorgenommen werden können, zusammengefasst. Außerdem wurde eine Verwaltung für Presets integriert. Für die Definition von LUTs wurden sowohl 2D-Komponentenfunktionen (Abschnitt 3.2.1) als auch die Interpolation zwischen 1D-TFs (Abschnitt 3.2.2) implementiert.

Laden von Datensätzen

Damit im Editor Schichtbilddarstellungen, DVRs und Histogramme angeboten werden können, ist es erforderlich den zu visualisierenden Grauwertdatensatz und die zugehörige DistanceMap zu laden. Diese können entweder von außen an das Editormodul angeschlossen oder intern über die Angabe eines Pfades geladen werden. Der gewünschte Modus ist mit Hilfe des „Use conncted Images“-Toggles unter der Registerkarte I/O zu wählen. Die Auswahl der Datensätze im Editor erfolgt ebenfalls unter dieser Registerkarte. Hierfür ist lediglich der Pfad des Grauwertdatensatzes anzugeben. Die DistanceMap wird unter demselben Pfad und demselben Dateinamen + Postfix „_dist“ gesucht. So muss der entsprechende Pfad nur einmal angegeben werden. Voraussetzung hierfür ist, dass die Datensätze entsprechend abgelegt sind. Nach dem erfolgreichen Laden der Datensätze wird auf deren Basis ein 2D-Histogramm berechnet und im HistogramViewer dargestellt.

Definition von LUTs durch Interpolation zwischen 1D-TFs

Um LUTs mit Hilfe von 1D-TFs zu definieren, ist zunächst der „Use“-Toggle unter der Registerkarte „1D-TF“ zu aktivieren. Dies ist erforderlich, um zwischen den beiden Definitionsmodi (Interpolation zwischen 1D-TFs, 2D-Komponentenfunktionen) zu wählen. Ist der Toggle aktiv gesetzt, können die Eigenschaften von maximal zehn 1D-TFs festgelegt werden. Die resultierende 2D-TF wird durch Interpolation aus den 1D-TFs bestimmt. Mit Hilfe der „Function“-Popupliste lässt sich eine 1D-TF für die Bearbeitung auswählen. Für jede dieser 1D-TFs können folgende Parameter festgelegt werden:

- Farbe
- Verlauf der Farbfunktion
- Maximale Opazität
- Verlauf der Opazitätsfunktion
- Distanz der Funktion
- Intensitätsintervall der Funktion
- Textuelle Beschreibung

Abbildung 5-9 erklärt die Interaktionselemente, die für eine Definition mit Hilfe von 1D-TFs bereitgestellt wurden.

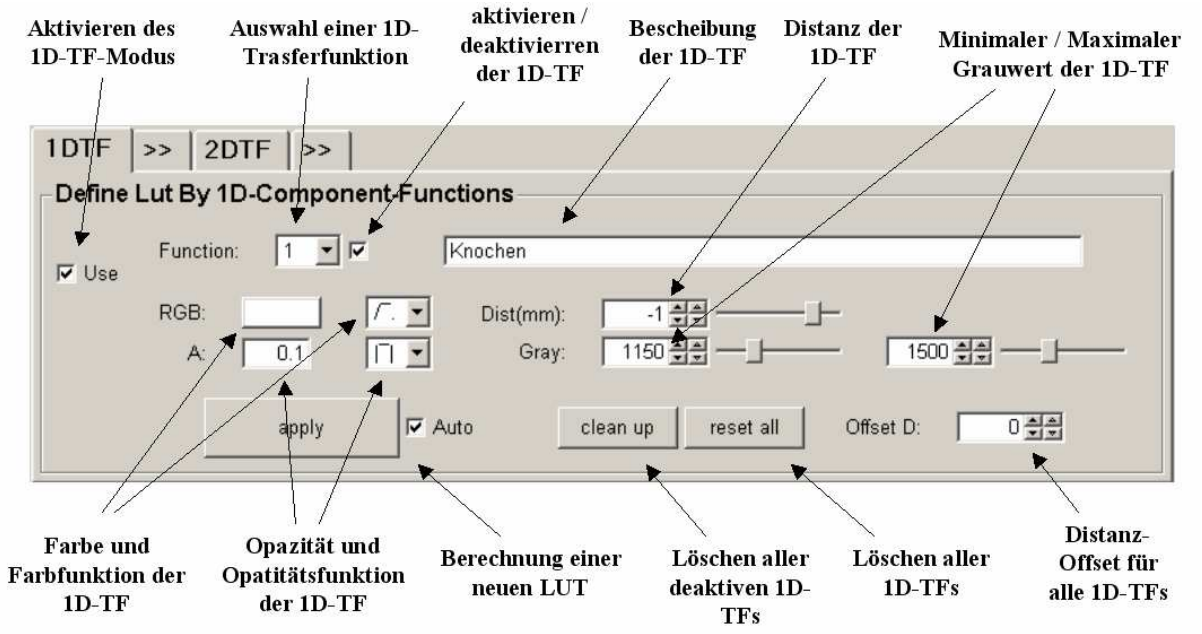
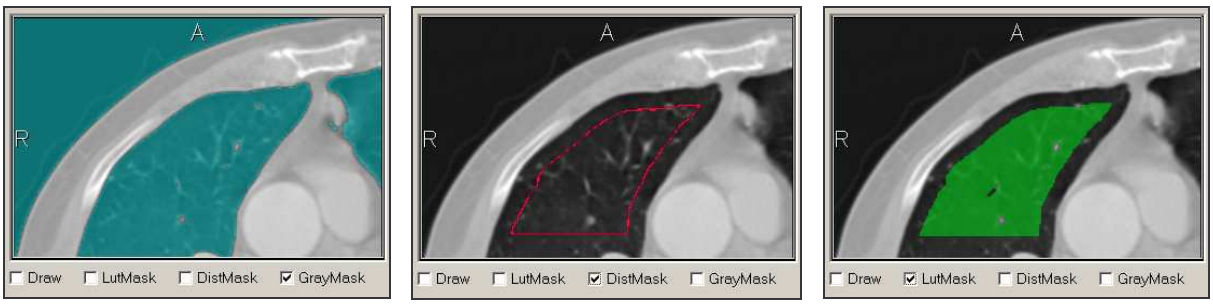


Abbildung 5-9: Parameter für die Definition von LUTs durch Interpolation zwischen 1D-TFs.

Bei gesetztem „Auto-Apply"-Toggle wird nach jeder Änderung eines Funktionsparameters eine aktuelle LUT, und falls aktiviert, ein aktuelles DVR berechnet. Falls dieser Toggle deaktiviert ist (auf langsamen Rechnern sinnvoll) kann die LUT mittels des „Apply"-Buttons aktualisiert werden. Dadurch finden weniger Aktualisierungen von LUT und DVR statt, was den Berechnungsaufwand erheblich reduziert.

Unterstützung beim Setzten von Distanzwert und Intensitätsintervall von 1D-TFs wird durch die Überlagerung von Schichtbilddarstellungen des verwendeten Grauwertdatensatzes mit speziellen Masken gegeben (Abbildung 5-10). Diese Schichtbilddarstellung befindet sich zusammen mit einem Volumenrenderer unter der Registerkarte „VRend". Die Schichtbilder lassen sich mit verschiedenen Masken überlagern. Zur Verfügung stehen eine Grauwertmaske, die alle Grauwerte, auf die die aktuelle 1D-TF Einfluss nimmt, einfärbt, ein Distanzmaske, die Rückschlüsse auf die Distanz einer 1D-TF zulässt, sowie ein LUT-Maske, die alle Voxel, die von der aktuellen LUT auf Alphawerte größer 0 abgebildet werden, hervorhebt.



(a) Grauwertmaske: Färbung aller Voxe, deren Intensität sich im Intensitätsintervall der aktuellen 1D-TF befindet.

(b) Distanzmaske: Färbung aller Voxel, die die Distanz der aktuellen 1D-TF aufweisen.

(c) LUT-Maske: Färbung aller Voxel, die von der resultierenden LUT auf Alpha-Werte größer 0 abgebildet werden.

Abbildung 5-10: Überlagerung von Schichtbildern mit speziellen Masken, als Unterstützung für das Setzen von Parametern.

Darüber hinaus erfolgt eine Darstellung der ausgewählten 1D-TF im HistogramViewer. 1D-TFs werden dabei als horizontale Linien repräsentiert, mit denen interagiert werden kann.

Durch vertikales Verschieben einer Linie wird der Distanzwert der zugehörigen 1D-TF manipuliert. Durch horizontales Verschieben der Kontrollpunkte an den beiden Linienenden wird das Intensitätsintervall der 1D-TF festgelegt.

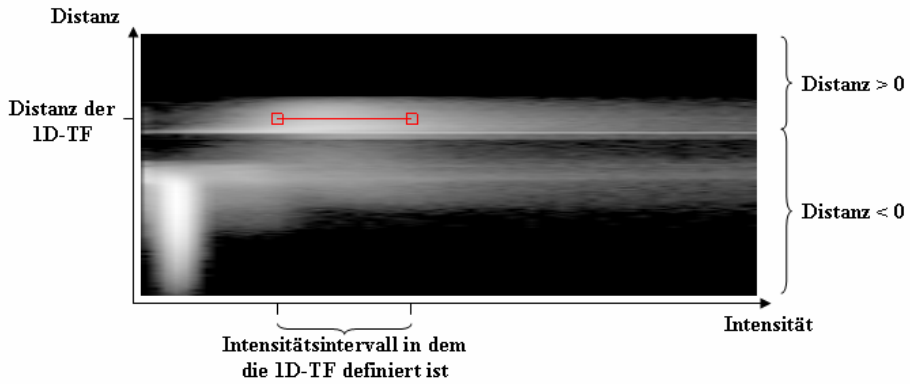


Abbildung 5-11: 2D-Histogramm, erzeugt aus Grauwertdatensatz und Distance-Map, dient als Orientierung für das Platzieren eines Interaktionselementes zum Setzen von Parametern. Die Intensität eines Pixels (Bin) des 2D-Histogramms repräsentiert die Anzahl von Voxeln mit der entsprechenden Intensität und Distanz.

Definition von LUTs durch 2D-Komponentenfunktionen

Weiterhin wurde die Definition von 2D-TFs durch Überlagerung von 2D-Komponentenfunktionen ermöglicht. Die Komponentenfunktionen legen dabei Eigenschaften von 2D-TFs innerhalb bestimmter Grauwert- und Distanzintervalle fest. Die Definition der Komponentenfunktionen erfolgt über nahezu dieselben Parameter wie sie bereits für 1D-TFs vorgestellt wurden. Unterschiede bestehen bei den Funktionstypen von Farb- und Opazitätsfunktion (sind zweidimensional) sowie in der Angabe eines Distanzintervalls anstatt eines einzelnen Distanzwertes. Mit Hilfe des Distanzintervalls wird festgelegt innerhalb welches Distanzbereichs sich eine Komponentenfunktion auswirkt.

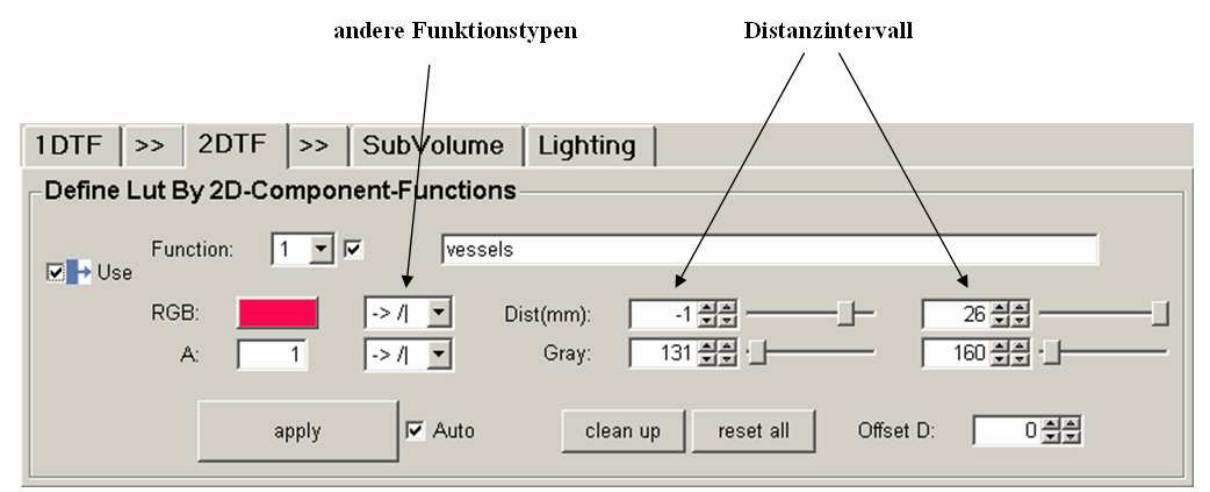


Abbildung 5-12: Die Registerkarte 2D-TFs.

Bei der Definition von Komponentenfunktionen können ebenfalls die bereits vorgestellten Schichtbildmasken verwendet werden. Außerdem erfolgt eine Darstellung im Histogramm-Viewer (Abbildung 4-16). Eine 2D-Komponentenfunktion wird hier als Recheck dargestellt,

an dessen Ecken sich Punkte befinden, die mit der Maus verschoben werden können. Damit wird Einfluss auf das Intensitäts- und Distanzintervall der aktuellen 2D-Komponentenfunktion genommen.

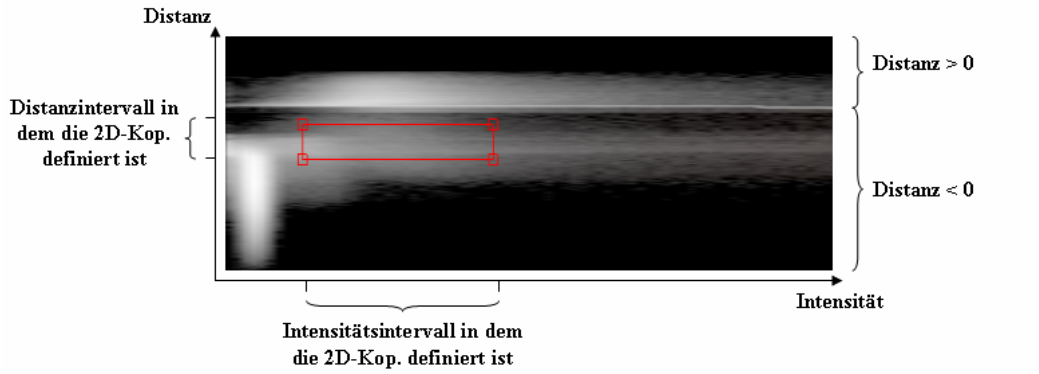


Abbildung 5-13: 2D-Histogramm dient als Orientierung für das Platzieren eines Interaktionselementes zum Setzen von Parametern der 2D-Komponentenfunktionen.

Laden von Presets

Eine weitere Möglichkeit LUTs zu erzeugen, besteht in dem Laden von Presets. Presets sind zu Gruppen ähnlicher Visualisierungsziele zusammengefasst. Diese können unter der Registerkarte „I/O“ geladen werden. Die Auswahl eines konkreten Presets erfolgt anschließend in der Preset-Box (Abbildung 5-14).

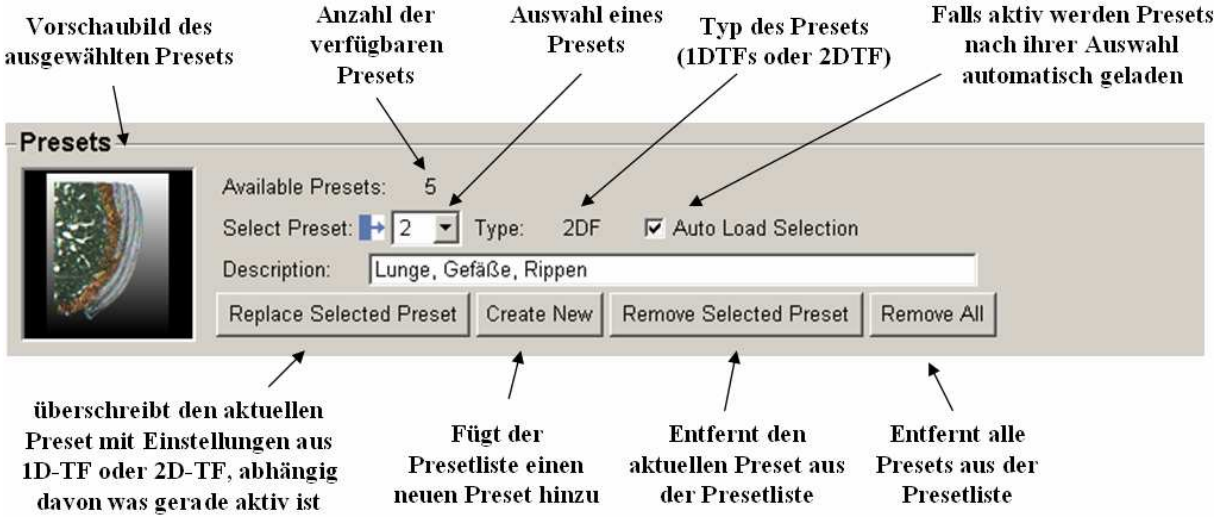


Abbildung 5-14: Preset-Box, Laden und Speichern von Presets.

Speichern von Presets

Mit Hilfe der Buttons „Replace Selectet Preset“ und „Create New“ können der aktuellen Presetliste neue Presets hinzugefügt werden. In beiden Fällen wird ein Preset mit den unter 1D-TF oder 2D-TF (je nachdem welcher Modus aktiv ist) getroffenen Einstellungen erzeugt. Dabei werden alle 1D-TF / 2D-Komponentenfunktionen, die deaktiviert sind, ignoriert. Damit Änderungen in Presetlisten auch zu einem späteren Zeitpunkt zur Verfügung stehen, können diese unter der Registerkarte „IO“ à „Save“ abgespeichert werden.

Für die Arbeit mit Presets ist es wünschenswert, wenn für jeden Preset ein Vorschaubild zur Verfügung steht. Die Vorschaubilder aller Presets einer Presetliste sind in Form eines Schichtdatensatzes unter dem Namen der zugehörigen Presetliste abgelegt. Nach der Veränderung einer Presetliste sollten die zugehörigen Vorschaubilder aktualisiert werden. Dies kann mit Hilfe des „Get Snapshot"-Buttons sowie der Registerkarte „I/O" à „Save Thumbnails" erfolgen.

Der integrierte Volumenrenderer

Um während des Definitionsprozesses der 2D-TFs auf DVRs mit der aktuellen 2D-LUT zurückgreifen zu können, wurde ein Volumenrenderer in den Editor integriert. Dieser kann unter der Registerkarte „VRend" geöffnet werden. Sobald eine gültige 2D-LUT zur Verfügung steht, wird der geladene Volumendatensatz unter Anwendung dieser LUT vom Volumenrenderer dargestellt. Unter den Registerkarten „SubVolume" und „Lighting" können Einstellungen des Renderers (Clipping und Beleuchtung) vorgenommen werden.

Einstellungen für die LUT-Generierung

Unter der Registerkarte „Properties" lassen sich Eigenschaften der zu erzeugenden LUTs genauer spezifizieren. Festgelegt werden kann hier Größe und Datentyp der LUT, die Methode mit der die LUT diskretisiert wird sowie Schnittbehandlungen von 2D-Komponentenfunktionen.

5.5 Zusammenfassung

Mit Hilfe des bereitgestellten Editors lassen sich DTFs für das DVR von Volumendaten erzeugen. Der dabei erforderliche Interaktionsaufwand ist als angemessen einzustufen. Die Zeit, die für die Erzeugung einer gewünschten TF benötigt wird, kann je nach Einarbeitung in die Editorfunktionen weniger als eine Minute bis zu mehreren Minuten betragen. Für die Definition von Parametern der TFs werden verschiedene Unterstützungen wie Masken und Interaktionselemente angeboten. Der Definitionsprozess läuft iterativ ab. Nach dem Festlegen einer initialen TF können aus dem resultierenden DVR, einer Ansicht der aktuellen LUT sowie verschiedenen Schichtbildmasken Rückschlüsse auf mögliche Änderungen von Parametern geschlossen werden. Dieser Prozess wird so lange wiederholt, bis die gewünschten Visualisierungsergebnisse erzielt werden. Durch die Verwendung von Presets lassen sich auf sehr schnellem Wege Visualisierungen erzeugen. Beachtet werden muss hierbei, dass die Presets gegebenenfalls an die aktuellen Datensätze anzupassen sind. Diese kann relativ schnell durch das Verschieben von Intensitätsintervallen geschehen.

6 Ergebnisse / Anwendungsgebiete von DTFs

Die Anwendung DTFs kann durch verschiedene Visualisierungsziele motiviert sein. Besonders nützlich sind DTFs, wenn eine distanzabhängige Steuerung von Opazitäts- bzw. Farbfunktion erwünscht ist. Im Folgenden werden einige Visualisierungsmöglichkeiten, die sich aus der Anwendung von DTFs ergeben, anhand von Beispielen diskutiert.

6.1 Visualisierung äquidistanter Gewebeschichten

Ein wesentliches Anwendungsgebiet von DTFs liegt in der dreidimensionalen Visualisierung von Gewebeschichten. Von besonderem Interesse sind in diesem Zusammenhang DVR-Visualisierungen von Organschichten, die sich in einem konstanten Abstand zur Organoberfläche befinden. Durch ein gezieltes „Abschälen“ können Strukturen im Inneren von Organen dargestellt werden (anatomische Reformatierung). Dabei besteht ein ständiger Bezug zur Entfernung der dargestellten Schicht von der Organoberfläche. Auf diese Weise lassen sich 3D-Darstellungen äquidistanter opaker Schichten erzeugen. Auf der Oberfläche dieser Schichten sind kleinste Strukturen erkennbar. Durch ein Verschieben der Distanz der dargestellten Schicht ist die Exploration des gesamten Organs möglich. Dabei stehen Interaktionsmöglichkeiten wie Rotation, Zoom oder Clipping zur Verfügung.

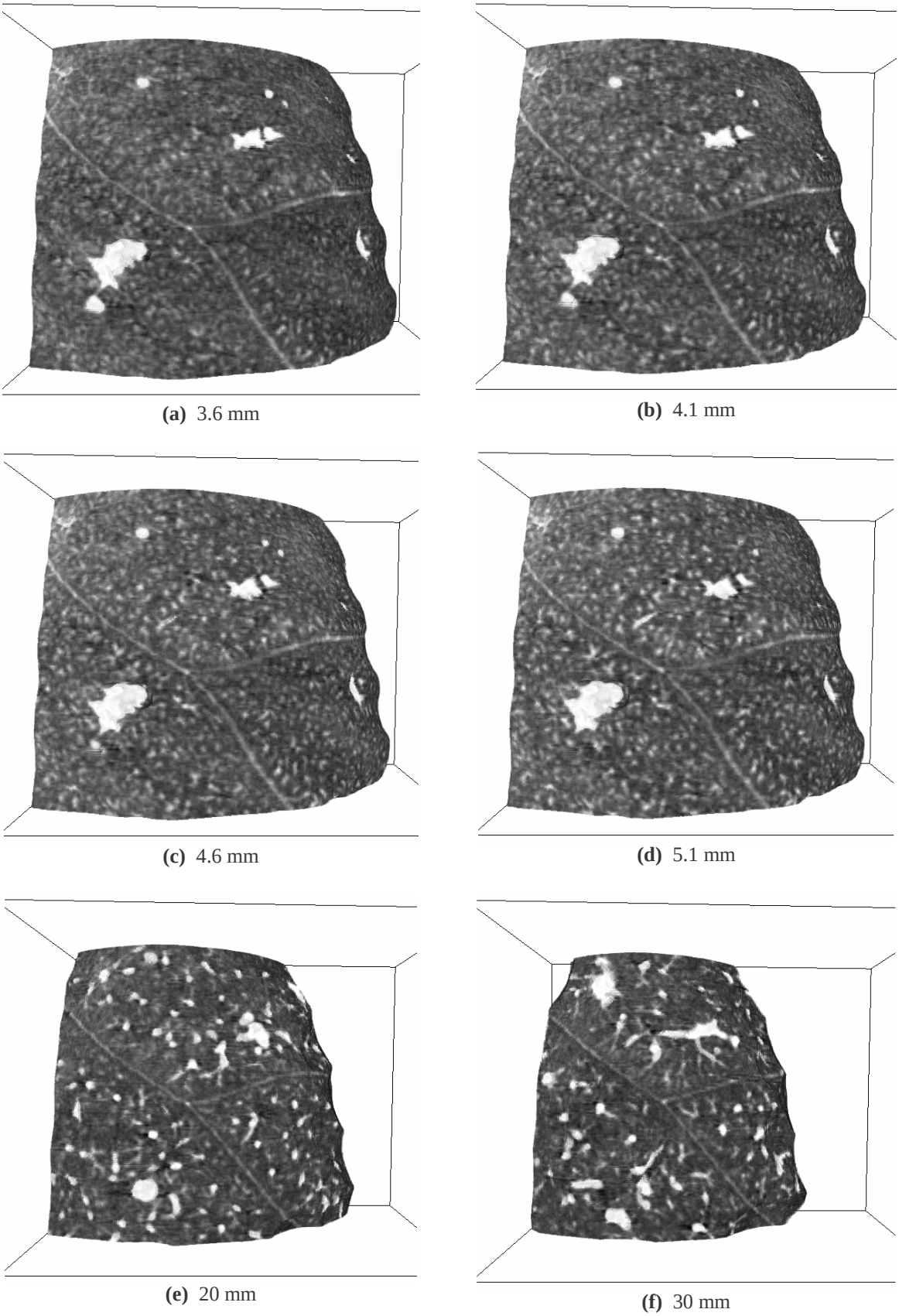
Diese Visualisierungstechnik vereinigt den größten Vorteil von Schichtbilddarstellungen, der in einer überlagerungsfreien Darstellung besteht, mit denen von 3D-Visualisierungen, die ein besseres räumliches Verständnis ermöglichen sowie zahlreiche Interaktionsmöglichkeiten bereitstellen. Sie kann deshalb unterstützend bei der Auswertung medizinischer Datensätze eingesetzt werden.

Im Folgenden werden einige Ergebnisse vorgestellt, die bei der Schichtvisualisierung von Lunge und Kortex erzielt wurden. Untersucht werden hierzu sowohl CT-Thoraxdatensätze als auch MRT-Aufnahmen des Kopfes.

Schichtvisualisierung der Lunge (CT)

Vor der Visualisierung sind zunächst geeignete DistanceMaps bereitzustellen. Auf das Vorgehen bei der Erzeugung von DistanceMaps unter Verwendung der Lungenoberflächen als Referenzflächen wurde in Abschnitt 4.4.1 eingegangen. Entscheidend ist dabei, dass die Lungenoberflächen gut aus den Daten segmentiert werden, da nur so eine korrekte Distanzberechnung erfolgen kann.

Für die Berechnung der 2D-LUTs wurde ein Preset gewählt, der eine schichtweise Darstellung der Lunge ermöglicht. Der Preset verfügt über einen Parameter, mit dessen Hilfe sich Einfluss auf die Distanz der dargestellten Schicht nehmen lässt (siehe Abschnitt 4.7.3). Dieser kann während der Visualisierung manipuliert werden. Es wurden sechs Darstellungen erzeugt, wobei die Distanz der sichtbaren Schicht verschoben wurde.



(a) 3.6 mm

(b) 4.1 mm

(c) 4.6 mm

(d) 5.1 mm

(e) 20 mm

(f) 30 mm

Abbildung 6-1: Schichtvisualisierung eines Teils des rechten Lungenflügels.

Unter der Voraussetzung, dass die Segmentierung der Lungenmaske automatisch erfolgt, ist der Interaktionsaufwand für das Erzeugen dieser Visualisierungen gering. Es wurde lediglich ein Preset zur opaken Lungendarstellung basierend auf CT-Datensätzen ausgewählt (Selektieren eines Vorschaubildes) und der Presetparameter zum Festlegen der darzustellenden Distanz (ein Skalarwert) eingestellt. Die Auflösung der Distanzdimension ist von dem Wertebereich der verwendeten DistanceMap sowie der Größe der eingesetzten LUT abhängig. Die DistanceMap aus diesem Beispiel hat einen Wertebereich von [-73 mm, 62 mm]. Außerdem wurden LUTs mit einer Größe von 256^2 verwendet. Daraus ergibt sich für die Distanzdimension eine Auflösung von (62-(-73) mm) / 256 = 0.53 mm. Diese ist als ausreichend anzusehen, da sie unter der Auflösung des verwendeten Bilddatensatzes (0.617×0.617×0.8 mm) liegt. Höhere Auflösungen der Distanzdimension können durch die Verwendung größerer LUTs erreicht werden. Bei einer LUT-Größe von 512^2 könnte in diesem Beispiel eine Auflösung von 0.26 mm, bei 1024^2-LUTs sogar 0.13 mm erreicht werden. 0.13 mm sind in jedem Falle ausreichend, da Schichten mit einem derart kleinem Distanzunterschied, keine optisch signifikanten Differenzen mehr aufweisen. Neue Grafikkarten (mindesten DirektX9 - fähig) unterstützen sehr große LUTs. Die Radeon 9600, 9800 Pro unterstützen beispielsweise Bittiefen von bis zu 11, was maximale LUT-Größen von 2048^2 erlaubt. Mit GeForceFX- Karten sind sogar Bittiefen von bis zu 12 möglich (LUT-Größe=4096^2).

Die Qualität der so erzeugten Visualisierungen liegt unter der von Schichtbildern. Dies ist auf die Neuabtastung der Daten beim DVR zurückzuführen (siehe Abschnitt 2.4.4). Es entstehen Renderingartefakte (siehe Abbildung 6-2), die sich jedoch durch die Verwendung hoher Abtastraten auf ein akzeptables Maß reduzieren lassen, sodass die Differenzierbarkeit von Strukturen auf den Oberflächen der dargestellten Schichten gewährleistet werden kann.

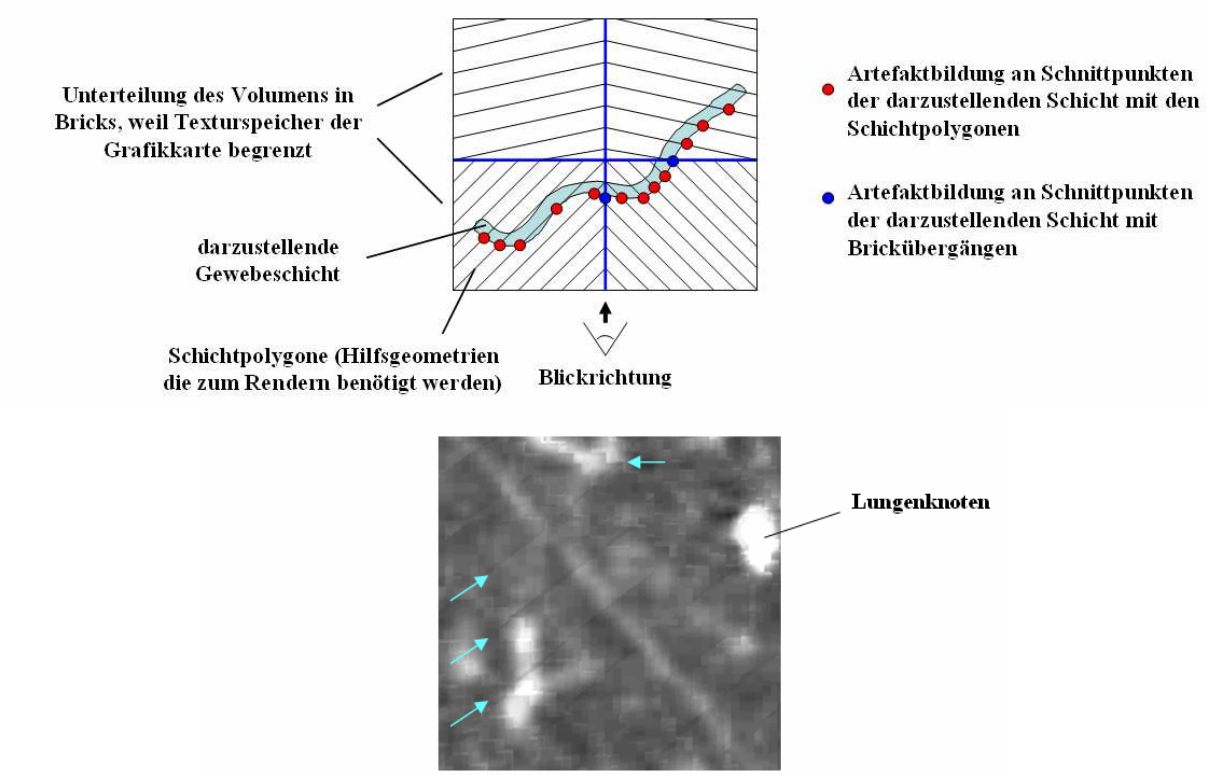


Abbildung 6-2: Renderingartefakte bei einer Visualisierung unter Verwendung von 3D-Texturen. Die im unteren Bildausschnitt dargestellte Visualisierung wurde bei starkem Zoom und mittlerer Abtastrate angefertigt. Es sind deutlich Artefakte an den Schnitten der dargestellten Schicht mit den Schichtpolygonen der Hilfsgeometrie zu erkennen.

Aufgrund der Neuabtastung des Volumens ist eine Verringerung der Bildschärfe gegenüber Schichtvisualisierungen festzustellen. Einem direkten optischen Vergleich können die 3D-Schichtvisualisierungen jedoch gut standhalten.

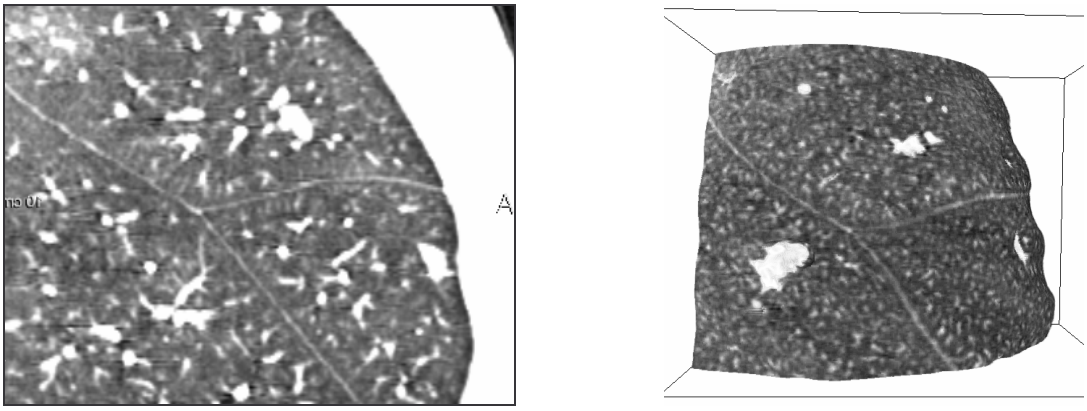
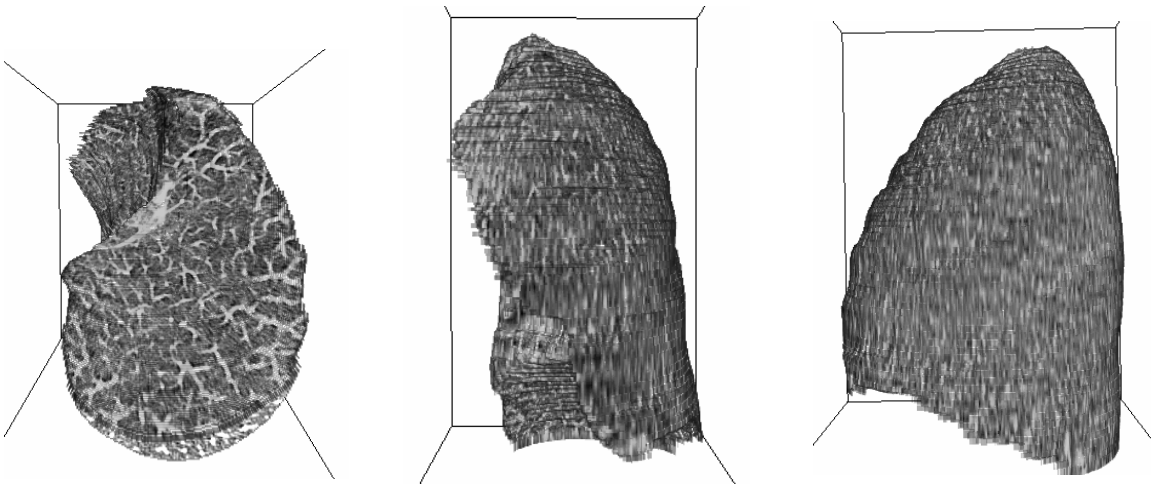


Abbildung 6-3: Vergleich Schichtbild und DVR. Das dargestellt Schichtbild stammt aus demselben Datensatz, mit dem das DVR erzeugt wurde.

Problematisch ist die Visualisierung von Datensätzen mit großem Schichtenabstand (>2 mm). Je mehr die Blickrichtung von einer axialen Darstellung abweicht, desto stärker ist mit Bildunschärfe zu rechnen. Dies ist darauf zurückzuführen, dass Volumendatensätze, die mit großem Schichtabstand erzeugt wurden, nur eine geringe Ortsauflösung in axialer Richtung aufweisen (Abbildung 6-4).



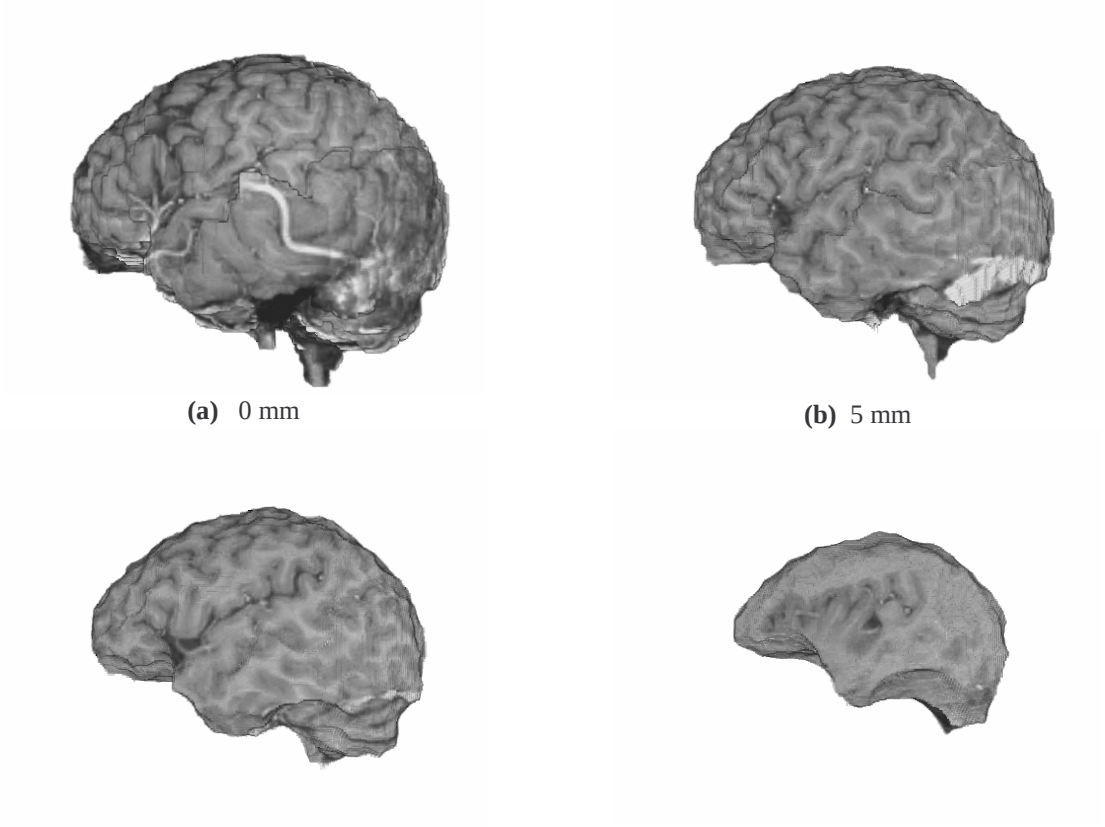
(a) axial **(b)** sagital **(c)** koronal

Abbildung 6-4: Probleme bei Datensätzen mit großem Schichtabstand. Für dieses Beispiel wurde ein Datensatz mit einem Schichtabstand von 4 mm verwendet. Während axiale Darstellungen aufgrund der hohen Ortsauflösung der Daten innerhalb einer Schicht scharf sind (a), ist in Darstellungen, bei denen die Blickrichtung stark von der Patientenachse abweicht, mit Bildunschärfe zu rechnen (b), (c).

Schichtvisualisierung des Kortex (MRT)

Bei der Anwendung von DTFs für das DVR von MRT-Daten ließen sich ebenfalls gute Ergebnisse erzielen. Im Folgenden werden Ergebnisse einer Schichtdarstellung des Kortex vor-

gestellt. Als Referenzfläche diente die Kortexoberfläche. Diese lässt sich mit Hilfe einer Wasserscheidentransformation (siehe Abschnitt 4.4, Kortexoberfläche) bestimmen.



(a) 0 mm **(b)** 5 mm

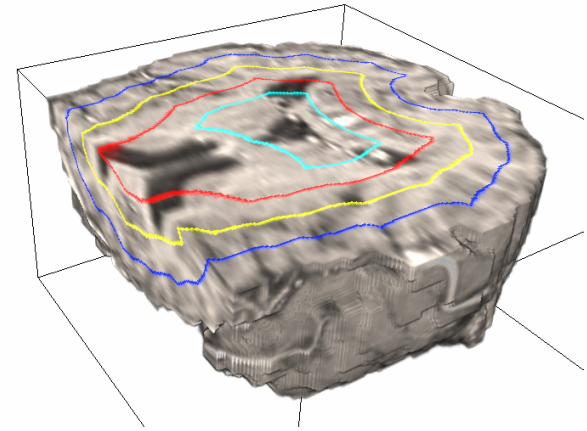
(c) 10 mm **(d)** 20 mm

Abbildung 6-5: Schichtvisualisierung des Kortex. Es wurde ein MRT-Datensatz mit einer Auflösung von 1×1×2 mm visualisiert. Als Referenzfläche wurde eine Maske der Kortexoberfläche verwendet. Es werden vier Schichten des Kortex mit unterschiedlicher Distanz zur Kortexoberflächen dargestellt.

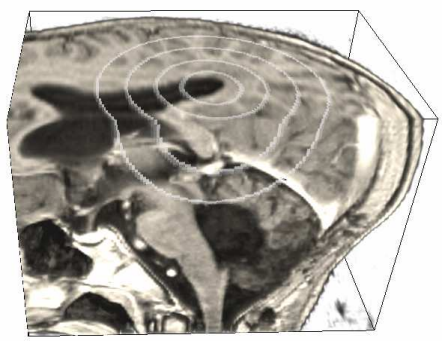
Prinzipiell eignen sich alle Strukturen des menschlichen Körpers, die in akzeptabler Qualität aufgelöst und segmentiert werden können, für eine 3D-Schichtvisualisierung. Weitere Anwendungen könnten deshalb im DVR von anderen Organen oder auch Knochen liegen.

6.2 Visualisierung von Distanzen

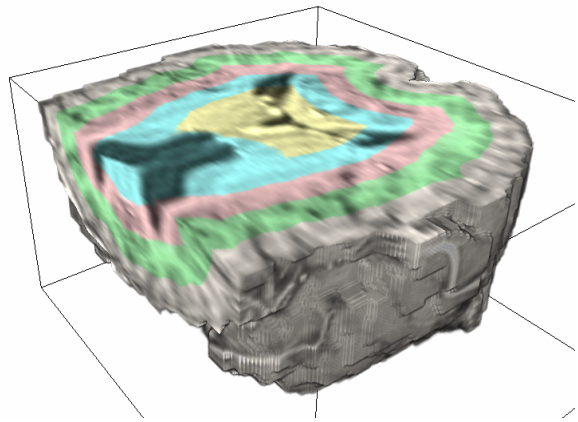
DTFs können dazu verwendet werden, Distanzinformationen in Visualisierungen einfließen zu lassen. Dadurch ist es auch ohne den Einsatz von Applikatoren möglich, Abstände zwischen Strukturen grob abzuschätzen. Eine wesentliche Rolle spielt dabei die Wahl von Referenzflächen, da sie den Ausgangspunkt für die Distanzcodierung darstellen. Abstände zu Referenzflächen können durch Unterschiede in Färbung oder Opazität visualisiert werden. Dabei ist darauf zu achten, dass die Erkennbarkeit von Strukturen so wenig wie möglich beeinträchtigt wird. Im Folgenden sind einige Möglichkeiten für Distanzvisualisierungen dargestellt.



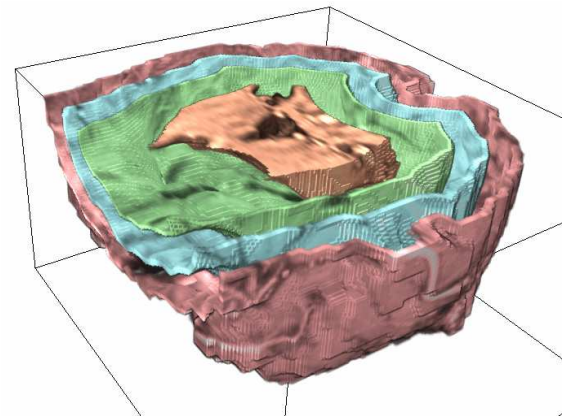
(a) Darstellen von Abständen zur Kortexoberfläche. 0.3 mm dünne gefärbte Schichten im Abstand von 10 mm (blau: 10 mm, gelb: 20 mm, rot: 30 mm, türkis: 40mm).



(b) Darstellen von Abständen zu einem selektierten Voxel. Dabei wird ebenfalls in 10 mm Schritten vorgegangen. Voxel an den entsprechenden Distanzen wurden grau eingefärbt. Mit demselben Verfahren lassen sich auch Abstände im Bezug zu Flächen oder anderen geometrischen Formen kennzeichnen.

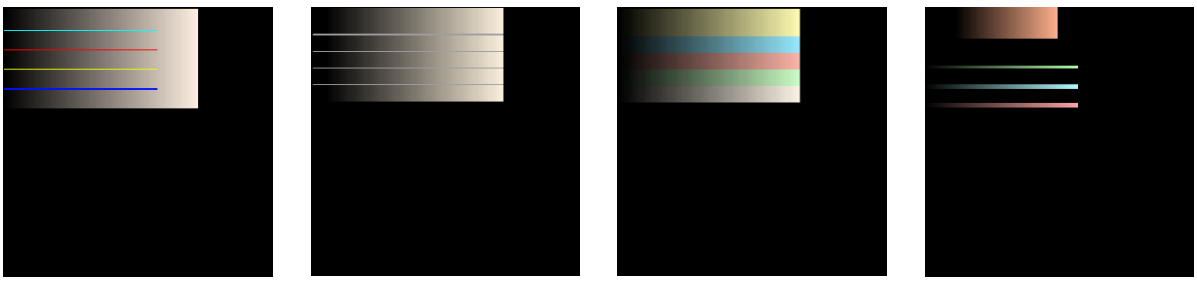


(c) Eine weitere Möglichkeit besteht darin, den gesamten Datensatz distanzabhängig einzufärben. Dabei werden verschiedenen Distanzintervallen unterschiedliche Farben zugewiesen.



(d) Darüber hinaus können, mit Hilfe einer distanzabhängigen Steuerung der Opazität, dünne Schichten in bestimmten Abständen dargestellt werden.

Abbildung 6-6: Distanzabhängige Visualisierung von Kortexschichten. Abstände zwischen einzelnen Strukturen lassen sich gut abschätzen. Für diese Beispiele wurde ein T1-gewichteter MRT-Datensatz mit einer Voxelgröße von 1×1×2 mm verwendet.



(a) **(b)** **(c)** **(d)**

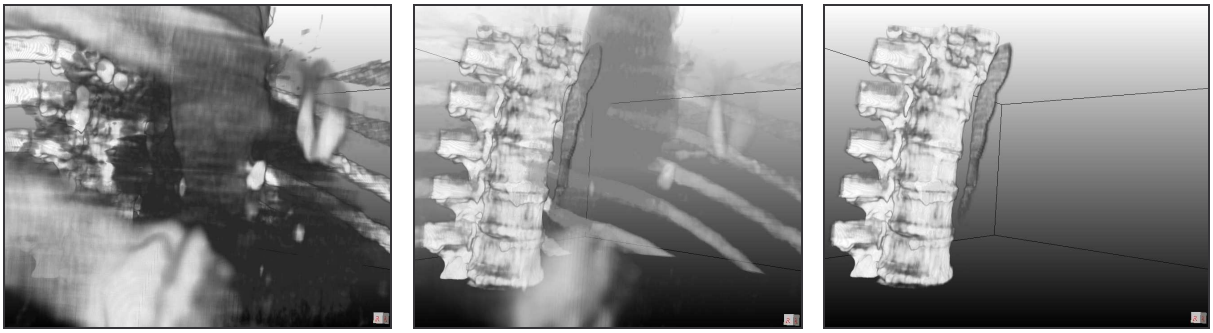
Abbildung 6-7: LUTs der Visualisierungen aus Abbildung 6-6.

Derartige Visualisierungen lassen sich beispielsweise für Lehrsysteme oder als Unterstützung bei der Exploration von Datensätzen einsetzen.

6.3 Distanzabhängiges Ausblenden von Strukturen

Für das DVR von medizinischen Datensätzen ist es mitunter wünschenswert, bestimmte Strukturen aus der Visualisierung auszuschließen. Wenn diese dieselben Intensitätswerte aufweisen wie diejenigen, die dargestellt werden sollen, ist kein Ausblenden mit rein intensitätsabhängigen TFs zu erreichen. Durch den zusätzlichen Freiheitsgrad distanzabhängiger TFs können Strukturen wesentlich gezielter ausgeblendet werden. Dies ist immer dann möglich, wenn sich die unerwünschten Strukturen mit Hilfe ihres Distanz- oder Intensitätsintervalls von den darzustellenden differenzieren lassen.

Ein Beispiel kann für das DVR der Wirbelsäule gegeben werden. Durch die Verwendung intensitätsabhängiger TFs lassen sich leicht alle Gewebe, deren Intensitäten unterhalb der von Knochen liegen, ausblenden. Problematisch ist jedoch, dass die Rippen sowie kontrastierte Blutgefäße meist große Teile der Wirbelsäule verdecken (siehe Abbildung 6-8, a). Dies kann durch den Einsatz von DTFs verhindert werden. Für dieses Beispiel ist es sinnvoll, die Mittellinie des Rückenmarks als Referenzfläche zu verwenden (Abschnitt 4.4). Alle Strukturen, die einen größeren Abstand als den halben Wirbeldurchmesser aufweisen, erhalten eine Opazität von 0. Das Ausblenden kann kontinuierlich innerhalb eines bestimmten Distanzintervalls oder abrupt ab einer konkreten Distanz erfolgen. Abbildung 6-8 (c) zeigt das vollständige Ausblenden aller Strukturen ab einer Distanz von 36 mm von der Mittellinie des Rückenmarks.



(a) Intensitätsabhängige Rampe im Bereich von 126 bis 426 HU

(b) ab einer Distanz von +36 mm wird zusätzlich eine Opazität von 0.1 zugewiesen

(c) ab einer Distanz von +36 mm werden alle Strukturen vollständig transparent gesetzt

Abbildung 6-8: Distanzabhängiges Ausblenden von Strukturen am Beispiel der Wirbelsäule. Auflösung des verwendeten CT-Datensatzes: 0.84×0.84×2 mm.

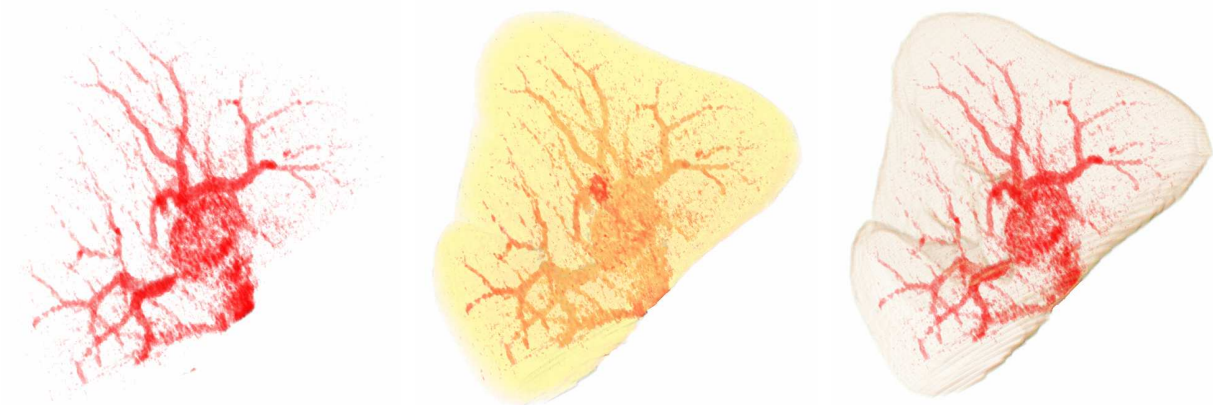
So kann leicht die Sicht auf die Wirbelkörper freigegeben werden. Darüber hinaus sind weitere Anwendungen, wie das distanzabhängige Ausblenden von Organen oder Muskeln denkbar. Außerdem ist mit Hilfe anatomisch unabhängiger Referenzflächen (siehe Abschnitt 4.4) das Erzeugen beliebiger Schnittansichten des Volumens möglich.

6.4 Distanzabhängige Darstellung von Kontextinformation

Oft ist es wünschenswert, Kontextinformationen in Visualisierungen einfließen zu lassen. Auf diese Weise wird eine bessere Orientierung ermöglicht. DTFs können bei der Darstellung von Kontextinformationen hilfreich sein, da sie eine distanzabhängige Steuerung der Abbildungseigenschaften erlauben. Damit lässt sich die Darstellung der Kontextinformation gezielt auf ein bestimmtes Diatanzintervall beschränken.

Bei der Visualisierung von Gefäßen innerhalb von Organen ist es beispielsweise sinnvoll, als Kontextinformation die Hülle des Organs transparent einzublenden. Dies kann nicht mit rein intensitätsabhängigen TFs erreicht werden, da sich die Organoberflächen nicht über ihre Intensität vom restlichen Organ separieren lässt. Alternativ besteht die Möglichkeit das gesamte Organ als Kontextinformation einzusetzen. Mit Hilfe von DTFs ist es möglich eine dünne Schicht an der Oberfläche des Organs einzublenden.

In Abbildung 6-9 (a) wird eine Visualisierung von Lebergefäßen ohne Kontextinformation dargestellt. Auffällig ist hierbei das der Bezug zur Organoberfläche fehlt. Auf die Ausdehnung des Organs kann lediglich über die Ausprägung des Gefäßbaumes geschlossen werden. Dadurch ist an den Randgebieten nur schwer einzuschätzen ob „weiße“ Gebiete nicht mehr zum Organ gehören oder sehr wenig Gefäße enthalten. Durch die Verwendung des gesamten Organs als Kontext kann diese Einschätzung wesentlich erleichtert werden, da die Konturen des Organs klar erkennbar sind (Abbildung 6-9, b). Ungünstig ist hierbei jedoch, dass die Gefäße von viel semitransparentem Gewebe überlagert werden und so undeutlich erscheinen. Durch die ausschließliche Verwendung der Organoberfläche als Kontextinformation kann dieses Problem weitgehend unterdrückt werde, da sich vor den Gefäßen nur eine dünne semitransparente Schicht befindet. Die Darstellung der Gefäße wird kaum beeinträchtigt (siehe Abbildung 6-9, c). Darüber hinaus wirkt die Darstellung wesentlich plastischer. Ein Ausschnitt der verwendete DTF ist in Abbildung 6-10 illustriert.



(a) Visualisierung der Lebergefäße ohne Kontext.

(b) Die ganze Leber wird als Kontextinformation verwendet.

(c) Ausschließlich die Organoberfläche wird als Kontext eingeblendet.

Abbildung 6-9: Visualisierung von Lebergefäßen.

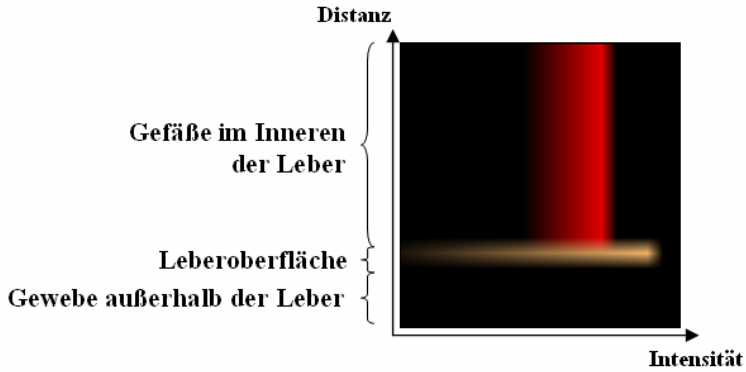


Abbildung 6-10: Ausschnitt aus der für die Erzeugung von Abbildung 6-9 (c) verwendeten DTF. Es werden drei Distanzintervalle mit unterschiedlichen Abbildungseigenschaften festegelegt.

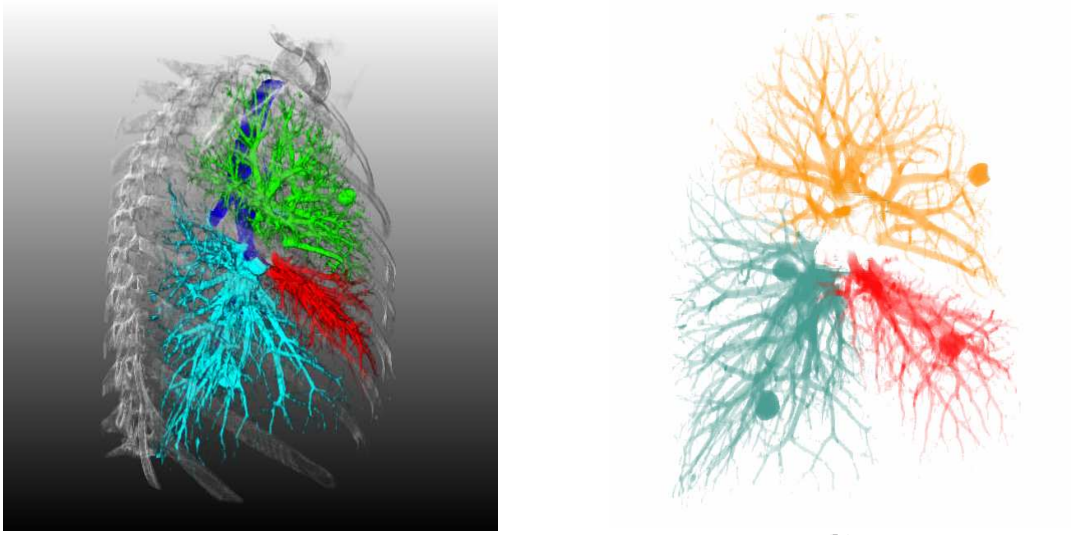
Ein weiteres Beispiel kann für die Visualisierung von Lungengefäßen gegeben werden (siehe Abbildung 6-11).

(a) Lungengefäße **(b)** Lungengefäße und Kontext

Abbildung 6-11: Visualisierung von Lungengefäßen mit und ohne Lungenoberfläche als Kontextinformation.

6.5 Lokale Transferfunktionen

Der im Zuge dieser Arbeit implementierte TF-Editor unterstützt die Definition zweidimensionaler TFs. Als Datendimensionen kann, wie beschrieben, neben der Intensität die Distanz zu einer Referenzfläche betrachtet werden. Darüber hinaus ist es möglich andere Datendimensionen einzusetzen. Experimentiert wurde in diesem Zusammenhang mit Segmentierungsmasken, die den Volumendatensatz in verschieden Bereiche unterteilen. Für diese Bereiche können anschließend unterschiedliche 1D-TFs festgelegt werden. Dies entspricht dem Konzept von lokalen TFs. Untersucht wurden CT-Lungendatensätze, für die bereits Segmentierungen der einzelnen Lungenlappen sowie der Bronchien vorlagen (siehe Abbildung 6-12).



(a) **(b)**

Abbildung 6-12: Visualisierung des rechten Lungenflügels. Die drei Lungenlappen wurden vorsegmentiert und mit unterschiedlichen 1D-TFs dargestellt. In der linken Darstellung wurden zusätzlich die Rippen sowie die Bronchien (blau) hinzugefügt. In der rechten Darstellung sind ausschließlich die semitransparenten Lungengefäße zu sehen.

7 Zusammenfassung / Ausblick

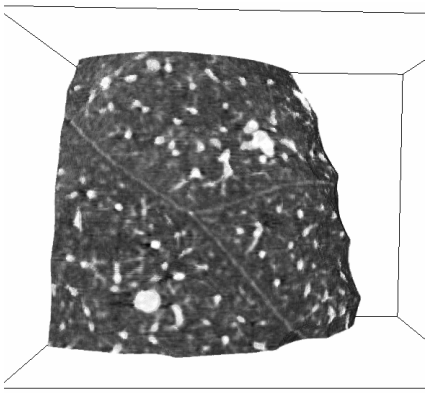
Bei dem direkten Rendering von Volumendaten kommt der Definition von TFs eine besondere Rolle zu. TFs bestimmen die Sichtbarkeit und Färbung von Strukturen und beeinflussen damit maßgeblich die Visualisierungsergebnisse. Sie bilden eine der wichtigsten Einflussmöglichkeiten, die einem Anwender zur Verfügung stehen, um seine Visualisierungsziele zu erreichen. Es ist deshalb von besonderem Interesse eine möglichst hohe Flexibilität bei der Definition von TFs bereitzustellen. Dies kann beispielsweise durch die Verwendung zusätzlicher Datendimensionen und die Bereitstellung geeigneter Repräsentationsformen geschehen.

Im Mittelpunkt dieser Arbeit steht die Einbindung von Positionsinformationen in den Visualisierungsprozess. Dies geschieht, indem Abstände zu Referenzflächen ausgewertet werden. Auf diese Weise wird eine distanzabhängige Steuerung von Farb- und Transparenzwerten ermöglicht. Da hierfür zwei Datendimensionen, die Intensität und die Distanz zu Referenzflächen, auszuwerten sind, ist die Definition 2D-TFs erforderlich. Zu diesem Zweck wurden aus bereits bewährten Konzepten zwei geeignete Repräsentationsformen für 2D-TFs abgeleitet, Interaktionskonzepte besprochen sowie Vorschläge für deren Implementierung diskutiert.

Aufbauend auf diesen Repräsentationen wurde ein Verfahren für die distanzabhängige Definition von TFs vorgestellt. Hierbei werden Distanzen in Bezug zu Referenzflächen ausgewertet. Die Bereitstellung von Referenzflächen spielt deshalb eine zentrale Rolle. Es werden verschiedene Methoden für deren Segmentierung erläutert. Im Mittelpunkt steht die Segmentierung von Organoberflächen. Diese sind beispielsweise für die Erzeugung von 3D-Schichtvisualisierungen von Organen als Referenzflächen von Interesse.

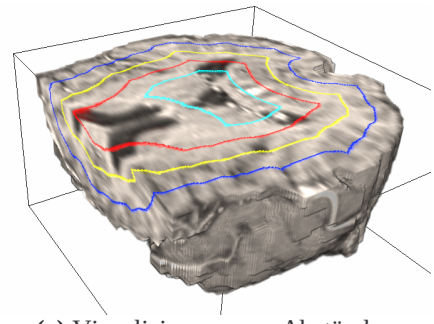
Für den medizinischen Einsatz ist die Verwendung von Presets interessant, da sich dadurch der Interaktionsaufwand für das Einstellen von Parametern erheblich reduzieren lässt. Aus diesem Grund wurde die Eignung distanzabhängiger TFs für den Einsatz von Presets untersucht. Bei der Visualisierung von CT-Aufnahmen konnten in verschiedenen Versuchen mit gleichen Presets und unterschiedlichen Datensätzen äquivalente Ergebnisse erzielt werden, ohne dass Adaptionen der Presets erforderlich waren. Bei der Verwendung von MRT-Aufnahmen zeigte sich, dass ohne eine Anpassung der Presets nur bedingt akzeptable Ergebnisse erreicht werden können. Die Anpassungen beschränken sich in den meisten Fälle auf eine Verschiebung / Skalierung von Intensitätsintervallen.

Anwendungsbereiche von DTFs liegen in der 3D-Schichtvisualisierung von Gewebestrukturen wie beispielsweise Organen, in dem Ausblenden unerwünschter Strukturen, in der Darstellung von Kontextinformationen sowie in der Visualisierung von Abständen (siehe Abbildung 7-1). Für diese Anwendungen wurden entsprechende Ergebnisse vorgestellt.

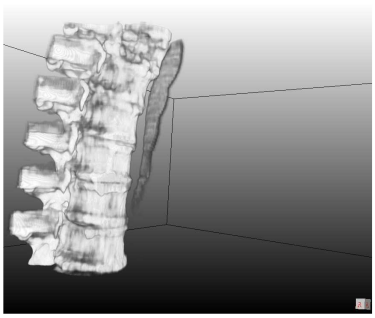


(a) 3D-Schichtvisualisierung von Organen

(b) Einblenden von Kontextinformationen



(c) Visualisierung von Abständen



(d) Distanzabhängiges Ausblenden von Strukturen

Abbildung 7-1: Anwendungsgebiete distanzabhängiger TFs.

Für weiterführende Arbeiten wäre beispielsweise der zusätzlichen Einsatz von Gradienteninformationen interessant. Eine Definition von TFs in Abhängigkeit von Intensität, Distanz und Gradientenstärke würde die Flexibilität weiter erhöhen und damit zusätzliche Visualisierungsmöglichkeiten erschließen. Beispielsweise wäre so ein distanzabhängiges Darstellen von Kanteninformationen möglich. Darüber hinaus könnten weitere Anwendungsgebiete von DTFs recherchiert und untersucht werden. Insbesondere der Einsatz anatomisch abgeleiteter und anatomisch unabhängiger Referenzflächen könnte Spielraum für zusätzliche Anwendungen schaffen. Interessant wären in diesem Zusammenhang beispielsweise das Aufspannen von Referenzflächen über die Rippen bzw. die Handknochen.

Abbildungsverzeichnis

Abkürzungsverzeichnis

1D	eindimensional
2D	zweidimensional
3D	dreidimensional
CT	Computertomographie
DTF	distanzabhängige Transferfunktion
DVR	direktes Volumenrendering
GVR	Gigavoxelrenderer, Volumenrenderer von MeVisLab
HU	Houndsfield-Units
LUT	Lookup – Tabelle
MeVis	Centrum für medizinische Diagnosesysteme und Visualisierung
MRT	Magnetresonanztomographie
PET	Positronenemissionstomographie
RGBA	Farbmodell basierend auf Rot-, Grün-, Blau- und Alphakomponente
TF	Transferfunktion

Literaturverzeichnis

1. Christoph Rezk-Salama: „*Volume Rendering Techniques for General Purpose Graphics Hardware*“, Dissertation, Universität Erlangen-Nürnberg, 2001.

2. H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. Avila, K. Martin, R. Machiraju, and J. Lee: „The transfer function bake-off”, In *Proc. IEEE Computer Graphics & Applications*, Band 21(3), S. 16-22, 2001.

3. Silvia Castro and Andreas König and Helwig Löffelmann and Eduard Gröller: „Transfer Function Specification for the Visualization of Medical Data”, Insitute of Computer Graphics, Vienne University of Technology, 1998.

4. Andreas H. König and Eduard M. Gröller: „Mastering Transfer Function Specification by using VolumePro Technology”, In *Proc. Spring Conference on Computer Graphics*, Band 17, S. 279–286, 2001.

5. J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes: *Computer Graphics: Principles and Practices*, 2. Aufl., AddisonWesley, 1990.

6. Heinz Morneborg: *Bildgebende Systeme für die medizinische Diagnostik*, Publicics MCD Verlag, 3. Aufl., 1995.

7. Arbeitsausschuss „Ärztliche Behandlung“ des Bundesausschusses der Ärzte und Krankenkassen: Positronen-Emissions-Tomographie (PET), Siegburg, 2002

8. Bjarne Stroustrup: *The C++ Programming Language*, Addison–Wesley, 2. Aufl., 1991.

9. Paolo Sabella: „A Rendering Algorithm for Visualizing 3D Scalar Fields”, *Computer Graphics*, Band 22(4), S. 51-55, 1988.

10. Min Chen, Arie E. Kaufman, Roni Yagel: *Volume Graphics*, Springer-Verlag, 2000.

11. M. Meißner, J. Huang, D. Bartz, K. Mueller, R. Crawfis: „A practical comparison of popular volume rendering algorithms”, In *Proc. 2000 Symp. on Volume Rendering*, S. 81-90, 2000.

12. Philippe Lacroute, Marc Levoy: „Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation”, In *Proc. SIGGRAPH '94*, S. 451-458, 1994.

13. Joe Kniss, Patrick McCormick, Allen McPherson, James Ahrens, Jamie Painter, Alan Keahey, Charles Hansen: „Interactive Texture-Based Volume Rendering for Large Data Sets”, In *IEEE Computer Graphics and Applications*, Band 21(4), S. 52-61, 2001.

14. Christof Rezk-Salama, Michael Scheuering: „Multitexturbasierte Volumenvisualisierung in der Medizin“, In *Bildverarbeitung für die Medizin 2001*, Springer, S. 137-141, 2001.

15. Ralf Ratering: *Texturbasiertes Volumen-Rendering Medizinischer Bilddaten*, am Institut für Informatik III, Rheinische Friedrich-Wilhelms-Universität Bonn, Fachbereich Computergrafik, 1998.

16. O. Wilson, A. Van Gelder, J. Wilhelms: „*Direct volume rendering via 3D textures*“, Tech. Rep. UCSC-CRL-94-19, University of California, Santa Cruz, 1994.

17. L. Westover: „Footprint Evaluation for Volume Rendering”, In *Computer Graphics*, Band 24(4), S. 367-376, 1990.

18. Lee Alan Westover: „*SPLATTING: A Parallel, Feed-Forward Volume Rendering Algorithm.*” PhD thesis, Department of Computer Science, University of North Carolina at Chapel Hill, 1991.

19. C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, T. Ertl: „Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-Textures and Multi-Stage Rasterization”, *Eurographics/SIGGRAPH Workshop on Graphics Hardware*, 2000.

20. James Kajiya, Brian Von Herzen: „Ray Tracing Volume Densities”, In *Computer Graphics*, Band 18(3), S. 165-174, 1984.

21. Jian Huang, Naeem Shareef, Roger Crawfis, P. Sadayappan, Klaus Mueller: „A Parallel Splatting Algorithm with Occlusion Culling”, 3rd Eurographics Workshop on Parallel Graphics and Visualization, Girona, Spain, 2000.

22. Rafael C. Gonzalez, Richard E. Woods: *Digital Image Processing*, 2. Aufl., Prentice Hall, New Jersey, 2002.

23. H. K. Hahn, Heinz-Otto Peitgen: „The Skull Stripping Problem in MRI Solved by a Single 3D Watershed Transform”, In *Proc. MICCAI*, LNCS 1935, S. 134-143, Springer, Berlin, 2000.

24. A.M. Alyassin and G.B. Avinash: „Semi-Automatic Bone Removal Technique from CTA Angiography Data”, In *Proc. of SPIE Medical Imaging Conference*, 2001.

25. Alan Watt: *3D-Computergrafik*, 3.Auflage, Pearson Studium, 2002.

26. G. L. Kindlmann: „*Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering*”, Master Thesis, Cornell University, 1999.

27. C. Rezk-Salama, P. Hastenreiter, J. Scherer, G. Greiner: „Automatic adjustment of transfer functions for direct volume rendering“, In *Proc. Vision, Modeling, and Visualization*, S. 357–364, 2000.

28. Shiaofen Fang, Tom Biddlecome and Mihran Tuceryan, „Image-Based Transfer Function Design for Data Exploration in Volume Visualization", In *Proc. IEEE Visualization*, S. 319-326, 1998.

29. Chandrajit L. Bajaj, Valerio Pascucci, Daniel R. Schikore: „The Contour Spectrum", In *IEEE Visualization*, S. 167-175, 1997.

30. J.M. Kniss, S. Premoze, M. Ikits, A. E. Lefohn, C.D. Hansen, E. Praun: „Gaussian Transfer Functions for Multi-Field Volume Visualization", In *IEEE Visualization*, S. 497-504, 2003.

31. Regina Helena Weber: „*Einfluss der Scanparameter auf das Abbildungsergebnis bei der CT-Angiographie*", Dissertation, Universität München, 2002.

32. J. Kniss, Gordon Kindlmann, Charles Hansen: „Multi-Dimensional Transfer Functions for Interactive Volume Rendering", In *IEEE Transactions on Visualization and Computer Graphics*, Band 8(3), S. 270-285, 2002.

33. Marc Levoy: „Display of Surfaces from Volume Data", In *IEEE Computer Graphics and Applications*, Band 8(3), S. 29-37, 1988.

34. Gordon Kindlmann, James W. Durkin: „Semi-automatic generation of transfer functions for direct volume rendering", In *Proc. of IEEE Volume Visualization*, S. 79-86, 1998.

35. Robert Sedgewick: *Algorithmen in C*, Addison-Wesley, 1. Aufl., 1992.

36. A. M. Alyassin: „Automatic Transfer Function Generation for Volume Rendering of High Resolution X-ray 3D Digital Mammography Images", SPIE Medical Imaging, San Diego, In *Visualization, Image-Guided Procedures and Display*, 2002.

37. J. Marks, B. Andalman, P.A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, S. Shieber: „Design galleries: a general approach to setting parameters for computer graphics and animation", In *Proc. of Siggraph' 97*, S. 389-400.

38. E. C. LaMar, B. Hamann, and K. I. Joy: „Multiresolution Techniques for Interactive Texture-Based Volume Visualization", In *IEEE Visualization*, S. 355–362, 1999.

39. J. Zhou, A. Döring, K. Toennies: „*Distance transfer function based Rendering*", Technical Report TR-ISGBV-04-01, Institute for Simulation and Graphics, University of Magdeburg, Germany, 2004.

40. Taosong He, Lichan Hong, Arie Kaufman, Hanspeter Pfister: „Generation of transfer functions with stochastic search techniques", In *Proc. of IEEE Visualization*, S. 227-234, 1996.

41. Fan-Yin Tzeng, Eric B. Lum, Kwan-Liu Ma: „A Novel Interface for Higher-Dimensional Classification of Volume Data", In *Proc. IEEE Visualization*, S. 505-512, 2003.

42. C. Rezk-Salama, P. Hastreiter, J. Scherer, G. Greiner: „Automatic adjustment of transfer functions for direct volume ren-dering“, In *Proc. Vision, Modeling, and Visualization*, S. 357–364, 2000.

43. Will Schroeder, Chandrajit Bajaj, Gordon Kindlmann, Hanspeter Pfister: „The Transfer Function Bake-Off“, In *IEEE Visualization*, S. 523-526, 2000.

44. Dieter Hönigmann, Johannes Ruisz, Christoph Haider: „Adaptive design of a global opacity transfer function for direct volume rendering of ultrasound data”, In *Proc. IEEE Visualization 2003*.

45. Gordon Kindlmann, Ross Whitaker,Tolga Tasdizen, Torsten Möller: „Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications “, In *Proc. IEEE Visualization 2003*.

46. T. J. Jankun-Kelly, K.-L. Ma: „A study of transfer function generation for time-varying volume data”, In *Proc. of Volume Graphics 2001 Workshop*, S. 51-65, 2001.

47. Simeon Potts, Torsten Möller: „Transfer Functions on a Logarithmic Scale for Volume Rendering”, In *Graphics Interface*, S. 57-63, 2004.

48. V. Srivastava, U. Chebrolu, K. Mueller: „Interactive transfer function modification for volume rendering using compressed sample runs”, In *Computer Graphics International*, S. 8-13, 2003.

49. Benjamin King: *Intuitive Einstellung von Transferfunktionen für das direkte Volume Rendering*, Diplomarbeit, Uni Hannover / Fachbereich Informatik, 2003.

50. B. Reitinger, Ch. Zach, A. Bornik, R. Beichel: „User-centric transfer function specification in augmented reality”, In *Journal of WSCG*, S. 355-362, 2004.

51. R. Mullick, N. Bryan, J. Butman, Confocal Volume Rendering: Fast Segmentation-Free Visualization of Internal Structures. In: *Medical Imaging 2000 - Image Processing*, SPIE, 2000.

52. Gabriele Lohmann: „Volumetric Image Analysis”, Wiley Teubner, 1998.

Anhang A Aufbau des DistLutEdior-Macros