

Realistische 3D-Visualisierung multiattributierter und multiparametrischer Volumendaten

Dissertation
zur Erlangung des Grades eines Doktors der Naturwissenschaften
am Fachbereich Informatik der Universität Hamburg

vorgelegt von

Ulf Tiede

aus Hamburg

Hamburg 1999

Genehmigt vom Fachbereich Informatik der Universität Hamburg auf Antrag von:

Prof. Dr. Karl Heinz Höhne, Institut für Mathematik und Datenverarbeitung in der Medizin (IMDM), Fachbereich Medizin, Universität Hamburg (1. Gutachter)

Prof. Dr.-Ing. em. Hermann Flessner, ehemals Arbeitsbereich Angewandte Informatik in Naturwissenschaft und Technik (ANT), Fachbereich Informatik, Universität Hamburg (2. Gutachter)

Prof. Dr.-Ing. H. Siegfried Stiehl, Arbeitsbereich Kognitive Systeme (KOGS), Fachbereich Informatik, Universität Hamburg (3. Gutachter)

Prof. Dr.-Ing. Bärbel Mertsching, Arbeitsbereich Informatikmethoden für Mikroelektronikanwendungen (IMA), Fachbereich Informatik, Universität Hamburg (Vorsitzende des Prüfungsausschusses)

Tag der Disputation: 9. März 1999

Kurzfassung

In den vergangenen Jahren sind zahlreiche Verfahren für die dreidimensionale (3D) Visualisierung medizinischer Volumendaten entwickelt worden. Viele Anwendungen erfordern eine Segmentierung der Daten, damit Strukturen, die den Blick auf die interessierenden Objekte verdecken, bei der Darstellung ausgeblendet werden können. Für tomographische bildgebende Verfahren, die den Partial-Volumen-Effekt aufweisen, ist die Schwellwertsegmentation weit verbreitet. Bisher existieren jedoch keine Methoden für eine realistische Rekonstruktion der Objektoberfläche aus den in ihrer räumlichen Auflösung begrenzten Attributvoxeln, die die Objektzugehörigkeit beschreiben. Dies führt insbesondere bei starken Vergrößerungen zu „voxeligen“ Oberflächen. Ebenso sind bisher keine Methoden und Datenstrukturen veröffentlicht worden, die eine Kombination verschiedener Visualisierungstechniken wie Oberflächendarstellung, Maximum-Intensitäts-Projektion, Volume-Rendering oder die Darstellung interner Strukturen auf Schnittebenen sowie die Visualisierung multiattribuierter Objekte in einer einzelnen Ansicht ermöglichen, um die für einen bestimmten Zweck oder ein bestimmtes Objekt jeweils günstigste Darstellungsmethode auswählen zu können.

In dieser Arbeit wird ein neues Verfahren entwickelt, das auf der Basis der bekannten Strahlverfolgungsmethode für Volumendaten eine realistische Rekonstruktion der Objektoberflächen ermöglicht. Dies wird durch die Bestimmung der Attributwerte im Subvoxelbereich für jede reellwertige Abtastposition entlang einer geometrisch exakten Abtastung des Datenvolumens erreicht. Dazu wird die Nachbarschaft der Abtastposition untersucht und aus den verschiedenen angrenzenden Attributvoxeln dasjenige Objekt ausgewählt, das unter Berücksichtigung der ursprünglichen Schwellwertsegmentation am besten paßt. Die zugehörige Objektgrenze wird durch Interpolation entlang des Abtaststrahls berechnet. Durch die Lokalisation der Oberfläche im Subvoxelbereich wird wiederum eine verbesserte Abschätzung der Oberflächennormalen aus den Grauwertgradienten erzielt.

Mit Hilfe frei definierbarer Schnittebenen wird eine regionenbasierte Visualisierung realisiert, die beiderseits der Schnittebenen nicht nur ein selektives Entfernen oder Einfärben von Objekten ermöglicht, sondern auch die freie Kombination verschiedenartiger Darstellungsmethoden. Außerdem wird ein neues Konzept für die objektbezogene Visualisierung multiattribuierter Volumina, d. h. unter verschiedenen Gesichtspunkten segmentierte Daten, entwickelt, um Objekte aus verschiedenen Wissensdomänen beliebig logisch miteinander verknüpfen und darstellen zu können. Dieses Konzept ermöglicht auch die Darstellung multiparametrischer Datensätze wie z. B. die registrierten anatomischen Farbschnittbilder und Computer-Tomographie-Aufnahmen aus dem Visible-Human-Projekt. Für die Steuerung des Systems wird eine leicht erweiterbare Beschreibungssprache entwickelt, deren Skriptfähigkeit die Realisation komplexer Anwendungen und die Erzeugung von Animationen mit Hilfe einfach editierbarer Parameterdateien ermöglicht.

Die Qualität der neu entwickelten Methoden wird an einer Reihe von Beispielen demonstriert und erzeugt im Fall der anatomischen Schnittbilder aus dem Visible-Human-Projekt nahezu photorealistische Abbildungen. Die Multiattributierbarkeit wird anhand eines dreidimensionalen Hirnatlasses auf der Basis eines MR-Volumens mit den Wissensdomänen Morphologie, Funktionsareale und Blutversorgung gezeigt. Dabei lassen sich z. B. alle diejenigen Volumenelemente (Voxel) einfärben, die zu einer bestimmten Hirnwinding gehören, an einer bestimmten motorischen Funktion beteiligt sind und von einem bestimmten Blutgefäß versorgt werden. Für die Anwendung in der Strahlentherapie werden multiple Objekte aus den Domänen Morphologie, Dosisverteilung, Zielvolumina und Risikoorgane gleichzeitig visualisiert.

Die beschriebenen Methoden bilden den Kern des VOXEL-MAN-Visualisierungssystems und ermöglichen eine bisher nicht erreichte Flexibilität der 3D-Darstellung medizinischer Volumendaten. Das System wird in verschiedenen klinischen Anwendungen und in der Entwicklung von Ausbildungssystemen, wie dreidimensionalen anatomischen Atlanten, eingesetzt.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Zielsetzung	3
2	Stand der Forschung	6
2.1	Segmentation	6
2.2	Visualisierung mit der Strahlverfolgungsmethode	8
2.3	Oberflächenbasierte Visualisierung	9
2.3.1	Polygonoberflächen	10
2.3.2	Voxeloberflächen	10
2.4	Volumenbasierte Visualisierung	10
2.4.1	Berechnung der Oberflächennormalen aus den Grauwertgradienten	11
2.4.2	Der Marching-Cubes-Algorithmus	11
2.4.3	Segmentationsfreie Visualisierung	12
2.4.3.1	Das Volume-Rendering	12
2.4.3.2	Die Maximum-Intensitäts-Projektion (MIP)	13
2.4.3.3	Die simulierte Röntgenbilddarstellung	13
2.4.4	Die attributgesteuerte Oberflächen- und Volumendarstellung	14
3	Methode	17
3.1	Berechnung der Strahlgeometrie	17
3.2	Klassifikationsgesteuerte Bestimmung der Attributwerte im Subvoxelbereich	18
3.3	Lokalisation der Objektgrenze im Subvoxelbereich	20
3.4	Berechnung der Oberflächennormalen	23
3.5	Interpolation von Attributwerten	26
3.6	Oberflächenbestimmung für multiparametrische Daten	28
3.7	Darstellung kleiner Strukturen	29
3.8	Regionenbasierte Visualisierung	30
3.8.1	Definition von Regionen	31
3.8.2	Schneiden	33
3.8.3	Realisierung von „Spanten“	34
3.8.4	Schneiden entlang des Sehstrahls	35
3.8.5	Freie Auswahl des Darstellungsverfahrens	35
3.9	Texturierung der Objektoberfläche	36
3.10	Visualisierung multiattribuierter Volumina	38
3.11	Datenstruktur für die multiattributierte und multiparametrische Visualisierung	39
4	Implementation	44
4.1	Benutzerschnittstelle	44
4.2	Performanz	46
4.2.1	Zugriff auf Randbereiche des Datenvolumens	46
4.2.2	Bounding-Volumes	49
4.2.3	Generierung eines multiattribuerten Labelvolumens	52
5	Anwendung und Ergebnisse	54
5.1	Vergleich der Qualität der Oberflächendarstellung	54
5.1.1	Demonstration an einem künstlichen Objekt	54
5.1.2	Auswirkung bei starken Vergrößerungen	54

5.1.3	Qualitätsvergleich der berechneten Attributwerte	57
5.1.4	Visible-Human-Projekt	59
5.2	Eigenschaften der Visualisierungsverfahren in verschiedenen Anwendungen . .	64
5.2.1	Gefäßdarstellung	64
5.2.2	Simulation dynamischer Vorgänge	64
5.2.3	Multiparametrische Daten	67
5.2.3.1	Visualisierung multiparametrischer MR-Volumina	68
5.2.3.2	Kombinierte 3D-Darstellung von PET und MR	68
5.2.4	Entwicklung intelligenter dreidimensionaler anatomischer Atlanten . .	69
5.2.4.1	Präparation durch selektives Schneiden	72
5.2.4.2	Simulation eines stereotaktischen Eingriffs	73
5.2.4.3	Simulation einer „intelligenten“ Röntgenbilddarstellung . . .	73
5.2.4.4	Integration modellierter Gefäße	73
5.2.4.5	Verwendung von Texturen	76
5.2.4.6	Erzeugung von Lehrfilmen	76
5.2.5	Strahlentherapieplanung	78
6	Schlußfolgerungen	81
	Literaturverzeichnis	83
A	Bestimmung der Gradientengewichtung	91
B	Abbildungstransformation	94
C	Syntax der Beschreibungssprache	97
C.1	General	97
C.2	Resolution	99
C.3	Color	99
C.4	Dataset	100
C.5	Gradient	102
C.6	Table	103
C.7	Texture	104
C.8	Shading	106
C.9	Domain	107
C.10	Object	108
C.11	Composition	109
C.12	Scene	111
C.13	Planes	111
C.14	Areas	112
C.15	Bounding Volume	113
C.16	Camera	114
C.17	Background	116
C.18	Lightsource	117
C.19	User Interface Command Language (UICL)	117
C.20	Miscellaneous	118

Abbildungsverzeichnis

1	Beispiele tomographischer Schnittbilder	1
2	Beispiel einer 3D-Darstellung	2
3	Verfahrensschritte für die Volumenvisualisierung	7
4	Das Phong'sche Beleuchtungsmodell	9
5	Volume-Rendering	13
6	Maximum-Intensitäts-Projektion (MIP)	14
7	Künstliche Röntgenbilddarstellung	14
8	Klassisches „Ray-Casting“	16
9	Dilation der Attribute	18
10	Bestimmung des Oberflächenschnittpunktes	21
11	Überlappende Intensitätsbereiche	22
12	Abhängigkeit der Objektgrenzen von den Intensitätsbereichen	23
13	Filterkern für den Grauwertgradienten	25
14	Interpolation der Attributwerte	27
15	Oberflächenbestimmung für RGB-Volumendaten	29
16	Klassifikation des Visible-Human durch Ellipsoide im RGB-Farbraum	30
17	Darstellung kleiner Strukturen	31
18	Selektives Schneiden von Objekten	31
19	Geometrie einer Schnittebene	32
20	Erzeugung einer eindeutigen Regionenkennung	33
21	Einbettung von Schnittebenen	35
22	Regionenbasierte Visualisierung	36
23	Interpolation der Objektfarbe	37
24	Datenstruktur für die 3D-Visualisierung	40
25	Mengentheoretische Verknüpfung von Objekten	41
26	Datenstruktur „intelligenter“ Bilder	43
27	Komplexe Komposition verschiedener Darstellungsformen	44
28	Objektdefinition durch eine Beschreibungssprache	45
29	Datenzugriff am Volumenrand	46
30	Erweiterter Clipping-Algorithmus	48
31	Ray-Clipping an der Volumengrenze	50
32	Strahlbegrenzung durch Bounding-Volumes	51
33	Zusammenfassung der Attributvolumina	53
34	Darstellung einer 1 Voxel dünnen Spirale	55
35	Visible-Human: Vergrößerte Darstellung des Auges	56
36	Vergleich von Blutgefäßen in vergrößerter Darstellung	56
37	Rekonstruktion eines Hüftgelenks	57
38	Darstellung der Attributwerte im Subvoxelbereich	58
39	Visible-Human: Anatomisches Schnittbild	60
40	Visible-Human: Rekonstruktion des Kopfes	60
41	Visible-Human: Berechnung der Oberflächenfarbe	62
42	Visible-Human: Komposition aus Oberflächen, Schnitten und Spanten	63
43	Visible-Human: Schädel, Gefäß- und Muskelstruktur	63
44	Visible-Human: Gleichzeitige Darstellung verschiedener Modalitäten	64
45	Darstellung von Gefäßen	65
46	Endoskopische Gefäßdarstellung	65

47	Simulation von Blutfluß	67
48	Darstellung multiparametrischer MR-Daten	68
49	Kombinierte Visualisierung von PET und MR	69
50	„Intelligente“ Volumen-Repräsentation	70
51	VOXEL-MAN-Benutzeroberfläche	71
52	Darstellung von Spanten im anatomischen Modus	72
53	Präparation eines Fötus	72
54	Darstellung der Hirnwindungen	73
55	Simulation einer Stereotaxie	74
56	Simulation eines Röntgenbildes	74
57	Hybride Darstellung von Volumenobjekten und Polygonstrukturen	75
58	Texturierung der Hautoberfläche	76
59	Texturierung einer Mumie	77
60	Strahlentherapieplanung: Thorax	79
61	Strahlentherapieplanung: Kopf	80
62	Zerlegung einer Kugel in $3 \times 3 \times 3$ isotrope Voxel	91
63	Berechnung der Schnittmenge von Kreis und Quadrat	93
64	3D-Transformation	96

Tabellenverzeichnis

1	Gradientengewichtung	25
2	Rechenzeiten mit und ohne Bounding-Volume	51
3	Charakterisierung von Objekten durch Ellipsoide im RGB-Merkmalraum	61
4	Schwellwerte verschiedener Objekte eines MR-Datensatzes	71
5	Rechenzeiten	77

1 Einleitung

Mit der Entwicklung der tomographischen bildgebenden Verfahren in der Medizin, insbesondere der Computertomographie (CT), der Magnet-Resonanz-Tomographie (MR) und der Positronen-Emissions-Tomographie (PET), gab es erstmals die Möglichkeit, die in diesen Daten enthaltene dreidimensionale (3D) Struktur der Organe zu rekonstruieren und mit Hilfe der Computergraphik darzustellen. Unabhängig von den physikalischen Grundlagen und Besonderheiten der einzelnen Verfahren (siehe hierzu [Wegener *et al.* 1996; Hendrick 1993; Wienhard *et al.* 1989]) ist diesen Verfahren gemeinsam, daß sie im Gegensatz zur konventionellen Röntgentechnik überlagerungsfreie zweidimensionale (2D) Schnittbilder erzeugen. Typische Beispiele für tomographische Aufnahmen sind in Abbildung 1 gezeigt.

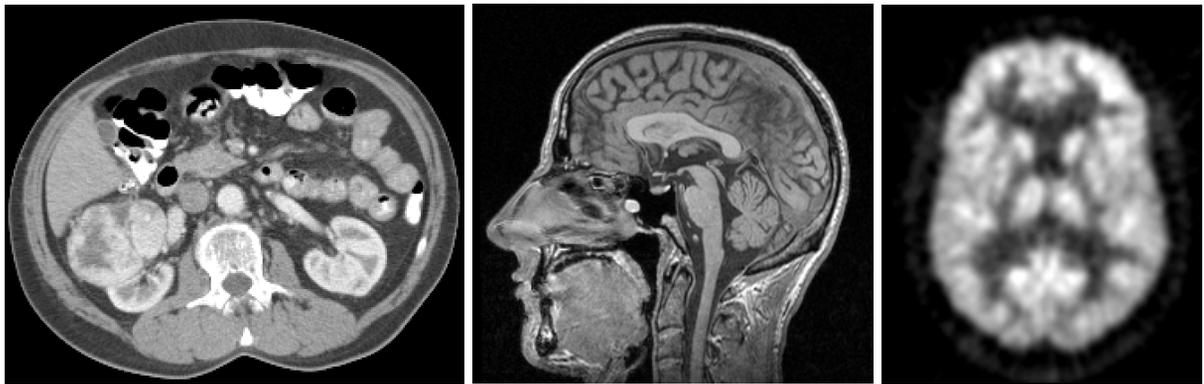


Abbildung 1: *Beispiele tomographischer Schnittbilder: Computertomogramm (CT) aus dem Bauchraum (links), sagittales Magnet-Resonanz-Tomogramm (MR) des Kopfes (Mitte), transversaler PET-Scan eines Gehirns (rechts).*

Die mentale Rekonstruktion dreidimensionaler Zusammenhänge und ihre Interpretation für Zwecke der Diagnostik und Operationsplanung ist komplex und erfordert u. a. jahrelange Erfahrung. Ziel der dreidimensionalen Rekonstruktion ist daher die möglichst realistische Darstellung der Oberflächen von Organen, die implizit in den Schnittbildern enthalten sind (Abbildung 2). Im Gegensatz zu den 2D-Tomogrammen, die direkt auf einem Bildschirm betrachtet werden können, erfordert die dreidimensionale Darstellung von Objekten aus einer räumlichen Schnittbildsequenz zwei Schritte:

1. Die Segmentation, d. h. die Zuordnung jedes Volumenelementes (engl. Voxel) zu einem im Bildvolumen enthaltenen Objekt.
2. Die Projektion der ausgewählten Strukturen bzw. Voxel auf eine zweidimensionale Bildebene entsprechend einer gewählten Blickrichtung und -position unter Berücksichtigung von Verdeckungen.

Die automatische Segmentation tomographischer Volumendaten ist ein nicht allgemein gelöstes Problem. Die in der vorliegenden Arbeit gezeigten Beispiele wurden mit Verfahren von [Bomans 1994], [Höhne, Hanson 1992] und [Schiemann *et al.* 1996] segmentiert. Nachfolgend wird die Segmentation als gegeben angenommen und nicht weiter behandelt.

Die zahlreichen bisher veröffentlichten Verfahren für die Visualisierung von Volumendaten lassen sich in zwei Klassen einteilen:

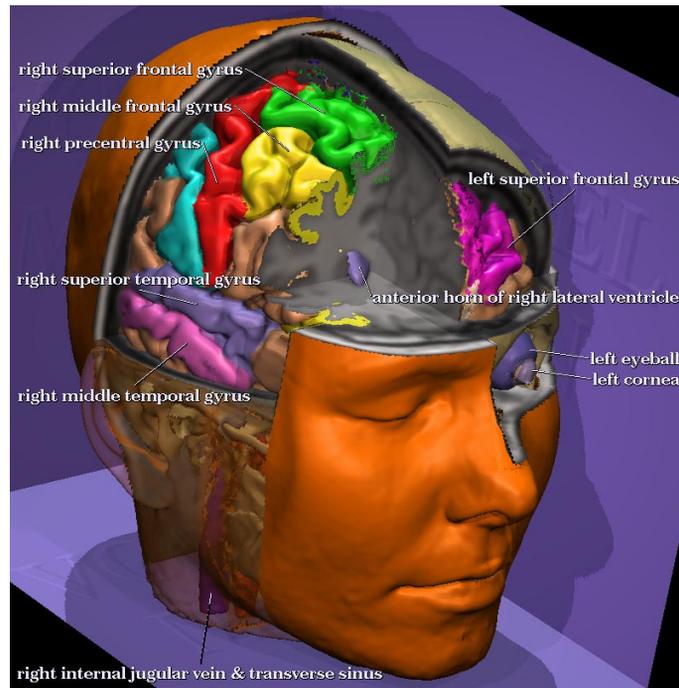


Abbildung 2: *Beispiel einer 3D-Darstellung, rekonstruiert aus 128 Kernspintomogrammen.*

1. Verfahren, die eine reine Oberflächenbeschreibung der Objekte in Form von Polygonen (meist Dreiecken) oder höheren Polynomen erzeugen.
2. Verfahren, die den ursprünglichen Stapel von Schnittbildern als diskretes Volumen, bestehend aus elementaren Volumenelementen (Voxel), auffassen.

Bei den polygonbasierten Verfahren werden die segmentierten Objekte auf ihre Oberfläche reduziert, wobei die ursprüngliche Grauwertinformation verloren geht. Gerade im diagnostischen Bereich sind interne Grauwertmuster aber von großem Interesse, auch wenn diese nicht oder nur sehr schwer weiter segmentierbar sind. Es ist also nicht von vornherein zu entscheiden, welche Voxel als bedeutungslos eingestuft und eliminiert werden können. Außerdem ist die Anzahl der generierten Oberflächensegmente bei medizinischen Datensätzen häufig so groß, daß nur durch Reduktion der Anzahl der Segmente eine handhabbare Menge entsteht. Diese Vergrößerung und der Verlust der Volumeninformation lassen die polygonbasierten Verfahren für eine realistische Visualisierung als eher ungeeignet erscheinen.

Die volumenbasierten Verfahren arbeiten mit den Intensitätswerten der Schnittbilder, so daß die in den Intensitätsmustern enthaltene Information für eine realistische Visualisierung ausgenutzt werden kann. Ein Kernproblem besteht jedoch darin, daß durch die diskrete Abtastung die Auflösung des Bildvolumens begrenzt ist und sich die bei der Segmentation gefundenen Organgrenzen als Treppenfunktion darstellen. Direkt aus dieser Funktion berechnete Oberflächennormalen, die für die Berechnung der Lichtreflexion notwendig sind, ergeben keine realistischen Bilder. Eine möglichst exakte Schätzung der echten Organgrenzen, und damit auch der Oberflächennormalen, ist deshalb eine Grundvoraussetzung für eine realistische Visualisierung. Es ist darüber hinaus eine Eigenschaft der volumenbasierten Verfahren, daß neben einer reinen Oberflächendarstellung auch „Durchleuchtungs“-Darstellungen wie Volume-Rendering, Röntgenprojektion und Maximum-Intensitäts-Projektion möglich sind.

Bei den bisher in der Literatur beschriebenen Projektionsverfahren für segmentierte Volumina entstehen erhebliche Artefakte bei Objekten, deren Größe in der Nähe der Auflösung der Daten liegt, sowie bei starker Vergrößerung und an den Objektkanten. Hier wird die diskrete Struktur der Volumendaten sichtbar. Eine nachträgliche Glättung der 3D-Abbildung stellt nur eine unzureichende Lösung dar. Eine Methode, die dieses Problem umgeht, gibt es bisher nicht.

Es gibt jedoch nicht nur verschiedene Visualisierungsformen, sondern es können auch mehrere darzustellende Parameter existieren. In zunehmendem Maße werden von einem Patienten sowohl Kernspintomogramme als auch korrespondierende PET-Schnittbilder oder multiple MR-Volumina mit unterschiedlichen Pulssequenzen und Parametergewichtungen erfaßt. In der Strahlentherapieplanung werden aus CT-Daten Dosisverteilungen berechnet sowie Zielvolumina und Risikoorgane bestimmt. Datenstrukturen und Operationen für solche **multiparametrischen** Bildvolumina, die einem Benutzer die uneingeschränkte Kombination der Parameter in Abhängigkeit von der Fragestellung, dem darzustellenden Objekt oder dem Bildausschnitt erlaubt, gibt es bisher nicht.

Je nach Gesichtspunkt können bei der Segmentation Objekte als unterschiedliche Voxelmengen definiert werden. Es entstehen dadurch räumlich verschiedene überlappende Objekte in Abhängigkeit davon, ob z. B. die Hirnrinde in die Hirnwindungen (Wissensdomäne „Struktur“) oder in die kortikalen Funktionsgebiete (Domäne „Funktion“) segmentiert wird. Diese Objekte werden durch domänenabhängige Attribute unterschieden. Jedes Voxel kann also **multiattributiert** sein. Diese Eigenschaft wird besonders für dreidimensionale anatomische Atlanten benötigt. Datenstrukturen, die eine (einschränkungsfreie) Visualisierung für beliebige Bedeutungen für die Attribute erlauben, wurden bisher nicht beschrieben.

1.1 Zielsetzung

Ziel dieser Arbeit ist deshalb die Entwicklung von Methoden und die Implementation eines Systems für die realistische 3D-Visualisierung von multiparametrischen und multiattributierten Volumendaten. Die Kernaufgabe besteht darin, die Lage und Orientierung der Oberflächen aus den diskreten attributierten Objekten im Subvoxelbereich zu bestimmen, um Aliasing-Effekte bei Animationen und starken Vergrößerungen, wie sie z. B. bei endoskopischen Ansichten erforderlich sind, zu vermindern.

Die folgende Funktionalität soll realisiert werden:

- die Objektoberflächen sollen möglichst naturgetreu dargestellt werden;
- die Volumenelemente sollen multiattributierbar sein, d. h., jedem Voxel können beliebige Attribute aus verschiedenen Domänen zugeordnet werden, z. B. hinsichtlich Morphologie, Funktion oder Blutversorgung, die aus verschiedenen bildgebenden Verfahren gewonnen oder durch Simulation erzeugt wurden wie etwa der Fluß eines Kontrastmittels durch die Blutgefäße;
- zwischen den Objekten aus unterschiedlichen Domänen sollen Verknüpfungen durchführbar sein, um z. B. diejenige Hirnregion zu markieren oder zu entfernen, in der das motorische Zentrum liegt und die von einem bestimmten Blutgefäß versorgt wird;
- die Objekte sollen aus multiparametrischen Daten definierbar sein, so daß z. B. Knochenstrukturen aus Computertomogrammen und Weichteile aus Kernspintomogrammen vi-

sualisiert werden oder die Hirnaktivität aus PET-Daten sichtbar gemacht oder die Oberflächenfarbe eines Objektes aus anatomischen Schnittbildern berechnet wird;

- die räumlichen Beziehungen zwischen den Objekten sollen repräsentiert und sichtbar gemacht werden, um z. B. ein Objekt, das vor dem dargestellten liegt, hinzuzufügen oder die weiße Hirnsubstanz dort einzufärben, wo eine bestimmte Hirnwindung angrenzt;
- verschiedene Visualisierungsalgorithmen wie Oberflächendarstellung, „Volume-Rendering“, Maximum-Intensitäts-Projektion etc. sollen innerhalb einer Darstellung benutzt werden können;
- das Bildvolumen soll mit Hilfe von frei definierbaren Schnittebenen in Regionen aufteilbar sein, in denen die Bildoperationen wie z. B. das Wegschneiden eines Objektes oder die Wahl einer anderen Visualisierungsart selektiv ausgeführt werden können;
- auf den Schnittebenen sollen die ursprünglichen Intensitätswerte dargestellt werden können;
- es soll die Möglichkeit der endoskopischen Darstellung bestehen, d. h. die Verlagerung des Betrachterstandortes in das Objekt hinein, z. B. in ein Blutgefäß oder die Speiseröhre;
- es sollen künstliche Objekte in polygonaler Repräsentation, wie z. B. Werkzeuge und nicht segmentierbare Strukturen, in die Darstellung integriert werden können;

Das System soll für verschiedene Anwendungen geeignet sein:

- als Visualisierungskern für die Verwendung im Rahmen von Segmentations- und Registrierungswerkzeugen;
- für klinische Anwendungen wie z. B. der Darstellung multiparametrischer Datensätze in der MR-Angiographie, funktionaler Aspekte aus registrierten MR- und PET-Volumina oder Dosisverteilungen in der Strahlentherapieplanung;
- für die Entwicklung dreidimensionaler anatomischer Atlanten auf der Basis klinischer MR- und CT-Datensätze sowie aus den anatomischen Schnittbildern des Visible-Human-Projekts;
- als Autorensystem für die Erzeugung von Lehrmaterial.

Um die vorgenannten Eigenschaften zu ermöglichen, ist die Entwicklung geeigneter Datenstrukturen und einer Benutzerschnittstelle, die eine konsistente und zugleich flexible Spezifikation des Visualisierungsziels erlaubt, erforderlich.

Die Arbeit gliedert sich wie folgt:

Im Abschnitt 2 wird zunächst der Stand der Forschung dargestellt, wobei insbesondere die bisher publizierten Darstellungsverfahren kurz rekapituliert werden.

Aufbauend auf diesen Verfahren werden im Abschnitt 3 die in dieser Arbeit neu entwickelten Methoden vorgestellt. Zunächst werden die Bestimmung der Attributwerte und die Lokalisation der Objektoberfläche im Subvoxelbereich erläutert, die Voraussetzung für eine realistische Visualisierung sind. Die Subvoxellage der Oberfläche dient dann einer verbesserten Abschätzung der Oberflächenneigung, die aus dem Grauwertgradienten abgeleitet wird. Für

solche Bereiche der Oberflächen, an denen kein Grauwertgradient berechenbar ist, wird ein einfaches Interpolationsverfahren für die Attribute angewendet.

Anschließend wird eine Erweiterung der Attributbestimmung für multiparametrische Daten, wie z. B. die Farbschnittbilder aus dem Visible-Human-Projekt, gezeigt. Für eine freie Kombinierbarkeit unterschiedlicher Darstellungsformen wird das Konzept der regionenbasierten Visualisierung eingeführt. Das Kapitel schließt mit der Entwicklung von Verfahren und Datenstrukturen für mehrfach attributierte Volumina, die insbesondere bei der Generierung dreidimensionaler anatomischer Atlanten Anwendung finden.

Im Abschnitt 4 wird auf zwei wesentliche Aspekte der Implementation eingegangen: Zum einen auf die Benutzerschnittstelle, die in Form einer Beschreibungssprache für die Steuerung der Visualisierung realisiert wird, und zum anderen auf Programmteile, die die Rechenzeit maßgeblich beeinflussen.

Die Anwendung der in dieser Arbeit entwickelten Verfahren auf ein künstliches Objekt sowie auf verschiedene klinische Datensätze und die Farbschnittbilder aus dem Visible-Human-Projekt wird in Abschnitt 5 gezeigt. Zugleich erfolgt anhand der Ergebnisse eine qualitative Bewertung. In den Schlußfolgerungen werden verschiedene Weiterentwicklungsmöglichkeiten aufgezeigt.

Im Anhang befinden sich die Ableitung der in dieser Arbeit entwickelten Faktoren für die Gradientengewichtung, die Reihenfolge der Transformationsschritte für die Erzeugung perspektivischer und stereoskopischer Ansichten sowie die Syntax der Beschreibungssprache für die Steuerung des Systems.

2 Stand der Forschung

Die klassische Computergraphik beschäftigt sich mit der Erzeugung und Darstellung von Objekten, die aus einfachen elementaren Strukturen wie Polygonen und insbesondere Dreiecken zusammengesetzt sind. Verschiedene Verfahren für die dreidimensionale Darstellung dieser Strukturen sind wohlbekannt und in zunehmendem Maße auf kommerziellen Graphiksystemen teilweise hardwaremäßig implementiert. Dagegen ist die Algorithmenentwicklung für die dreidimensionale Visualisierung diskreter Voxelfelder, wie sie die tomographischen Verfahren in der Medizin erzeugen, noch nicht soweit entwickelt. In Abbildung 3 sind die prinzipiellen Verfahrensschritte und Methoden der als Volumenvisualisierung bezeichneten Technik dargestellt.

Ausgangspunkt für die Volumenvisualisierung sind eine oder mehrere Schnittbildsequenzen. Diese Schnittbilder werden gegebenenfalls in einem Vorverarbeitungsschritt gefiltert, um die Bildqualität zu erhöhen. Üblich ist auch eine Datenreduktion durch die Verringerung der Intensitätsauflösung und das Entfernen leerer Hintergrundbereiche. Durch Interpolation zwischen den Schnittbildern werden isotrope Volumina mit kubischen Volumenelementen erzeugt. Existieren mehrere Akquisitionen desselben Objekts, bei denen die Volumina nicht deckungsgleich sind, ist zusätzlich eine Registrierung der Daten erforderlich, damit korrespondierende Voxel am selben Ort liegen.

2.1 Segmentation

Für eine individuelle Darstellung unterschiedlicher Strukturen ist bei fast allen Darstellungsverfahren ihre Kennzeichnung als einzelne Objekte erforderlich. Dieser Verfahrensschritt wird als Segmentierung bezeichnet. Bei den tomographischen Verfahren (CT, MR, PET etc.) stellt sich ein Objekt als Wertebereich der Meßgröße dar. Die Grenze zwischen zwei Objekten ist durch einen Intensitätssprung charakterisiert. Durch die begrenzte räumliche Auflösung der Aufnahmeverfahren tritt in denjenigen Volumenelementen, in denen die Grenzfläche zwischen den Objekten verläuft, eine Mittelung der Intensitäten der Objekte auf, die in der Medizin als Partial-Volumen-Effekt bezeichnet wird.

In sehr vielen Fällen lassen sich Objekte durch ein für sie spezifisches Intensitätsintervall charakterisieren. Dieses Verfahren wird Schwellwertsegmentation genannt. Für die Grenzvoxel, die den Partial-Volumen-Effekt aufweisen, erfolgt dadurch die Zuordnung zu demjenigen Objekt, dessen Intensitätsanteil überwiegt [Ney *et al.* 1991; Pommert *et al.* 1991; Pommert *et al.* 1992b]. Räumlich disjunkte Objekte können durch gleiche oder überlappende Intensitätsintervalle charakterisiert werden. Diese einfache Methode hat sich für viele praktische Anwendungen durchgesetzt, da sie bei der Computertomographie, der Kernspintomographie und auch bei optischen Schnittbilddaten die Segmentierung einer Vielzahl von Objekten erlaubt. Die Wahl der Schwellwerte kann dabei abhängig von der Qualität des Datensatzes und der zu segmentierenden Objekte schwierig sein. Auf diese Problematik soll hier nicht weiter eingegangen werden.

Da schwellwertbasierte Objekte Voraussetzung für die in dieser Arbeit entwickelten Visualisierungsverfahren sind, wird auf andere Verfahren nicht eingegangen. Eine ausführliche Diskussion anderer Verfahren und die Behandlung von Problemfällen findet sich in [Bomans 1994]. Ein interaktives Segmentationskonzept, das neben Schwellwerten morphologische Filter (Erosion, Dilation) und eine Zusammenhangskomponentenanalyse verwendet, ist

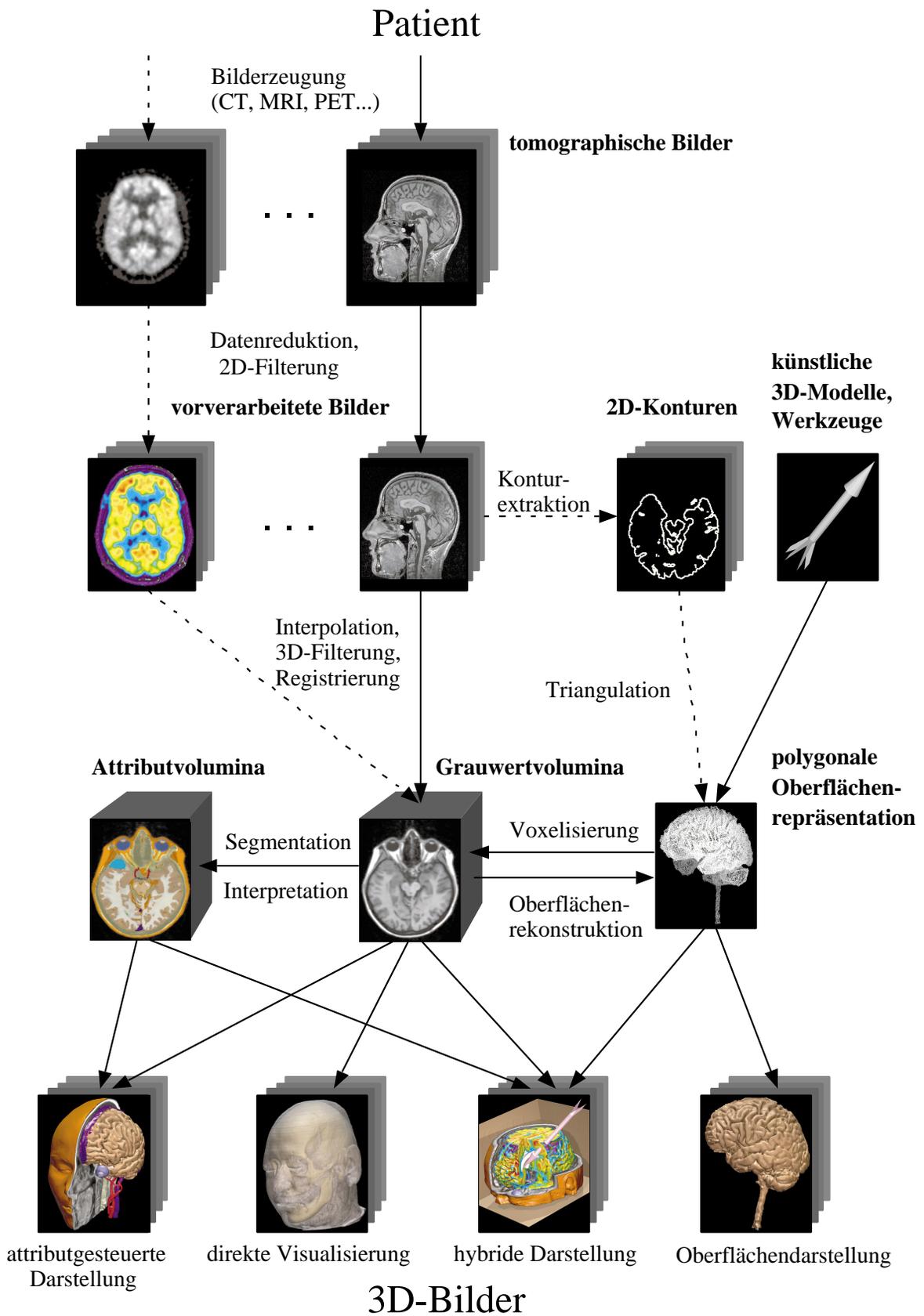


Abbildung 3: Verfahrensschritte für die Visualisierung tomographischer Volumendaten.

in [Höhne, Hanson 1992; Schiemann *et al.* 1992] beschrieben. Lassen sich die Objekte auch damit nicht trennen, so muß die Objektzuordnung manuell erfolgen. Den bisher publizierten Verfahren für segmentierte Volumina ist gemeinsam, daß sie den Verlauf der Objektoberfläche nur mit der Auflösung der ursprünglichen Diskretisierung der Volumendaten wiedergeben. Die Information über den genauen Verlauf der Oberfläche, die sich aus den Grauwerten und den verwendeten Schwellwerten ergibt, wird dabei nicht ausgenutzt.

2.2 Visualisierung mit der Strahlverfolgungsmethode

Für die Bestimmung der sichtbaren Oberflächen eines durch Segmentation gewonnenen Objekts von einem virtuellen Betrachterstandort aus hat sich das Strahlverfolgungsverfahren (engl. Ray-Tracing oder Ray-Casting) [Foley *et al.* 1990, Seite 701 ff] wegen seiner Einfachheit und der Möglichkeiten, Beleuchtungseffekte wie z. B. Transparenz und Schattenwurf zu realisieren, weitgehend durchgesetzt. Bei dieser Methode wird für jeden Bildpunkt (engl. Pixel) des resultierenden zweidimensionalen Bildes ein Abtaststrahl entsprechend der Abbildungsgeometrie erzeugt. Für jeden Strahl müssen dann die Schnittpunkte mit den Objektoberflächen bestimmt werden. Je nach verwendeter Datenstruktur unterscheiden sich die Methoden erheblich. Bei der in der Computergraphik am weitesten verbreiteten polygonalen Objektrepräsentation können die Schnittpunkte direkt berechnet werden [Badouel 1990]. Der Aufwand für dieses Verfahren steigt erheblich mit der Komplexität der Objekte, also mit der Anzahl der Polygone, die auf Schnittpunkte mit den Sehstrahlen getestet werden müssen. Die grundlegenden Techniken des Ray-Tracings sind heute Lehrbuchwissen. Viele nützliche Informationen sind außerdem im World-Wide-Web verfügbar wie z. B. [Grimstead 1996].

Bei der Volumenrepräsentation der Objekte gibt es dagegen keinen direkten Zugriff auf die Objekte. Hierbei muß vielmehr für jede neue Ansicht durch Abtastung des Volumens (resampling) für jede Strahlposition die entsprechende Objektzugehörigkeit bestimmt werden. Dieser Vorgang wird auch als diskretes Ray-Tracing bezeichnet [Yagel *et al.* 1992a]. Für jeden Strahl müssen zunächst der Eintrittspunkt und die Länge des Strahls im Volumen berechnet werden. Dieser Vorgang wird als Strahlbegrenzung (engl. clipping) bezeichnet. Ein effizienter Algorithmus hierfür findet sich in [Blinn 1991]. Die eigentliche Abtastung erfolgt dann vom Eintrittspunkt in das Datenvolumen bis zum Austrittspunkt. Klassische Strahlverfolgungsverfahren verwenden hierfür dreidimensionale diskrete Linienalgorithmen [Foley *et al.* 1990, Seite 74 ff]), um die langsamere Gleitkommazahlarithmetik zu vermeiden [Yagel 1992].

Trifft der Strahl auf seinem Weg auf eine Objektoberfläche, so wird an der Schnittstelle die Oberflächennormale berechnet und mit Hilfe eines Beleuchtungsmodells die Lichtintensität bestimmt, die in Richtung des Betrachters reflektiert wird. Wird kein Objekt getroffen, so wird der korrespondierende Bildpunkt auf die Hintergrundfarbe gesetzt.

Am weitesten verbreitet ist wegen seiner Einfachheit das Phong'sche Beleuchtungsmodell (Abbildung 4). Neben der Oberflächennormalen \vec{N} , der Blickrichtung \vec{V} und der Richtung des einfallenden Lichts \vec{L} gehen in dieses Modell drei objektspezifische Parameter ein, die Farbe des Objekts, die meist als RGB-Tupel im Rot-Grün-Blau-Farbraum angegeben wird, der Anteil der Streulichtkomponente und der diffusen Reflexion sowie eine spiegelnde Komponente für die Modellierung von Glanzlichtern. Die Glanzlichtkomponente wird in der Regel als unabhängig von der Wellenlänge angenommen, so daß die Farbe des Objekts hier nicht eingeht. Ein empirischer Parameter n steuert, ob die Oberfläche stark spiegelig ($n > 10$) oder eher matt ($n \leq 10$)

erscheint.

Das Modell läßt sich leicht auf mehrere Lichtquellen erweitern. Ebenso ist die Modellierung von Scheinwerferlicht mit begrenztem Lichtkegel möglich ([Foley *et al.* 1990, Seite 731 ff]). Daneben sind zahlreiche verbesserte Beleuchtungsmodelle, wie etwa der „Radiosity“-Ansatz von [Greenberg *et al.* 1986], entwickelt worden, die die Reflexion des Lichts von unterschiedlichen Materialien gemäß physikalischer Gesetzmäßigkeiten approximieren. Eine Übersicht verschiedener Beleuchtungsmodelle mit Implementation befindet sich in [Hall 1989].

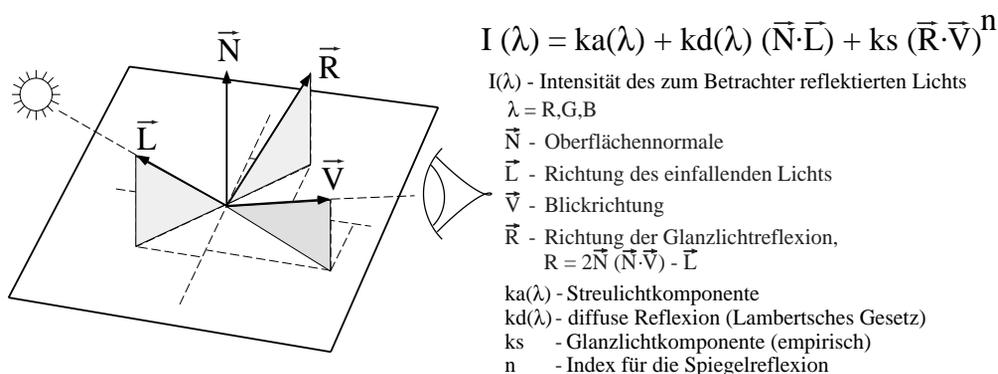


Abbildung 4: Das Phong'sche Beleuchtungsmodell.

Für natürliche Objekte wie die menschlichen Organe sind die Reflexionseigenschaften aber nicht bekannt, so daß diese durch empirische Werte ersetzt werden müssen. Der zum Teil erheblich höhere Rechenaufwand für diese Beleuchtungsmodelle erscheint daher kaum gerechtfertigt.

Die rekursive Anwendung der Strahlverfolgungsmethode ermöglicht die indirekte Beleuchtung der Objekte durch andere entfernte Objekte sowie die Berechnung transparenter Ansichten mit und ohne geometrischer Brechung des Strahls an den Oberflächen entsprechend den Materialeigenschaften. Für die Berechnung von Schlagschatten wird wiederum ein Strahl von der Objektoberfläche ausgehend in Richtung Lichtquelle erzeugt und geprüft, ob die Lichtquelle von dem betreffenden Oberflächenpunkt sichtbar ist. Wird auf dem Weg zur Lichtquelle ein nicht transparentes Objekt getroffen, so wird der entsprechende Oberflächenpunkt beschattet und erhält kein Licht.

Für die Berechnung der Strahlgeometrie mit Hilfe homogener 4×4 Matrizen sei auf Lehrbücher wie [Foley *et al.* 1990, Kapitel 6] verwiesen. Im Anhang B ist die Reihenfolge der einzelnen in der Regel nicht kommutativen Transformationsschritte erläutert, um zentralperspektivische und stereoskopische Ansichten zu erzeugen.

2.3 Oberflächenbasierte Visualisierung

Für eine ausschließliche Visualisierung von Objektoberflächen sind eine Reihe spezialisierter Datenstrukturen und Darstellungsverfahren entwickelt worden. In der Regel ist bei diesen Verfahren die Datenmenge der resultierenden Oberflächenbeschreibung eines Objekts erheblich kleiner als die der ursprünglichen Schnittbildsequenz, so daß eine schnelle Visualisierung möglich ist.

2.3.1 Polygonoberflächen

Für die Erzeugung einer polygonalen Objektbeschreibung aus tomographischen Bildern werden die Schnittbilder in der Regel manuell in zweidimensionale Konturen segmentiert. Aus den Stützpunkten der Konturen werden durch heuristische Verfahren dreidimensionale Polygonoberflächen erzeugt. Dieser Vorgang wird als Triangulation bezeichnet [Fuchs *et al.* 1977; Tiede *et al.* 1985; Tönnies 1989]. Da das Zuordnungsproblem, welche Stützpunkte benachbarter Konturen zu einem Polygon verbunden werden sollen, für komplexe Fälle nicht automatisch gelöst werden kann, haben diese Verfahren an Bedeutung verloren. Außerdem steht hierfür seit der Entwicklung des in Abschnitt 2.4.2 beschriebenen Marching-Cubes-Algorithmus ein wesentlich besseres Verfahren zur Verfügung.

Der Hauptvorteil der polygonalen Objektrepräsentation liegt in der hardwaremäßigen Unterstützung der Darstellung. Dadurch lassen sich je nach Leistungsfähigkeit der Graphikhardware mehr oder minder komplexe Objekte in Echtzeit visualisieren. Demgegenüber existieren spezielle Hardwarearchitekturen für eine Beschleunigung der Visualisierung von Volumendaten bisher nur als Prototypen. Beispielfhaft seien hier **Voxel Processor** [Goldwasser, Reynolds 1987], **PARCUM** [Jackèl 1985] und **CUBE** [Kaufman, Bakalash 1988] erwähnt. Ein Vergleich dieser und anderer Architekturen findet sich in [Kaufman 1991].

Erstmals ist für 1999 von Mitsubishi eine kommerziell erhältliche Hardwareerweiterung (**VolumePRO**) für Personal-Computer (PC) angekündigt [Pfister 1996], die allerdings noch erhebliche funktionale Beschränkungen aufweist. So können beispielsweise keine perspektivischen Abbildungen erzeugt werden, wie sie für endoskopische Ansichten erforderlich sind.

2.3.2 Voxeloberflächen

Im „Cuberille“-Ansatz [Herman, Liu 1979] werden ausgehend von einem durch Segmentation erzeugten Binärvolumen die Grenzflächen zwischen Objekt und Hintergrund durch einen Suchalgorithmus bestimmt. Der Abstand dieser Grenzflächen zur Projektionsebene wird dargestellt, wobei weiter entfernte Flächen dunkler erscheinen als solche, die näher zum Betrachter liegen. Spätere Verbesserungen schätzen die Normalvektoren der Objektoberfläche an den Grenzflächen ab und verwenden einfache Beleuchtungsmodelle. Die binäre Volumeninformation ermöglicht jedoch nur eine geringe Anzahl unterschiedlicher Neigungswinkel der Normalvektoren, so daß Diskretisierungsartefakte sichtbar sind.

Bei den objektraumorientierten Abtastverfahren „Back-to-Front“ (BTF) und „Front-to-Back“ (FTB) [Frieder *et al.* 1985] wird das Binärvolumen bezüglich des Speicherzugriffs optimal durchlaufen. Bei der Rückprojektion der Objektvoxel auf die Bildebene müssen Aliasing-Effekte kompensiert werden.

Von den oberflächenbasierten Verfahren ist heute nur noch die polygonale Repräsentation von Bedeutung, da sie in der Computergraphik weit verbreitet ist und zahlreiche Systeme für die Generierung und Visualisierung existieren.

2.4 Volumenbasierte Visualisierung

Der wesentliche Vorteil der Volumenrepräsentation gegenüber den oberflächenbasierten Verfahren besteht darin, daß Informationen über die innere Struktur der Objekte in Form der Originalgrauwerte vorhanden sind, die bei einer Oberflächenrepräsentation verloren gehen. Zu jedem

Punkt im Volumen existieren Intensitätswerte, die z. B. auf Schnittebenen durch die Objekte dargestellt werden und für eine Beurteilung besonders in der radiologischen Diagnostik wichtig sein können.

2.4.1 Berechnung der Oberflächennormalen aus den Grauwertgradienten

Wie bereits erwähnt, gelingt es bei binären Voxeloberflächen nicht, natürlich aussehende Oberflächen zu erzeugen, weil nur eine geringe Anzahl von Neigungswinkeln erzeugt werden kann. In [Höhne, Bernstein 1986] wird erstmals ein Verfahren beschrieben, das den Partial-Volumen-Effekt tomographischer Volumendaten ausnutzt, um von schwellwertsegmentierten Objekten die Oberflächennormalen \vec{N} der Grenzvoxel benachbarter Objekte aus den Grauwertgradienten \vec{G} an den Oberflächenpunkten $P = (x, y, z)$ abzuleiten:

$$\vec{N} = \frac{\vec{G}(V(P))}{|\vec{G}(V(P))|}, \text{ mit} \quad (1)$$

$$\vec{G}(V(P)) = \left(\frac{\partial V}{\partial x}, \frac{\partial V}{\partial y}, \frac{\partial V}{\partial z} \right),$$

wobei $V(P) = V(x, y, z)$ das Grauwertvolumen repräsentiert. Die Komponenten G_x , G_y und G_z des Gradientenvektors an der Position $P = (x, y, z)$ werden durch den zentralen Differenzenquotienten der Grauwerte g der benachbarten Voxel des Grenzvoxels entlang der drei Raumachsen approximiert:

$$G_x = \frac{1}{2} (g_{(x+1,y,z)} - g_{(x-1,y,z)})$$

$$G_y = \frac{1}{2} (g_{(x,y+1,z)} - g_{(x,y-1,z)})$$

$$G_z = \frac{1}{2} (g_{(x,y,z+1)} - g_{(x,y,z-1)})$$

Der Faktor $\frac{1}{2}$ hat keinen Einfluß auf die Richtung des Normalenvektors und ist für die Berechnung überflüssig. Dieses Verfahren erzeugt eine feine Abstufung und große Variabilität der Oberflächennormalen, die zu einer neuen Qualität der 3D-Darstellung führt. Die Kombination von Schwellwerten und Grauwertgradienten hat sich in der praktischen Anwendung seither bewährt, da die Oberflächen bei der richtigen Wahl des Schwellwerts besonders glatt erscheinen und der tatsächlichen Oberfläche wahrscheinlich sehr nahe kommen. Eine quantitative Untersuchung dieses Phänomens ist Gegenstand einer anderen Untersuchung [Pommert 1999].

2.4.2 Der Marching-Cubes-Algorithmus

Einen Zwitter aus Volumen- und Oberflächenvisualisierung stellt der Marching-Cubes-Algorithmus [Lorensen, Cline 1987] dar. Hierbei wird einerseits aus dem Grauwertvolumen unter Ausnutzung des Partial-Volumen-Effekts mit Hilfe von Schwellwerten und Interpolation die Objektgrenze im Subvoxelbereich durch Anordnen von Dreiecken approximiert. Es wird aber andererseits eine polygonale Repräsentation der Objektoberflächen erzeugt, wobei die ursprüngliche Volumeninformation verloren geht. Dieses Verfahren ist den heuristischen Triangulationsansätzen weit überlegen. Die Zahl der erzeugten Dreiecke kann

für medizinische Objekte leicht mehrere hunderttausend bis einige Millionen betragen, so daß eine Darstellung auch mit spezieller Graphikhardware nur schwer möglich ist. In diesem Fall muß die Objektbeschreibung weiter reduziert werden [Schroeder *et al.* 1992; Wilmer 1993].

Ein inverses Verfahren, das umgekehrt zum Marching-Cubes-Algorithmus aus polygonalen Objektbeschreibungen Volumendatensätze erzeugt (Voxelisierung), ist in [Kaufman *et al.* 1993] beschrieben. Dabei werden die Polygone entsprechend der gewählten räumlichen Auflösung des resultierenden Volumens abgetastet und alle Volumenelemente, die zum Objekt gehören, markiert. Die Qualität der Darstellung ist jedoch stark von der Auflösung des Volumens abhängig, wobei eine Verdoppelung der Auflösung eine Verachtfachung des benötigten Speicherplatzes und entsprechend mehr Rechenzeit erfordert.

2.4.3 Segmentationsfreie Visualisierung

Eine wesentliche Eigenschaft der Volumenrepräsentation ist es, daß sie auch eine Visualisierung ohne eine explizite Repräsentation der Objekte erlaubt. Hierbei wird während der Projektion das Intensitätsprofil entlang des Sehstrahls nach Maßgabe der in ihr enthaltenen Intensitätsmuster zu einer Bildintensität komprimiert. Die wesentlichen Verfahren dieser Art werden im folgenden skizziert.

2.4.3.1 Das Volume-Rendering

Bei dem von [Drebin *et al.* 1988; Levoy 1988] entwickelten Volume-Rendering werden den Grauwerten unterschiedliche Durchsichtigkeitswerte zugeordnet. Bei der Projektion werden diese Transparenzwerte mit dem Betrag des Grauwertgradienten an der jeweiligen Abtastposition gewichtet und ergeben so für jedes Voxel einen Gesamttransparenzwert, die sogenannte Opazität. Entlang eines Sehstrahls wird für jedes Voxel ein Schattierungswert mit dem Phong-schen Beleuchtungsmodell errechnet und mit dem Opazitätswert gewichtet. Dieser Vorgang wird solange fortgesetzt, bis entweder die Resttransparenz eine Schwelle unterschreitet, bei der die dann nicht berücksichtigten Voxel nur noch unwesentlich zum Resultat beitragen, oder ein nicht durchsichtiges Voxel erreicht wird. Formal ergibt sich der Intensitätswert I eines Bildpunktes (x, y) aus der rekursiven Gleichung [Tiede *et al.* 1990]:

$$I(x, y) = F(x, y, 0, 1) \quad \text{mit}$$

$$F(x, y, z, l) = \begin{cases} l \cdot \text{Hintergrundfarbe, falls } l \leq t \text{ oder } z \geq z_{max} \\ l \cdot O(x, y, z) \cdot S(x, y, z) + F(x, y, z + 1, l \cdot (1 - O(x, y, z))) , \text{ sonst} \end{cases}$$

- t : Schwellwert für verbleibende Restlichtstärke
- z_{max} : maximale Strahllänge
- l : Anteil des einfallenden Lichtes
- $O(x, y, z)$: $W(g(x, y, z)) \cdot G(x, y, z)$, Opazität
- $W(g(x, y, z))$: Transparenzwert des Grauwertes g
- $G(x, y, z)$: gewichteter Betrag des Grauwertgradienten
- $S(x, y, z)$: Lichtreflexion gemäß Beleuchtungsmodell

Durch die Gewichtung der Grauwerte mit dem Gradienten werden die Grenzflächen zwischen Materialien mit unterschiedlichen Intensitäten hervorgehoben, während homogene Grauwertbereiche unterdrückt werden. Dadurch entstehen zum Teil schwer interpretierbare Ansichten mit multiplen transparenten Oberflächen (Abbildung 5). In der einfachsten Form, bei der alle Grauwerte unterhalb eines Schwellwerts die Opazität 0 erhalten und alle darüber liegenden den Wert 1, entspricht das Volume-Rendering der Schwellwertsegmentation mit Oberflächenschattierung. Im allgemeinen erzeugt das Volume-Rendering keine expliziten Objekte und Oberflächen, die einer weiteren Manipulation zugänglich wären. Für die Problematik bei der Zuweisung der Transparenzwerte zu den Grauwerten und deren Gewichtung durch den Gradienten sei auf [Wiebecke *et al.* 1989; Tiede *et al.* 1990; Wulff 1992] verwiesen.



Abbildung 5: *Volume-Rendering: Multiple transparente Oberflächen entstehen durch Gewichtung von Grauwerten mit den Grauwertgradienten entlang des Abtaststrahls. Je stärker der Betrag des Gradienten ist, desto undurchsichtiger ist das Voxel an der Abtastposition.*

2.4.3.2 Die Maximum-Intensitäts-Projektion (MIP)

Hierbei wird entlang jedes einzelnen Sehstrahls das Maximum der Grauwerte ermittelt und dargestellt. Diese Technik ist nur sinnvoll auf entsprechende Datensätze wie etwa in der CT- und MR-Angiographie anwendbar, wo durch Kontrastmittelgabe oder spezielle Akquisitionparameter die Blutgefäße auf den Schichtbildern hell erscheinen, alle anderen Strukturen aber dunkel abgebildet werden (Abbildung 6). Der Nachteil dieses Verfahrens ist, daß der Betrachter keinerlei Information über gegenseitige Verdeckungen gewinnt und somit nicht entscheiden kann, ob eine Struktur vor oder hinter einer anderen liegt oder ob scheinbar zusammenhängende Objekte wirklich zusammenhängend sind.

2.4.3.3 Die simulierte Röntgenbilddarstellung

Aus einem CT-Volumen, das ja mit Hilfe von Röntgenstrahlen erzeugt worden ist, können konventionelle Röntgenbilder in erster Näherung dadurch erzeugt werden, daß aus dem Intensitätsprofil entlang des Sehstrahls entsprechend dem Absorptionsgesetz eine Bildintensität berechnet wird. Auf diese Weise lassen sich Röntgenbilder aus beliebigen Blickrichtungen und mit unterschiedlichen Strahlgeometrien erzeugen (Abbildung 7).

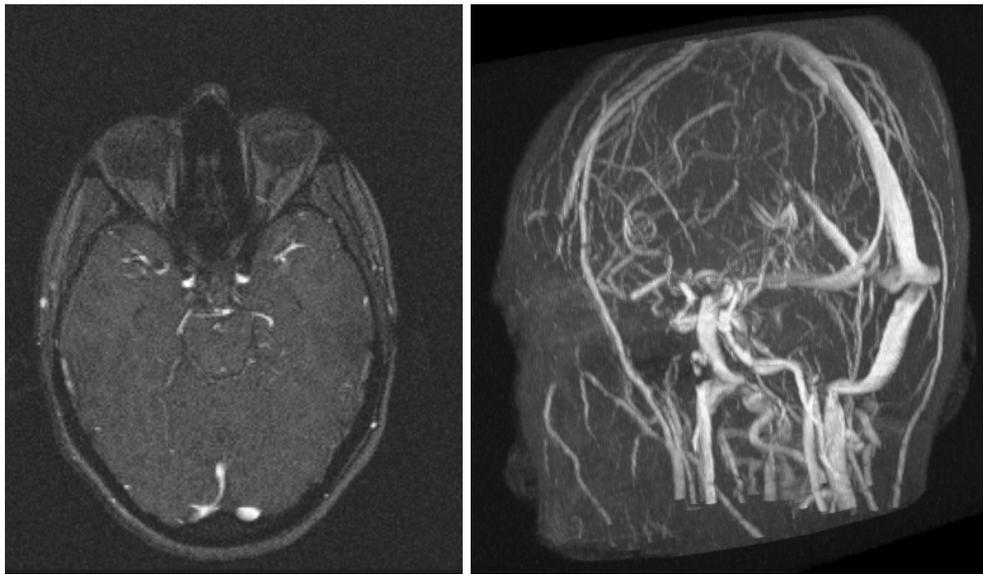


Abbildung 6: Schichtbild aus einer MR-Angiographie-Sequenz (links) und einer daraus erzeugten Maximum-Intensitäts-Projektion (MIP) der Blutgefäße des Kopfes (rechts).

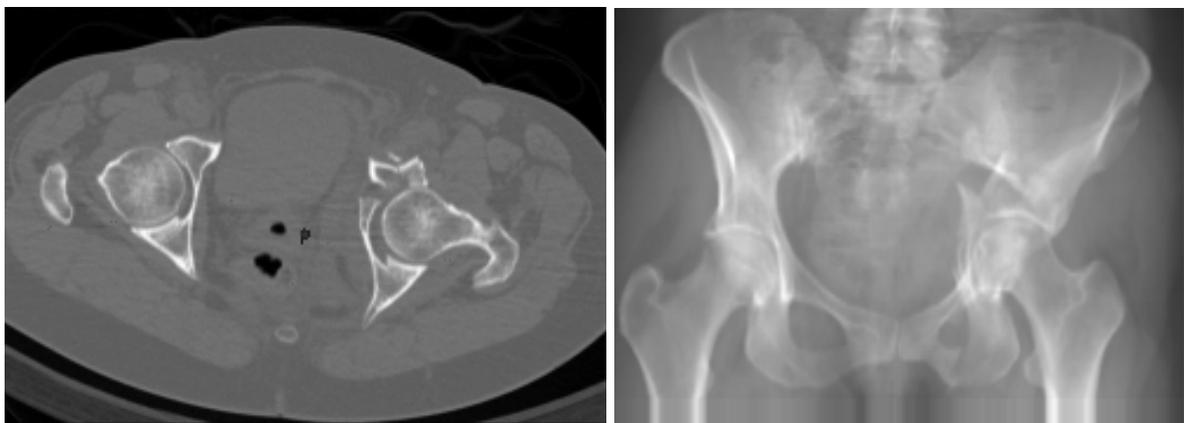


Abbildung 7: Transversale CT-Schicht im Hüftgelenkbereich (links) und aus dem entsprechenden CT-Volumen simulierte Röntgenbildprojektion aus frontaler Sicht.

Die Vorteile der segmentationsfreien Darstellungsverfahren liegen darin, daß das eventuell zeitaufwendige Segmentieren entfällt, keine weiteren Volumina für die Objektbeschreibungen notwendig sind und vor allem, daß die Qualität der Darstellung durch einfache Überabtastung und Interpolation der Grauwerte auf Kosten der Rechenzeit praktisch beliebig erhöht werden kann [Pommert *et al.* 1992a]. Der wesentliche Nachteil liegt jedoch in dem Rückschritt zu einer Projektionsdarstellung, bei der alle im Gesichtsfeld befindlichen Organe überlagert dargestellt werden und so eine visuelle Interpretation behindert wird.

2.4.4 Die attributgesteuerte Oberflächen- und Volumendarstellung

So einfach die segmentationsfreien Verfahren auch zu handhaben sind und so ansprechende Bilder sie auch liefern, für eine realistische und quantitative Visualisierung, wie sie in der Medizin nötig ist, sind sie nicht geeignet. Deshalb ist die Volumenvisualisierung explizit voxelweise durch Attribute charakterisierter Objekte die Methode der Wahl.

Die Verwendung von Attributen zur Steuerung der Visualisierung von Volumendaten wurde unabhängig voneinander von [Höhne *et al.* 1987] und [Stiehl, Jackèl 1987] entwickelt. Hierbei wird jedem Voxel neben seiner Grauwertintensität auch ein Attributwert bezüglich einer Domäne, wie z. B. die Zugehörigkeit zu einem bestimmten Organ, zugeordnet. Alle Voxel mit demselben Attributwert bilden ein elementares Objekt, so daß sich z. B. eine bestimmte Hirnwindung aus denjenigen Volumenelementen zusammensetzt, die den Attributwert „23“ haben. Jedem Objekt sind Darstellungsparameter wie Farbe und Reflexionseigenschaften zugeordnet, damit verschiedene Objekte auch unterschiedlich darstellbar sind.

Die Zuordnung der Attributwerte erfolgt soweit möglich durch geeignete Klassifikatoren wie der in Abschnitt 2.1 beschriebenen Schwellwertsegmentation. Das Problem der attributgesteuerten Volumenvisualisierung besteht darin, aus den diskreten Attributwerten, die die Objektinformation enthalten, die Objektgrenze möglichst genau zu bestimmen. Bei der Abtastung des Volumens mit Hilfe klassischer Linienalgorithmen wird der jeweilige Attributwert durch das der Abtastposition am nächsten gelegene Voxel bestimmt (Abbildung 8, links). Solange die Größe eines Voxels in etwa der eines resultierenden Bildpunktes der Projektionsebene entspricht, ist die voxelige Oberflächenstruktur, die dieser „nearest-neighbour“ Zugriff erzeugt, nicht sichtbar. Der Effekt tritt jedoch beim Versuch der Überabtastung auf, wenn bei starken Vergrößerungen die Projektion eines Voxels viele Bildpunkte überdeckt. Ziel ist es also, die Objektoberfläche möglichst gut im Subvoxelbereich zu approximieren, um derartige Artefakte zu vermeiden (Abbildung 8, rechts).

Eine Möglichkeit besteht in der Erhöhung der räumlichen Auflösung der Volumendaten [Yagel *et al.* 1992a]. Nach dem Abtasttheorem müßte die räumliche Auflösung mindestens doppelt so hoch sein wie die maximale in den Daten vorkommende Frequenz. Aber schon eine (nicht unwahrscheinliche) Unterteilung jedes Voxels in $10 \times 10 \times 10$ Subvoxel erhöht den Speicherbedarf für ein durchschnittliches Volumen von 10 Megabyte auf 10 Gigabyte. Ebenso müßte die Rechnerleistung um den Faktor 1000 höher sein, um dieselbe Geschwindigkeit wie mit der Originalauflösung zu erreichen. Außerdem kann die Auflösung bei tomographischen Daten oft entweder aus technischen Gründen oder wichtiger noch, um den Patienten nicht unnötig zu belasten, gar nicht erhöht werden. Dieser Weg ist somit für klinische Daten nicht praktikabel.

In [Drebin *et al.* 1988] werden den Voxeln Wahrscheinlichkeitswerte für ihre Zugehörigkeit zu einem bestimmten Objekt zugeordnet, um eine binäre Objektklassifikation zu vermeiden. Dadurch ist es möglich, ein Voxel z. B. zu 70% dem Knochen und zu 30% den Weichteilen zuzuordnen. Für eine beliebige Abtastposition kann so der an dieser Stelle überwiegende Anteil eines Objekts durch Interpolation der Wahrscheinlichkeitswerte berechnet werden. Allerdings ist die Zuweisung von Wahrscheinlichkeitswerten für eine größere Anzahl von Objekten in einem Voxel und deren Interpolation nicht einfach, und der Speicherbedarf für die notwendige Datenstruktur steigt stark an.

Werden die Volumendaten bezüglich verschiedener Wissensbereiche segmentiert [Stiehl 1987; Höhne *et al.* 1988], d. h., werden den Voxeln gleichzeitig mehrere Attribute aus verschiedenen Domänen zugeordnet, entstehen neue Objekte, die sich aus einer mengentheoretischen Verknüpfung der einzelnen Attribute ergeben. Zum Beispiel können sich die Voxelmengen eines Funktionsgebiets (z. B. das Sprachzentrum) und eines Versorgungsgebiets eines bestimmten Blutgefäßes überlappen. Das daraus resultierende multiattributierte Objekt hat also die Eigenschaft, sowohl Teil des Sprachzentrums zu sein als auch von einem bestimmten Blutgefäß

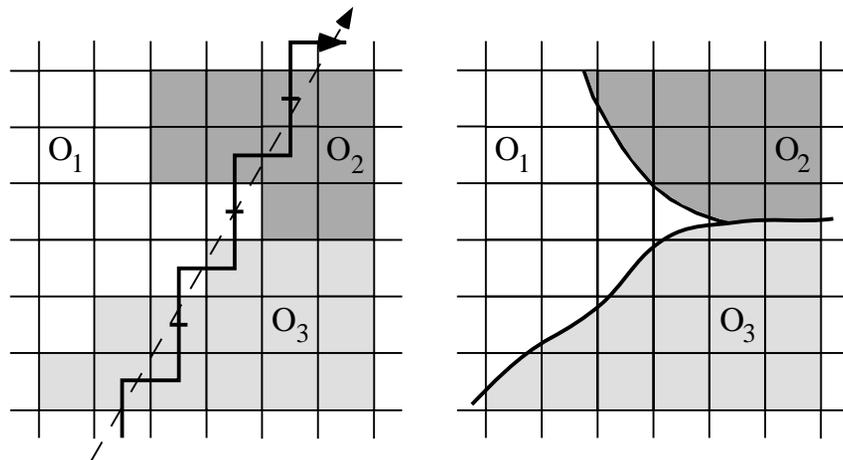


Abbildung 8: *Klassisches „Ray-Casting“: Das Volumen wird entlang der Sehstrahlen an diskreten Voxelpositionen abgetastet (links). Ziel ist eine möglichst gute Approximation der tatsächlichen Objektgrenze im Subvoxelbereich (rechts).*

versorgt zu werden. In [Stiehl 1987] wurde ein Szenario für die Anwendung multiattribuierter Volumendaten im medizinischen Kontext entworfen, die vorgeschlagene XCE-Repräsentation (extended cell enumeration) ist aber nur rudimentär beschrieben [Stiehl, Jackèl 1987; Tönnies, Lemke 1994]. Ein konsistentes Konzept für eine Realisierung verschiedener objekt- (oder regionen-) abhängiger Darstellungsvarianten multiattribuierter Volumendaten findet sich in der Literatur nicht.

Für die etablierten Verfahren der Visualisierung medizinischer Volumendaten existieren zahlreiche häufig als Pilotprojekt implementierte Softwarepakete wie z. B. ANALYZE [Robb 1996], InViVo [Sakas 1996], ViewNix [Udupa 1995], VolVis [Kaufman 1996], The Visualization Toolkit [Schroeder *et al.* 1998], um nur einige zu nennen. Auch die kommerziell erhältlichen universellen graphischen Systeme wie AVS, HP-Voxelator, IBM-Data-Explorer, Khoros, OpenGL und SGI-Explorer verfügen über eine Reihe von Funktionen für die Volumenvisualisierung. Adressen, Verfügbarkeit und teilweise auch Preise zahlreicher Systeme sind in [Montgomery 1998] zusammengestellt. In [Kaufman *et al.* 1994] ist eine Übersicht der Forschungseinrichtungen, die sich mit der Volumenvisualisierung befassen, zusammengestellt. Eine Sammlung wichtiger Veröffentlichungen findet sich in [Kaufman 1991; Höhne, Pommert 1996].

Zusammenfassend ist zu sagen, daß keines der bisher publizierten Verfahren und Systeme die in Abschnitt 1.1 genannten Forderungen für segmentierte Volumina erfüllt, nämlich die Rekonstruktion der Oberfläche im Subvoxelbereich, die Möglichkeit, multiple Objekte aus unterschiedlichen Domänen zu definieren und mengentheoretisch miteinander zu verknüpfen, sowie verschiedenartige Darstellungsformen regionen- und objektbezogen in einer einzelnen Ansicht zu kombinieren.

3 Methode

Voraussetzung für eine realistischere Darstellung von Oberflächen und aufgeschnittenen Objekten auch bei starker Vergrößerung ist die möglichst genaue Bestimmung des Attributwertes an der jeweiligen Position des Abtaststrahls. Die „wahre“ Oberfläche eines Objektes kann prinzipiell wegen der Unterabtastung der Aufnahmeverfahren nicht rekonstruiert werden. Deshalb beziehen sich die Begriffe „genau“ und „exakt“ im folgenden nicht auf das reale Objekt, sondern auf die Möglichkeit, die durch den verwendeten Klassifikator definierte Objektgrenze in Subvoxelauflösung zu verfeinern.

Dazu wird ein geometrisch exaktes Verfahren zur Abtastung entlang des Sehstrahls entwickelt, das die Oberflächenposition im Subvoxelbereich bestimmt und für diese Subvoxelpositionen den korrekten Attributwert liefert.

Für die Untersuchung innerer Strukturen von Objekten, die nicht weiter segmentierbar sind, wird ein Konzept entwickelt, das eine selektive Manipulation der Objekte (z. B. wegnehmen, transparent machen, einfärben) in durch Schnittebenen definierten Regionen erlaubt.

Danach wird die Verallgemeinerung der Objektdefinition für eine mehrfache Attributierung der Voxel aus verschiedenen Domänen beschrieben. Schließlich wird eine neue Datenstruktur entwickelt, die die unterschiedlichen Darstellungsverfahren abhängig von Objekt und Region miteinander kombinierbar macht.

3.1 Berechnung der Strahlgeometrie

Viele Volumenvisualisierungsalgorithmen basieren aus Geschwindigkeitsgründen auf der Erzeugung paralleler Sehstrahlen, die das Datenvolumen abtasten. Hierbei braucht die Strahlgeometrie für eine bestimmte Ansicht nur einmal berechnet zu werden. Für alle Bildpunkte einer Ansicht wird dann dieselbe Geometrie verwendet. Neben dem geometrisch nicht korrekten Aussehen solcher Parallelprojektionen liegt ein gravierender Nachteil dieser Vorgehensweise darin, daß keine endoskopischen Ansichten berechnet werden können, d. h., der Betrachter muß sich immer außerhalb des Volumens befinden. Aus diesem Grund wurde das Modell der punktförmigen Kamera mit zentral-perspektivischer Abbildung gewählt, bei dem der Betrachter an eine beliebige Stelle, also auch innerhalb eines Blutgefäßes, der Speiseröhre oder der Bronchien, positioniert werden kann. Bei diesem Modell hat der Betrachter selbst keine Ausdehnung, so daß er in beliebig kleinen Strukturen plaziert werden kann. Die Berechnung der Strahlgeometrie erfordert hierbei die Verwendung homogener Koordinaten, wobei jedoch der Sonderfall behandelt werden muß, daß die homogene Komponente kleiner oder gleich „0“ werden kann. Das trifft für alle Koordinaten zu, die hinter oder genau im Betrachterstandort liegen. Eine genaue und verständliche Beschreibung der Phänomene homogener Koordinaten und deren Lösung findet sich in [Blinn 1993].

Bei den klassischen Linienabtastalgorithmen wird der Sehstrahl durch diskrete Voxelpositionen approximiert (vgl. Abbildung 8). Dabei geht die Subvoxelposition verloren, die im weiteren Verlauf des im Rahmen dieser Arbeit entwickelten Verfahrens noch benötigt wird. Aus diesem Grund erfolgt die Abtastung geometrisch korrekt unter Verwendung reellwertiger Inkremente. Die dabei entstehenden Positionen liegen im Rahmen der Rechengenauigkeit exakt auf dem Sehstrahl im Subvoxelbereich.

3.2 Klassifikationsgesteuerte Bestimmung der Attributwerte im Subvoxelbereich

Die Bestimmung der Objektgrenze im Subvoxelbereich erfordert, daß die Objektzugehörigkeit der Abtastpositionen ebenfalls im Subvoxelbereich ermittelt wird. Die Verwendung desjenigen Attributvoxels, das der jeweiligen Abtastposition am nächsten gelegenen ist, reicht hierfür nicht aus. Ein naheliegender Ansatz ist die trilineare oder höhere Interpolation des Intensitätswertes an der Abtastposition im Subvoxelbereich in Verbindung mit einer nachfolgenden Überprüfung durch einen Schwellwerttest, ob dieser Wert innerhalb des Intensitätsbereiches desjenigen Objekts liegt, das sich der Abtastposition am nächsten befindet. Liegt der Intensitätswert innerhalb, so befindet sich die Subvoxelposition des Abtaststrahls im Objekt, andernfalls außerhalb. Auf diese Weise wird das richtige Objekt aber nur dann gefunden, wenn die Grenzfläche nicht durch ein benachbartes (verschiedenes) Attributvoxel verläuft. Um dies zu garantieren, wird in [Bomans 1994] eine Dilation der Attributvoxel eines ausgewählten Objekts durchgeführt, wodurch sich die Objektbeschreibung ausdehnt und die Objektgrenze innerhalb der eigenen Attributvoxel verläuft (Abbildung 9, links). Durch den Schwellwerttest kann so immer festgestellt werden, ob die Abtastposition zum (dilatierten) Objekt gehört oder nicht. Gleichzeitig werden die Attributvoxel der benachbarten Objekte aber erodiert, so daß diese nicht mehr richtig bestimmt werden können. Wird beispielsweise das Gehirn dilatiert, so ist dadurch eine korrekte Ermittlung der Oberfläche des angrenzenden Knochens nicht mehr möglich.

Eine andere Möglichkeit besteht darin, für schwellwertsegmentierte Objekte die Intensitätsbereiche, die die Objekte charakterisieren, während der Strahlabtastung einzuschränken. Die Objektgrenzen verlaufen bei richtiger Wahl des Intensitätsbereichs innerhalb der zugehörigen Attributvoxel. Allerdings werden die Objekte kleiner, als sie ursprünglich segmentiert wurden und es entsteht eine fragwürdige Verschiebung der Objektgrenzen (Abbildung 9, rechts). Es ist zudem unklar, wie stark der Intensitätsbereich eingeschränkt werden muß, damit die Objektgrenze sicher innerhalb des zugehörigen Attributvoxels verläuft, so daß dieses Verfahren als eher ungeeignet eingestuft werden muß.

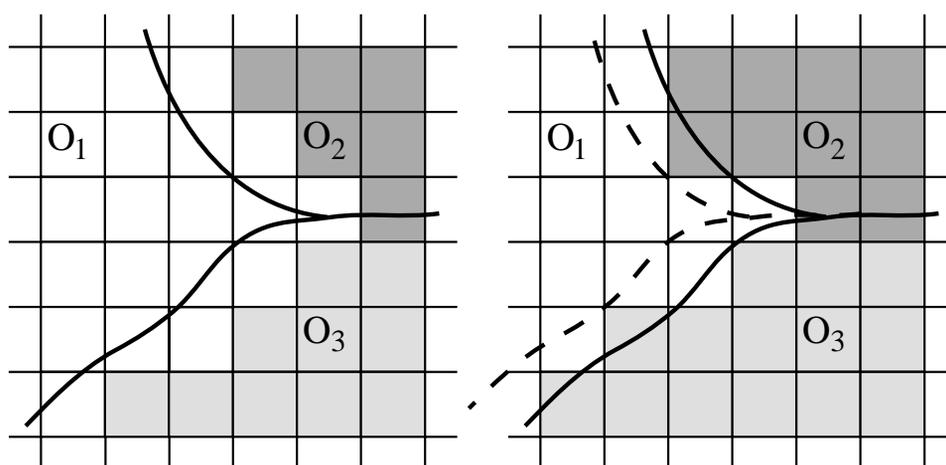


Abbildung 9: Links: Durch Dilation von Objekt O_1 kann dessen Oberfläche im Subvoxelbereich durch den Schwellwerttest korrekt bestimmt werden (vgl. Abbildung 8). Die gleichzeitige Erosion der Objekte O_2 und O_3 zerstört aber deren Erkennbarkeit. Rechts: Bei einer Modifikation des Schwellwerts während der Darstellung entsteht eine fragwürdige Verschiebung der Objektgrenzen (gestrichelte Linie).

In dieser Arbeit wurde ein neues Verfahren entwickelt, das die Objektzugehörigkeit während der Strahlverfolgung im Subvoxelbereich aus den acht Attributvoxeln der $2 \times 2 \times 2$ -Umgebung der jeweiligen Abtastposition unter Berücksichtigung der (Schwellwert-)Klassifikation bestimmt. Die $2 \times 2 \times 2$ -Umgebung einer (reellwertigen) Abtastposition $P = (x, y, z)$ umfaßt die Volumenelemente $V_{(i,j,k)}$ mit den Ganzzahlkoordinaten $i = \lfloor x \rfloor, \lfloor x + 1 \rfloor$; $j = \lfloor y \rfloor, \lfloor y + 1 \rfloor$ und $k = \lfloor z \rfloor, \lfloor z + 1 \rfloor$. Allgemein gilt für $n \times n \times n$ -Umgebungen: $i = \lfloor x - \frac{n}{2} + 1 \rfloor \cdots \lfloor x + \frac{n}{2} \rfloor$ für gerade n und $i = \lfloor x - \frac{n-1}{2} \rfloor \cdots \lfloor x + \frac{n-1}{2} \rfloor$ für ungerade n . Die Wertebereiche für j und k ergeben sich entsprechend.

Die Objektbestimmung an einer Abtastposition im Subvoxelbereich arbeitet nach folgendem Schema:

1. An der jeweiligen Abtastposition P_i wird der Intensitätswert g_i aus dem Grauwertvolumen durch (trilineare) Interpolation erzeugt.
2. Initialisierung eines Zählers mit „0“.
3. Für jedes der acht Attributvoxel der $2 \times 2 \times 2$ -Umgebung der Abtastposition wird untersucht, ob der interpolierte Grauwert g_i innerhalb des Intensitätsbereiches desjenigen Objekts liegt, das durch den jeweiligen Attributwert repräsentiert wird (Klassifikationsbedingung). Ist das der Fall, wird der Zähler um 1 erhöht. Gleiche Attributvoxel in der Umgebung werden aber nur einmal berücksichtigt, so daß ein in der Umgebung mehrfach auftauchendes Objekt nicht mehrfach berücksichtigt wird.
4. Nachdem die acht Attributvoxel untersucht wurden, befindet sich der Zähler in einem von drei Zuständen:
 - (a) Der Zähler ist „0“, d. h., keines der Objekte um die Abtastposition erfüllt seine jeweilige Klassifikationsbedingung. Der Intensitätswert g_i liegt also außerhalb aller den Objekten zugeordneten Intensitätsbereiche. In diesem Fall wird die Abtastposition als nicht klassifiziert markiert und es wird zur nächsten Abtastposition vorangegangen.
 - (b) Der Zähler hat den Wert „1“: Genau ein Objekt erfüllt seine Klassifikationsbedingung an der Abtastposition. Dieses Objekt wird für die weitere Darstellung benutzt.
 - (c) Der Zähler ist größer als „1“, das bedeutet, daß an der Abtastposition für mehr als ein Objekt die jeweilige Klassifikationsbedingung erfüllt ist. Dieser Spezialfall tritt unter verschiedenen Bedingungen auf, z. B. bei überlappenden Schwellwerten. So sind die verschiedenen Windungen der grauen Hirnsubstanz alle durch denselben Intensitätsbereich charakterisiert, so daß die Trennung manuell durchgeführt werden muß (siehe auch Abschnitt 3.3). An diesen „künstlichen“ Objektgrenzen existiert keine Grauwertkante.

Eine eindeutige Lösung kann hier nicht aus der Klassifikation der Objekte abgeleitet werden. Für diese Situation und für Objekte, die vollständig manuell segmentiert wurden und deshalb keinen Klassifikator besitzen, wurde ein Verfahren zur Objekt- und Oberflächenbestimmung durch Interpolation der Attributwerte entwickelt (siehe Abschnitt 3.5).

Mit diesem Verfahren wird für diejenigen Bereiche, wo die Objekte durch Schwellwerte segmentierbar sind, der korrekte Attributwert an der Abtastposition im Subvoxelbereich bestimmt.

Auch die problematischen Orte im Datenvolumen, die z. B. durch manuelle Segmentation oder morphologische Operatoren entstehen, werden erkannt (hier ist der Wert des Zählers aus dem eben beschriebenen Algorithmus größer als „1“). Der Attributwert ist hier im Subvoxelbereich nicht eindeutig bestimmbar. Diese Bereiche sind jedoch in der Regel lokal begrenzt, so daß eine sich ergebende Qualitätsminderung keine Auswirkungen auf die übrigen Bereiche hat.

Das Verfahren erscheint aufwendig, weil theoretisch acht verschiedene Objekte aneinandergrenzen können und somit acht verschiedene Klassifikationsberechnungen erforderlich wären. In der Praxis stoßen jedoch nur selten mehr als zwei oder drei verschiedene Objekte an einer Position zusammen, so daß der Mehraufwand in der Regel gering ist.

Die Methode ist leicht erweiterbar, falls Objekte aus unterschiedlichen Volumina segmentiert wurden. In diesem Fall erfolgt in Schritt 3 die Berechnung des interpolierten Intensitätswertes in jeweils demjenigen Volumen, aus dem das zu testende Objekt segmentiert wurde.

3.3 Lokalisation der Objektgrenze im Subvoxelbereich

Nachdem für die Abtastposition die Objektzugehörigkeit bestimmt wurde, müssen die für die Darstellung erforderlichen Parameter für das betreffende Objekt ermittelt werden. Soll das Objekt nicht dargestellt werden, wird einfach zur nächsten Strahlposition vorangegangen. Andernfalls muß die Oberflächennormale, ein Einheitsvektor, der senkrecht zur Oberfläche steht und für die Berechnung von Lichtreflexionen notwendig ist, berechnet werden. In [Höhne, Bernstein 1986; Tiede *et al.* 1990; Pommert *et al.* 1990] wurde gezeigt, daß für tomographische Aufnahmeverfahren, die einen Partial-Volumen-Effekt erzeugen, die Grauwert-Gradienten-Methode eine anderen Verfahren, wie z. B. dem Z-Buffer-Gradienten, überlegene Qualität der Oberflächennormalen erzeugt. Bisherige Verfahren berechnen den Gradienten an der Abtastposition des Strahls. Diese entspricht aber nicht der genauen Lage der Oberfläche, so daß Aliasing-Effekte auftreten. Bei dünnen Objekten kann es zudem zu vollkommen falschen Ergebnissen kommen [Pommert *et al.* 1992b].

Im Rahmen dieser Arbeit wurde deshalb ein Verfahren entwickelt, um die durch die (Schwellwert-)Segmentation definierten Grenzflächen der Objekte im Subvoxelbereich exakt zu bestimmen [Tiede *et al.* 1998]: Befindet sich an der augenblicklichen Abtastposition des Sehstrahls ein Objekt, das dargestellt werden soll, so muß zwischen dieser Position und der vorangegangenen Abtastposition die Objektgrenze verlaufen, andernfalls hätte die vorangegangene Position die Darstellung bestimmt. Es gilt also, die Lage der Oberfläche zwischen den beiden Abtastpunkten zu ermitteln. Das Verfahren arbeitet folgendermaßen:

1. Für das Objekt an der Abtastposition $P_i = (x_i, y_i, z_i)$ wird der Intensitätswert g_i durch trilineare Interpolation der Grauwerte bestimmt. Dieser Wert ist in der vorangegangenen Bestimmung der Objektzugehörigkeit (siehe Abschnitt 3.2) bereits berechnet worden und kann somit übernommen werden.
2. Für die vorangegangene Abtastposition $P_{i-1} = (x_{i-1}, y_{i-1}, z_{i-1})$ wird auf die gleiche Weise der Wert g_{i-1} ermittelt.
3. Bei der Verwendung von Schwellwerten für die Klassifikation der Objekte ergibt sich der Schnittpunkt $B = (x, y, z)$ der Objektoberfläche mit dem Abtaststrahl als

$$B = P_i - t(P_i - P_{i-1}) \quad \text{mit} \quad t = \frac{S - g_{i-1}}{g_i - g_{i-1}}, \quad (2)$$

wobei S den unteren Schwellwert angibt, falls $g_{i-1} < S$ ist. Andernfalls ist S der obere Schwellwert des Intensitätsbereiches, der das Objekt charakterisiert.

Geometrisch bedeutet dies, daß die Strecke $\overline{P_{i-1}P_i}$ im selben Verhältnis geteilt wird, wie der Schwellwert die Intensitätswerte g_{i-1} und g_i teilt. Diese lineare Interpolation der Objekt-oberfläche stellt nur eine erste Näherung dar, da die Intensität der durch Interpolation berechneten Grauwerte entlang des Abtaststrahls nicht unbedingt linear verläuft (Abbildung 10). Eine analytische Methode, um die korrekten Schnittpunkte zu bestimmen, ist aufwendig und für höhere Interpolationen, wie etwa die kubische Spline-Interpolation, nicht bekannt. Deshalb wurde ein Bisektionsalgorithmus implementiert, mit dem innerhalb weniger Iterationen die Oberfläche hinreichend genau approximiert werden kann. Dabei wird auf der Mitte M der Strecke $\overline{P_{i-1}P_i}$ der Intensitätswert g durch Interpolation bestimmt. Ist $g < S$, so wird $P_i = M$ gesetzt, andernfalls $P_{i-1} = M$. Dieser Vorgang wird solange wiederholt, bis $|g - S| < \epsilon$ ist. Das Verfahren arbeitet auch bei Interpolationen höherer Ordnung korrekt, solange höchstens ein Extremwert des Grauwertverlaufs zwischen zwei Abtastpositionen existiert.

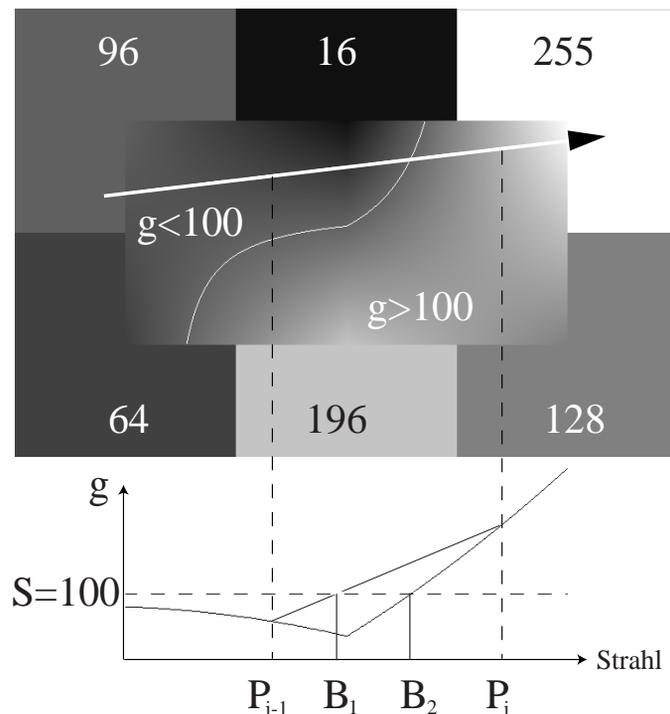


Abbildung 10: Lineare Interpolation der Grauwerte an den Abtastpositionen und Anwendung von Gleichung 2 (Seite 21) erzeugt die Objektgrenze an der Stelle B_1 . Durch Bisektion der Strecke $\overline{P_{i-1}P_i}$ wird eine bessere Approximation des tatsächlichen Oberflächenpunktes bei B_2 erzielt. Man beachte den nichtlinearen Verlauf der Grauwerte entlang des Abtaststrahls und die sprunghafte Änderung an der Stelle, wo sich die Umgebung der Abtastposition ändert (Unstetigkeit in der 1. Ableitung).

Bei einer realen Segmentation treten verschiedene Problembereiche auf wie überlappende Intensitätsbereiche benachbarter Objekte oder manuell getrennte Objekte mit demselben Schwellwert. Bei dem in Abschnitt 5.2.4 behandelten MR-Datensatz wurden z. B. zunächst die Gesamtheit der Organe (hier als Weichteile bezeichnet) mit einem Intensitätsbereich von 45-255 vom Hintergrund getrennt und anschließend das Gehirn mit einem Bereich von 61-255 definiert (Abbildung 11). Eine Zerlegung der Hirnsubstanz in die einzelnen Hirnwindungen ist überhaupt nur manuell möglich, da im Bildmaterial keinerlei Intensitätsunterschiede vorhanden sind.

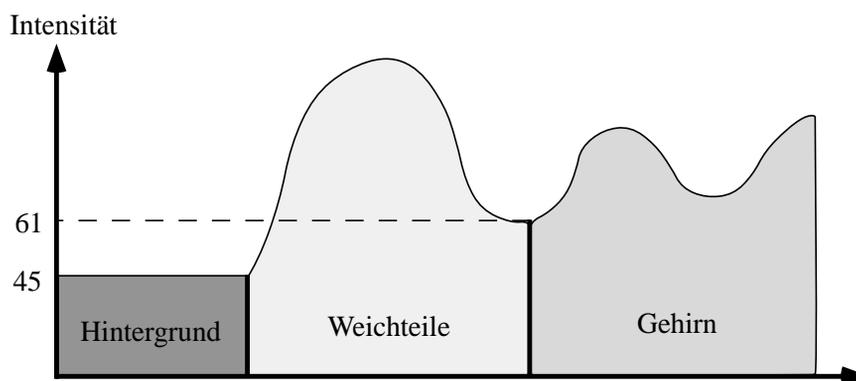


Abbildung 11: Die Intensitätsbereiche der Weichteile (45-255) und des Gehirns (61-255) überlappen sich: Die Lokalisation der Grenzfläche zwischen Weichteilen und Gehirn ist schwierig.

Im einzelnen können folgende Situationen auftreten:

1. Die Intensitätsbereiche T_i und T_{i-1} der Objekte O_i und O_{i-1} an den Abtastpositionen P_i und P_{i-1} grenzen aneinander. Das ist die optimale Situation, bei der es eine gemeinsame Grenzfläche für die Objekte gibt (Abbildung 12, links).
2. Die Intensitätsbereiche sind disjunkt und berühren sich nicht. Es entstehen zwei Oberflächen und eine Lücke, in der kein Objekt definiert ist. Dieser Fall ist unproblematisch, weil jeweils die vom Betrachter weiter entfernte Oberfläche dargestellt wird (Abbildung 12, Mitte).
3. Die Intensitätsbereiche überlappen sich, aber zumindest einer der beiden Bereiche erzeugt eine Grenzfläche (Abbildung 12, rechts).
4. Die Intensitätsbereiche überlappen sich derart, daß keine Grenzfläche zwischen den benachbarten Voxeln entsteht. Diese Situation tritt bei Verwendung morphologischer Operatoren (Erosion und Dilation) und bei manuell segmentierten Objekten wie etwa den Hirnwindungen auf.

Bis auf Situation 4 kann die Grenzfläche zwischen zwei Objekten immer durch Interpolation oder Bisektion im Subvoxelbereich bestimmt werden:

1. Liegt der Intensitätswert g_i innerhalb des Schwellwertbereichs von Objekt O_i und g_{i-1} außerhalb (Fälle 1, 2), so definiert der untere Schwellwert von Objekt O_i die Grenzfläche, falls $g_i > g_{i-1}$, andernfalls ist es der obere Schwellwert.

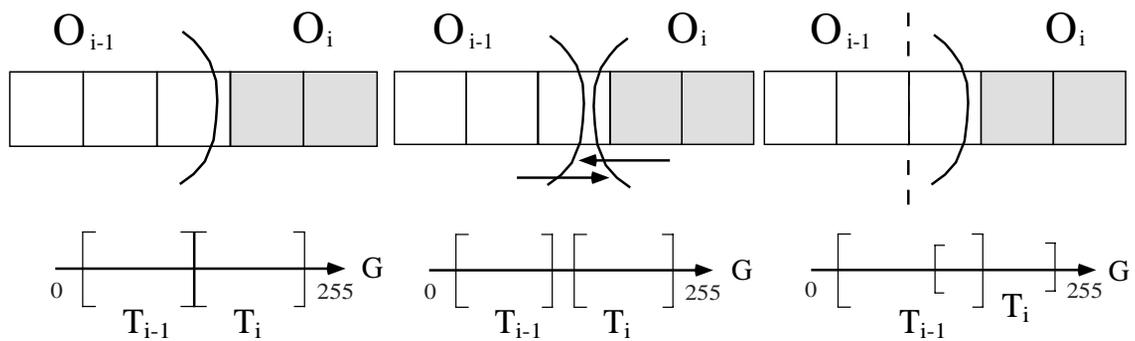


Abbildung 12: In Abhängigkeit der Intensitätsbereiche aneinandergrenzender Objekte ergeben sich unterschiedliche Situationen für den Verlauf der Grenzfläche (siehe Text). Bei disjunkten Bereichen (Mitte) ist die Wahl der Oberfläche von der Blickrichtung (Pfeile) abhängig.

2. Erzeugt der Schwellwertbereich von Objekt O_i keine Grenzfläche (Situation 3), so wird derselbe Test mit dem Objekt O_{i-1} durchgeführt. Befindet sich für O_{i-1} nun g_{i-1} im Schwellwertbereich und g_i nicht, so wird die Grenzfläche durch O_{i-1} erzeugt (Abbildung 12 rechts).
3. Andernfalls kann aus den Schwellwerten für die Objekte O_i und O_{i-1} keine Grenzfläche abgeleitet werden (Situation 4). In dieser Situation werden die Werte g_i und g_{i-1} durch die in Abschnitt 3.5 beschriebene Interpolation der Attributwerte erzeugt.

3.4 Berechnung der Oberflächennormalen

Für Oberflächendarstellungen der Objekte ist die Bestimmung der Oberflächennormalen von größter Bedeutung. Hier zeigt sich der große Unterschied zur Polygonwelt. Bei den aus Dreiecken zusammengesetzten polygonalen Objekten können die Normalvektoren aus den Koordinaten der Eckpunkte der Dreiecke direkt berechnet werden (siehe Abschnitt 3.8.1). Für benachbarte Dreiecke werden häufig die Normalvektoren an den gemeinsamen Eckpunkten durch Mittelung berechnet, um die facettenhafte Erscheinung der Oberfläche zu reduzieren. Bei der Abtastung der Dreiecke für die Darstellung auf der Bildebene müssen die Normalvektoren dann durch Interpolation bestimmt werden. Diese Techniken sind wohl bekannt.

In der Volumenwelt lassen sich so für die Volumenelemente nur 6 diskrete Oberflächenrichtungen erzeugen. Es sind zahlreiche Varianten, die die Oberflächennormale aus dem 3D-Kontext der Grenzfläche oder dem Z-Puffer ableiten, veröffentlicht worden. Ein Vergleich dieser Techniken findet sich in [Yagel *et al.* 1992b]. Mit der Entwicklung des Grauwert-Gradienten-Verfahrens wurde es möglich, Oberflächenneigungen mit feinen Abstufungen und mit großer Variabilität zu erzeugen. In [Pommert *et al.* 1990; Tiede *et al.* 1990] wurde gezeigt, daß die Berechnung der Oberflächennormalen aus dem Grauwertgradienten allen anderen Verfahren deutlich überlegen ist. Bei den bisher veröffentlichten Verfahren für segmentierte Volumendaten, die die Objektfläche nicht im Subvoxelbereich bestimmen, erfolgt die Berechnung der Oberflächennormalen an der Abtastposition gemäß Gleichung 1 (Seite 11).

Die Bestimmung der Oberflächenneigung eines Objekts im Subvoxelbereich erfordert eine Interpolation der Grauwerte, aus denen die Neigung der Oberfläche abgeleitet wird. Für Interpolationen höherer Ordnung, wie der kubischen Spline-Interpolation, kann der Gradientenvektor

analytisch aus der 1. Ableitung der Interpolationsfunktion berechnet werden. Für die trilineare Interpolation ist dies nicht sinnvoll, da deren 1. Ableitung an den Voxelgrenzen nicht stetig ist und Artefakte in der Darstellung die Folge sein könnten. Statt dessen werden die Komponenten des Gradientenvektors am Ort (x, y, z) auf folgende Weise approximiert:

$$G_x = \frac{f(x + \Delta x, y, z) - f(x - \Delta x, y, z)}{2\Delta x}$$

$$G_y = \frac{f(x, y + \Delta y, z) - f(x, y - \Delta y, z)}{2\Delta y}$$

$$G_z = \frac{f(x, y, z + \Delta z) - f(x, y, z - \Delta z)}{2\Delta z}$$

wobei $f(\cdot)$ den interpolierten Intensitätswerten entspricht und G_x , G_y und G_z die partiellen Ableitungen in Richtung der x -, y - und z -Achse bilden. Die Oberflächennormale ergibt sich durch Normalisierung der Gradientenkomponenten mit dem Betrag des Gradientenvektors (vgl. Gleichung 1, Seite 11). Für isotrope Volumendaten ist $\Delta x = \Delta y = \Delta z$. Der daraus resultierende konstante Nenner des Differenzenquotienten hat keinen Einfluß auf die hier interessierende Richtung des Gradientenvektors, sondern bewirkt lediglich eine Skalierung des Gradientenbetrags, so daß für die Berechnung als Nenner „1“ angenommen wird und damit die Division entfällt. Die Wahl von Δx , Δy und Δz hat erheblichen Einfluß auf die Darstellung. Kleine Werte betonen die hohen Frequenzen in den Daten, wodurch auch kleine Variationen der Oberflächenneigung sichtbar werden. Allerdings wird damit auch häufig das Rauschen in den Daten betont, während große Werte eine Glättung der Oberfläche bewirken, wodurch kleine Details möglicherweise verlorengehen [Schroeder *et al.* 1998, Seite 234]. In vielen Fällen stellt der Wert „1“ eine praktikable Lösung dar. In der in dieser Arbeit entwickelten Datenstruktur (Abschnitt 3.11) kann dieser Wert objektabhängig variiert werden.

Eine Verbesserung der Approximation der Oberflächennormalen kann durch Berücksichtigung aller Volumenelemente in der 18er- oder 26er-Nachbarschaft des Oberflächenvoxels erreicht werden. Die Komponente G_x des Gradientenvektors wird dann wie folgt berechnet:

$$G_x = \sum_{i=-1}^1 \sum_{j=-1}^1 \sum_{k=-1}^1 W_x(i, j, k) f(x + i, y + j, z + k)$$

Dabei repräsentiert $f(\cdot)$ wieder die interpolierten Intensitätswerte und W_x den Filterkern für die Gewichtung der Intensitätswerte in die x -Richtung. Die Berechnung von G_y und G_z erfolgt analog mit entsprechenden Filterkernen W_y und W_z (Abbildung 13).

Ein gebräuchlicher Filterkern ist der Zucker-Hummel-Operator [Ballard, Brown 1982, Seite 83], der eine Kugel approximiert. Die Kugelform des Operators erscheint sinnvoll, da dadurch keine der resultierenden Richtungen des Gradientenvektors und damit der Oberflächennormalen bevorzugt wird. Die Gewichtungsfaktoren des Operators entsprechen den Abständen vom Mittelpunkt zur Oberfläche einer Kugel (Abbildung 13, links). Diese Gewichtung entspricht aber nur in erster Näherung einem kugelförmigen Filterkern, weil die Gewichtung den jeweiligen

Volumenanteil des Kugelabschnitts im betreffenden Voxel nicht korrekt berücksichtigt. In der vorliegenden Arbeit wurde die Approximation deshalb entsprechend modifiziert (Abbildung 13, rechts). Die Herleitung der Werte für eine anteilige Volumengewichtung befindet sich im Anhang A.

	k=0			k= -1 und k=1			k=0			k= -1 und k=1		
j= -1	$-1/\sqrt{2}$	0	$1/\sqrt{2}$	$-1/\sqrt{3}$	0	$1/\sqrt{3}$	-F18	0	F18	-F26	0	F26
j=0	-1	0	1	$-1/\sqrt{2}$	0	$1/\sqrt{2}$	-F6	0	F6	-F18	0	F18
j=1	$-1/\sqrt{2}$	0	$1/\sqrt{2}$	$-1/\sqrt{3}$	0	$1/\sqrt{3}$	-F18	0	F18	-F26	0	F26
	i= -1	i=0	i=1				i= -1	i=0	i=1			

Abbildung 13: Filterkern W_x für die Grauwertgradientenberechnung: Zucker-Hummel-Operator mit $F_6 = 1$, $F_{18} = 0.707107$ und $F_{26} = 0.577350$ (links), Volumengewichtung mit $F_6 = 0.942908$, $F_{18} = 0.508789$ und $F_{26} = 0.171782$ (rechts); W_y und W_z ergeben sich durch Vertauschen von i und j bzw. i und k .

Aus der Tabelle 1 läßt sich ersehen, daß mit dem Filterkern aus den Volumenanteilen 89.6% der Gradienteninformation mit nur 18 Nachbarn errechnet wird, beim Zucker-Hummel-Operator hingegen nur 75.8%. Die 8 Eckvoxel, die einen Anteil von 30.8% an der Gesamtberechnungszeit des Gradienten haben, beeinflussen das Ergebnis bei der Volumengewichtung nur zu 10.4%, so daß für eine beschleunigte Darstellung auf sie verzichtet werden kann.

	Berechnungszeit		Mittelpunktabstand		Volumengewichtung	
F_6	$\frac{6}{26}$	(23.1%)	1	(31.4%)	0.942908	(43.1%)
F_{18}	$\frac{12}{26}$	(46.1%)	$\frac{1}{\sqrt{2}} = 0.707107$	(44.4%)	0.508789	(46.5%)
F_{26}	$\frac{8}{26}$	(30.8%)	$\frac{1}{\sqrt{3}} = 0.577350$	(24.2%)	0.171782	(10.4%)

Tabelle 1: Gewichtungsfaktoren der Filterkerne aus Abbildung 13. Die Prozentangaben entsprechen dem jeweiligen Anteil am Gesamtergebnis und ergeben sich aus $6F_6 + 12F_{18} + 8F_{26} = 100\%$. Bei der Volumengewichtung werden die Eckvoxel zugunsten der 6er-Nachbarn des Zentralvoxels deutlich geringer bewertet.

Für undurchsichtige Oberflächen wird der Grauwertgradient nur einmal berechnet. Beim gradientengewichteten Volume-Rendering dagegen muß er für jede Position entlang des Abtaststrahls bestimmt werden, so daß sich der Rechenaufwand hier besonders stark auswirkt. Bei Vorausberechnung und Speicherung des Gradientenvektors für jedes Volumenelement in einem Zusatzvolumen kann die Anzahl der Interpolationen während der Strahlabtastung auf drei, d. h. für jede Komponente des Gradienten eine, reduziert werden. Der zusätzlich erforderliche Speicherbedarf von 12 Byte pro Voxel (3 Gleitkommazahlen) übersteigt die Nutzdaten bei weitem (z. B. 2 Byte pro Voxel für CT-Daten und 1 Byte pro Voxel für die Speicherung der Objektinformation) und ist für große Volumina nicht möglich. Durch Kompression der Komponenten kann der Bedarf auf 3 Byte pro Voxel reduziert werden. Die Folge sind aber Verluste in der Variabilität der möglichen Neigungswinkel der Oberfläche, die sich negativ auf die Darstellungsqualität auswirken können.

3.5 Interpolation von Attributwerten

Ein großes Problem stellen die Oberflächenpositionen dar, an denen die Klassifikatoren der benachbarten Objekte auf dem Abtaststrahl keine Grenzfläche erzeugen, weil die Segmentation z. B. manuell durchgeführt wurde oder morphologische Operatoren angewendet wurden. Eine Berechnung der Oberflächenlage und der Orientierung der Oberfläche aus den Grauwerten ist daher sinnlos.

Da die Attributwerte im Gegensatz zu den Originalgrauwerten keinen Partial-Volumen-Effekt mehr aufweisen, ermöglichen sie nur wenige unterschiedliche Orientierungen der Oberfläche. In [Bomans 1994] wird daher eine Glättung der Attributwerte mit einer Gauß-Verteilung vorgeschlagen. Dazu wird ein zusätzliches Grauwertvolumen erzeugt und jedem Voxel, das zu einem bestimmten Objekt gehört, ein konstanter Grauwert G zugeordnet. Alle anderen Voxel erhalten den Wert „0“. Anschließend wird das Volumen mit der Gauß-Verteilung gefiltert. Je nach Parametrisierung des Gauß-Filters entsteht dabei eine mehr oder weniger breite Region ansteigender Intensität an der Objektgrenze. Als Schwellwert für die Objektbestimmung im Subvoxelbereich wird $G/2$ verwendet. Die Oberflächennormale wird wie im Abschnitt 3.4 beschrieben aus dem Grauwertgradienten im Gauß-Volumen abgeleitet. Da die Gauß-Filterung sehr weiche Übergänge liefert, werden die Resultate ähnlich glatt wie bei Objekten, die mit Schwellwerten segmentierbar sind. In [Gibson 1998] wird anstelle der Glättung mittels eines Gauß-Filters ein Volumen erzeugt, das für jedes Voxel den kürzesten Abstand zur Objekt-oberfläche enthält. Aus diesem Abstandsvolumen lassen sich insbesondere kleine Objekte gut rekonstruieren.

Ein Nachteil dieser Verfahren besteht darin, daß ein zusätzliches Volumen und damit entsprechend mehr Speicher benötigt wird. Außerdem erfordert jede Änderung eines Objektes die erneute Berechnung des Zusatzvolumens. Eine direkte Berechnung während der Strahlverfolgung scheidet wegen des Rechenzeitaufwandes aus.

In der Regel handelt es sich jedoch nur um vereinzelte Orte im Datenvolumen, an denen keine schwellwertbasierte Grenzfläche bestimmt werden kann, so daß es sinnvoller erscheint, diese Stellen ausfindig zu machen und die Grenzfläche lokal auf andere Weise zu bestimmen. In dieser Arbeit wurde deshalb ein einfaches Interpolationsverfahren auf der Basis der Attributvolumina entwickelt: In der $2 \times 2 \times 2$ -Umgebung einer Abtastposition können maximal acht unterschiedliche Attributwerte, also acht verschiedene Objekte, aneinandergrenzen. Bei realen Daten sind es aber selten mehr als zwei oder drei unterschiedliche Attributwerte. Für jeden dieser Attributwerte werden in einem $2 \times 2 \times 2$ -Hilfswürfel diejenigen Voxel mit „1“ besetzt, die in der Umgebung der Abtastposition denselben Attributwert enthalten wie der zu testende Attributwert, die übrigen Voxel mit „0“. Anschließend wird aus dem Hilfswürfel ein Intensitätswert durch trilineare Interpolation berechnet, wobei die Bruchteile x' , y' und z' der Koordinatenwerte der Abtastposition $P = (x, y, z)$ als Koeffizienten für die Interpolation dienen, mit $x' = x - \lfloor x \rfloor$, $y' = y - \lfloor y \rfloor$ und $z' = z - \lfloor z \rfloor$. Die Abtastposition gehört dann zu demjenigen Objekt, bei dem die Interpolation im Hilfswürfel einen Wert größer oder gleich 0.5 erzeugt. Man kann leicht überlegen, daß es unabhängig von der Zahl und Anordnung der verschiedenen Objekte in der Umgebung der Abtastposition höchstens ein Objekt geben kann, das diese Bedingung erfüllt. Falls keines der Objekte diese Bedingung erfüllt, wird dasjenige Objekt mit dem größten Interpolationswert gewählt, falls die Abtastposition zusätzlich auch im Schwellwertbereich des Objekts liegt, andernfalls wird die Position als nicht klassifiziert angenommen (vgl. Abschnitt

3.2, Seite 19).

Für die Berechnung der Oberflächennormale (siehe Abschnitt 3.4) wird ein entsprechend größerer Würfel erzeugt, um die Gradientenbestimmung ebenfalls mit interpolierten Werten durchführen zu können ($5 \times 5 \times 5$ Voxel bei trilinear Interpolation und $\Delta x = \Delta y = \Delta z = 1$). In Abbildung 14 sind einige der 256 unterschiedlichen Konfigurationen, die sich aus der Anordnung der Attributvoxel für ein Objekt ergeben können, und die sich daraus ergebenden Oberflächen gezeigt. Hierbei ist zu erkennen, daß die Zusammensetzung von Konfigurationen mit gleichen Seitenflächen eine geschlossene Oberfläche erzeugt.

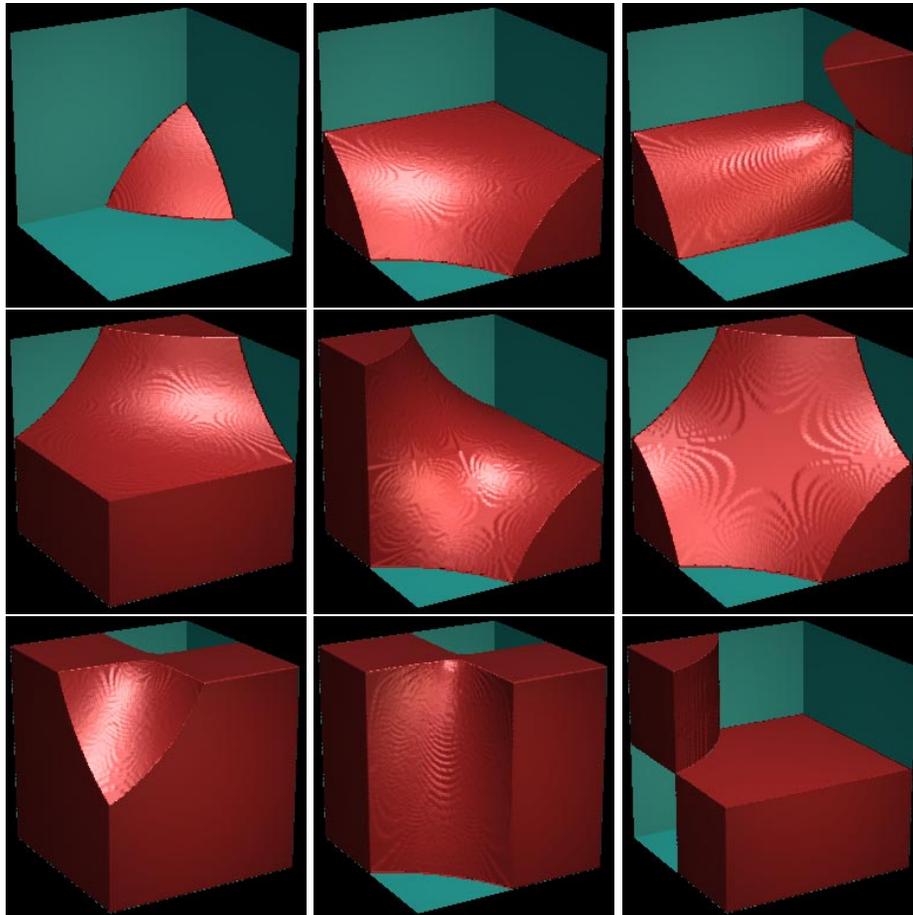


Abbildung 14: *Verschiedene Anordnungen der zu einem Objekt gehörenden Voxel in einer $2 \times 2 \times 2$ -Umgebung und der sich durch trilineare Interpolation ergebenden Oberfläche. Konfigurationen mit gleicher Seitenfläche lassen sich zu Objekten mit geschlossener Oberfläche zusammensetzen.*

Die Zahl der möglichen Abstufungen der mit diesem Verfahren erzeugten Oberflächenneigungen ist deutlich geringer als bei der Grauwert-Gradienten-Methode. In gewissem Maße kann diese Variabilität durch Anwendung höherer Interpolationen in einer größeren Umgebung, z. B. $4 \times 4 \times 4$ für Spline-Interpolationen, verbessert werden. Außerdem ist für dieses Verfahren kein weiteres Gesamtvolumen erforderlich, wie bei der Glättung der Attributwerte mit einem Gauß-Filter. Es kann zudem zur Laufzeit während der Strahlabtastung berechnet werden, so daß die Attributvoxel modifizierbar bleiben. Auf jeden Fall erscheint es sinnvoller, dieses Verfahren an den Orten einzusetzen, wo keine schwellwertbasierte Grenzfläche existiert, anstatt die unnatürlichen diskreten Voxelgrenzen zu verwenden.

3.6 Oberflächenbestimmung für multiparametrische Daten

Bisher wurde gezeigt, wie die Oberflächen für Objekte, die durch Grauwertbereiche charakterisiert sind, im Subvoxelbereich berechenbar sind. Für multiparametrische Daten, wie sie die Farbschnittbilder des Visible-Human-Projekts (vgl. 5.1.4) darstellen, gilt es, entsprechend zu verfahren.

Dazu wurden auf den Schnittbildern einige Volumenelemente eines zu segmentierenden Objekts interaktiv ausgewählt. Die Form der Verteilung der chromatischen Merkmale der ausgewählten Elemente im Merkmalsraum der drei Farbkomponenten Rot, Grün und Blau legte eine Beschreibung der Form als Ellipsoid nahe.

Ein Ellipsoid für die Charakterisierung eines Objekt wird folgendermaßen bestimmt [Schiemann *et al.* 1997]:

1. Interaktive Auswahl einer kleinen Anzahl (mindestens 4) von Volumenelementen, die zum Objekt gehören. Das Resultat ist ein Satz von Farbtupeln.
2. Entfernen von „Ausreißern“: Als „Ausreißer“ werden diejenigen Farbtupeln betrachtet, deren Abstand zum Schwerpunkt aller Farbtupel größer als ein wählbares Vielfaches des mittleren Abstands der Farbtupel ist.
3. Erneute Berechnung des Schwerpunktes aus den verbliebenen Farbtupeln. Dieser ist zugleich Zentrum $Z_{rgb} = (Z_r, Z_g, Z_b)$ des Ellipsoids.
4. Die Richtung der Achsen des Ellipsoids entsprechen den Eigenvektoren der aus den Farbtupeln ermittelten Kovarianzmatrix. Die durch die Eigenvektoren definierte Basis B_{rgb} in Form von drei orthogonalen Einheitsvektoren läßt sich in Form einer 3×3 Matrix M schreiben:

$$M = \begin{pmatrix} B_{r_1} & B_{g_1} & B_{b_1} \\ B_{r_2} & B_{g_2} & B_{b_2} \\ B_{r_3} & B_{g_3} & B_{b_3} \end{pmatrix}$$

5. Die Längen L_r , L_g und L_b der Achsen werden so bestimmt, daß alle nach der Entfernung der „Ausreißer“ verbliebenen Tupel innerhalb des Ellipsoids liegen.

Ein Objekt umfaßt dann alle diejenigen Voxel, deren Farbtupel innerhalb des betreffenden Ellipsoids liegen.

Die implizite Gleichung für ein reelles Ellipsoid lautet:

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 - 1 = 0 \quad \text{mit } a, b, c > 0$$

Um für ein beliebiges Farbtupel F_{rgb} mit den Komponenten F_r , F_g und F_b an einer Abtastposition zu prüfen, ob es innerhalb eines bestimmten Ellipsoids liegt, muß die Transformation $F'_{rgb} = (F_{rgb} - Z_{rgb}) M$ durchgeführt werden.

Die Anwendung der Ellipsoiden-Gleichung ergibt dann:

$$\left(\frac{F'_r}{L_r}\right)^2 + \left(\frac{F'_g}{L_g}\right)^2 + \left(\frac{F'_b}{L_b}\right)^2 \begin{cases} < 1, F_{rgb} \text{ liegt innerhalb} \\ = 1, F_{rgb} \text{ liegt auf dem Rand} \\ > 1, F_{rgb} \text{ liegt außerhalb} \end{cases}$$

Da ein Ellipsoid eine Fläche 2. Ordnung darstellt (Abbildung 15), muß der Quotient t aus Gleichung 2 (Seite 21), der den Schnittpunkt des Abtaststrahls mit der Oberfläche bestimmt, durch eine quadratische Gleichung berechnet werden. Der Aufwand ist also erheblich höher als für normale Schwellwerte. Auch hier gilt, wie bei der Bestimmung der Oberflächen auf Grauwertdaten, daß der Intensitätsverlauf der durch (trilineare) Interpolation berechneten Farbwerte zwischen den Abtastpositionen nicht unbedingt linear verläuft (Abbildung 10), so daß die Oberfläche durch Bisektion (Abschnitt 3.3) besser approximiert wird.

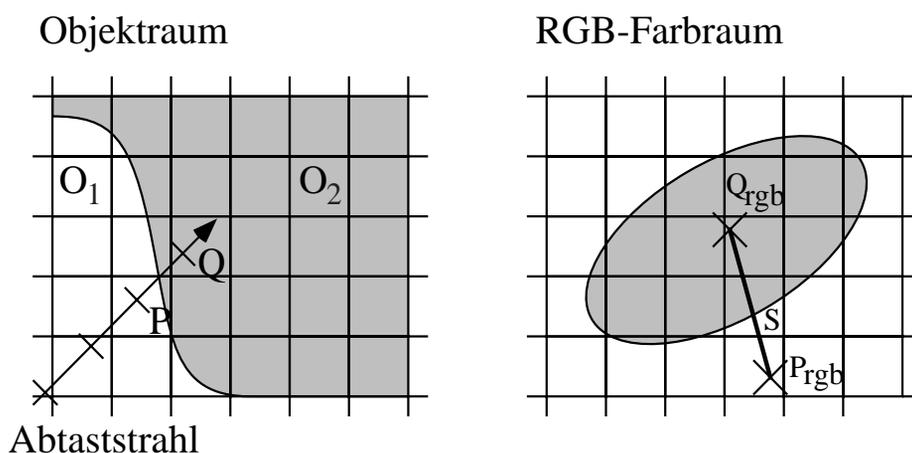


Abbildung 15: Bei der Klassifikation der Objekte durch Ellipsoide im RGB-Farraum wird die Objektfläche dort angenommen, wo entlang des Abtaststrahls die Strecke \overline{PQ} im selben Verhältnis geteilt wird, wie der Punkt S die Strecke der zugehörigen Farbwerte P_{rgb} und Q_{rgb} teilt (vgl. Abbildung 10).

Abbildung 16 zeigt die Lage der Ellipsoide, mit denen die Klassifikation verschiedener Organe im Kopf des Visible-Human durchgeführt wurde, im RGB-Farraum. Die numerische Spezifikation der Ellipsoide befindet sich in Tabelle 3 in Abschnitt 5.1.4.

3.7 Darstellung kleiner Strukturen

In den vorangegangenen Abschnitten wurde deutlich, daß die Position der Oberfläche im Subvoxelbereich bestimmt werden muß. Im Bereich der Kanten oder an dünnen Stellen eines Objekts kann es vorkommen, daß die gewählte Schrittweite der Abtastung sowohl in der Bildebene als auch entlang des Abtaststrahls zu groß ist und das Objekt verfehlt wird (Abbildung 17, links). Daraus resultieren Aliasing-Effekte, die sich besonders bei Animationen störend bemerkbar machen.

Das in dieser Arbeit entwickelte Verfahren der Objektbestimmung an einer beliebigen Abtastposition im Subvoxelbereich (Abschnitt 3.2) ermöglicht eine Erhöhung der Abtastfrequenz, d. h. eine beliebige Verkleinerung des Abstandes zwischen zwei Abtastpositionen, zur Reduktion der

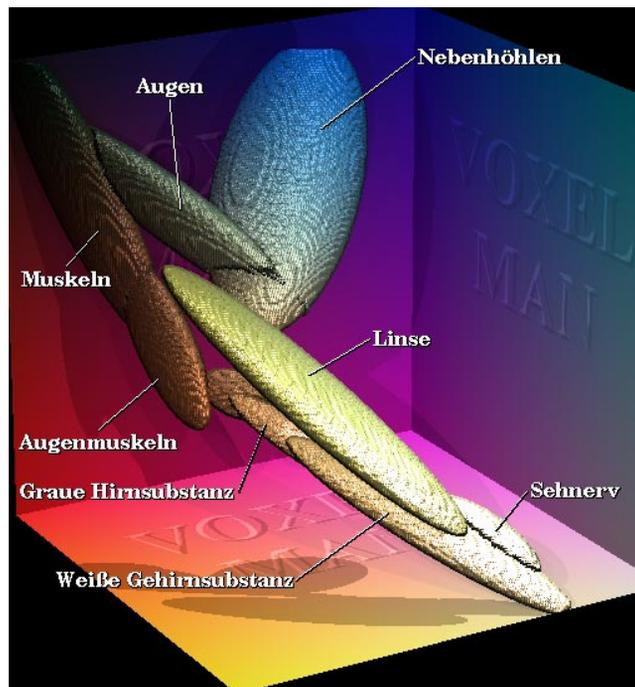


Abbildung 16: Klassifikation des Kopfes des Visible-Human durch Ellipsoide im 3D-Merkmalraum: Aus der Lage der Ellipsoide im RGB-Farbraum ist deutlich zu erkennen, daß einige Objekte durch disjunkte Ellipsoide charakterisiert sind, während sich beispielsweise die Ellipsoide für die Augen und Nebenhöhlen überschneiden.

Aliasing-Effekte (Abbildung 17, rechts), was die bisher publizierten Techniken für die Visualisierung segmentierter Volumendaten nicht leisten können.

Da das Vorhandensein und die Lage sehr dünner Stellen und scharfer Objektkanten nicht bekannt ist, müßte die Abtastrate so weit erhöht werden, daß eine Berechnung nicht mehr in vernünftiger Zeit durchgeführt werden kann. Ein Kompromiß zwischen Genauigkeit und Rechenaufwand liegt etwa in einer Schrittweite entlang der Abtastrichtung, die dem Abstand zweier Bildpunkte in der Projektionsebene entspricht. Überdeckt die Projektion eines Volumenelements beispielsweise vier Bildpunkte, was einer Vergrößerung um den Faktor 2 entspricht, so sollte der Abstand zweier Abtastpunkte entlang der Strahlrichtung $1/2$ Voxel betragen. Ebenso ist eine adaptive Überabtastung sowohl in der Bildebene als auch entlang des Abtaststrahls möglich, wobei die Schrittweite bei starken Änderungen der Farbe, der Oberflächennormalen oder des Z-Puffers benachbarter Bildpunkte entsprechend verkleinert wird. Auf eine weitergehende Behandlung dieser wohlbekannteten Techniken soll hier verzichtet werden (siehe hierzu [Watt 1992]).

3.8 Regionenbasierte Visualisierung

Der große Vorteil des Volumenmodells gegenüber der polygonalen Objektrepräsentation besteht darin, daß zu jedem Raumpunkt ein Intensitäts- oder Farbwert existiert. Diese Werte sind besonders in der radiologischen Diagnostik von Bedeutung und können auf beliebig orientierten Schnittflächen angezeigt werden (Abbildung 18). Neben der Möglichkeit, Objekte innerhalb einer Schnittregion zu entfernen [Höhne *et al.* 1987], wurde eine Methode für die regionenbasierte Visualisierung entwickelt, die sowohl die selektive Manipulation, wie etwa das

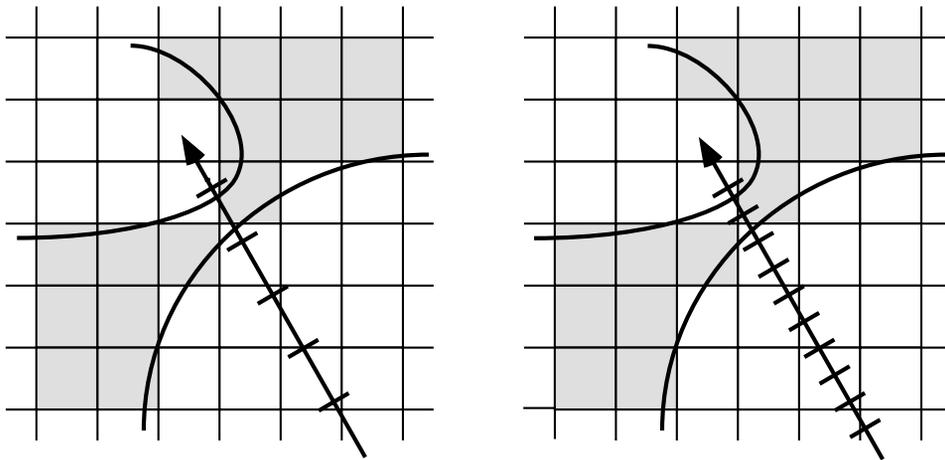


Abbildung 17: *Dünne Objekte werden bei zu großer Schrittweite verfehlt (links). Die Objektbestimmung im Subvoxelbereich ermöglicht eine Erhöhung der Abtastfrequenz, so daß auch kleine Strukturen erfaßt werden (rechts).*

Färben oder Entfernen von Objekten, als auch die freie Wahl der Darstellungsform, wie z. B. Oberflächendarstellung oder Maximum-Intensitäts-Projektion, beiderseits beliebig orientierbarer Schnittebenen ermöglicht.



Abbildung 18: *Selektives Schneiden: Auf den Schnittebenen sind die MR-Werte angezeigt; Ventrikelsystem, Augen und Sehnerv wurden nicht geschnitten.*

3.8.1 Definition von Regionen

Die Lage einer Schnittebene E im Raum ist geometrisch durch drei nicht kollineare Punkte $P_1 = (x_1, y_1, z_1)$, $P_2 = (x_2, y_2, z_2)$ und $P_3 = (x_3, y_3, z_3)$ eindeutig bestimmt. Die Gleichung einer Ebene lautet:

$$Ax + By + Cz + D = 0$$

Dabei bilden die Koeffizienten A , B , und C die auf der Ebene senkrecht stehenden Ebenenormale $\vec{N} = A\vec{e}_x + B\vec{e}_y + C\vec{e}_z$, die mit Hilfe des Kreuzprodukts berechnet wird (Abbildung 19):

$$\vec{N} = \frac{\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}}{\left| \overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3} \right|}$$

$$\overrightarrow{P_1P_2} = \vec{a} = (x_2 - x_1)\vec{e}_x + (y_2 - y_1)\vec{e}_y + (z_2 - z_1)\vec{e}_z$$

$$\overrightarrow{P_1P_3} = \vec{b} = (x_3 - x_1)\vec{e}_x + (y_3 - y_1)\vec{e}_y + (z_3 - z_1)\vec{e}_z$$

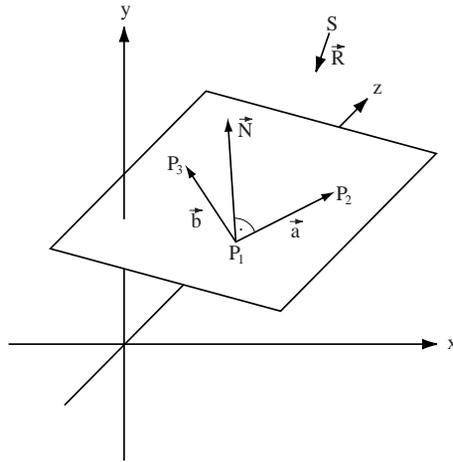


Abbildung 19: Eine Ebene ist durch drei nicht kollineare Punkte P_1 , P_2 und P_3 eindeutig bestimmt.

Daraus ergeben sich die (nicht normalisierten) Koeffizienten der Ebenengleichung als:

$$A = N_x = a_y b_z - a_z b_y$$

$$B = N_y = a_z b_x - a_x b_z$$

$$C = N_z = a_x b_y - a_y b_x$$

$$D = -(Ax_1 + By_1 + Cz_1),$$

da P_1 die Ebenengleichung erfüllt.

Die Hessesche Normalform erhält man durch Division der Koeffizienten mit $\sqrt{A^2 + B^2 + C^2}$. Der Schnittpunkt einer Ebene $E = (A, B, C, D)$ mit einem Abtaststrahl $r(t) = S + tR$ mit Startpunkt $S = (x, y, z)$ und Richtung $R = (u, v, w)$ ergibt sich aus der Berechnung des Parameters t [Badouel 1990]:

$$A(x + tu) + B(y + tv) + C(z + tw) + D = 0$$

$$\Rightarrow t = -\frac{Ax + By + Cz + D}{Au + Bv + Cw}$$

Im Fall $Au + Bv + Cw = 0$ sind Ebene und Sehstrahl koplanar und es gibt keinen Schnittpunkt. Ist $t < 0$, so liegt der Schnittpunkt hinter dem Startpunkt S und die Ebene wird für diesen Strahl nicht berücksichtigt. Für mehrere Ebenen wird die Länge des Abtaststrahls durch diejenige Ebene begrenzt, die das kleinste $t \geq 0$ erzeugt.

Das Volumen wird durch die so definierten Schnittebenen in Teilvolumina, hier Regionen genannt, untergliedert. Damit die Objekte in den verschiedenen Regionen unterschiedlich behandelt werden können, müssen die Regionen eindeutig identifizierbar sein. Eine solche Kennung läßt sich aus den Vorzeichen der einzelnen Ebenengleichungen zusammensetzen. Für einen beliebigen Punkt $P = (x, y, z)$ und eine Ebene E_i gilt:

$$A_i x + B_i y + C_i z + D_i \begin{cases} < 0, \text{ falls } P \text{ auf der negativen Seite der Ebene liegt} \\ = 0, \text{ falls } P \text{ genau auf der Ebene liegt} \\ > 0, \text{ falls } P \text{ auf der positiven Seite der Ebene liegt} \end{cases}$$

Ergibt beispielsweise die Ebenengleichung der Ebene 3 für den Startpunkt S des Abtaststrahls einen negativen Wert, so wird an der Bitposition 3 der Kennung eine „1“ gesetzt, andernfalls eine „0“. Nach Anwendung auf alle Ebenen E_i ist die Region, in der sich der Startpunkt S befindet, eindeutig identifiziert (Abbildung 20).

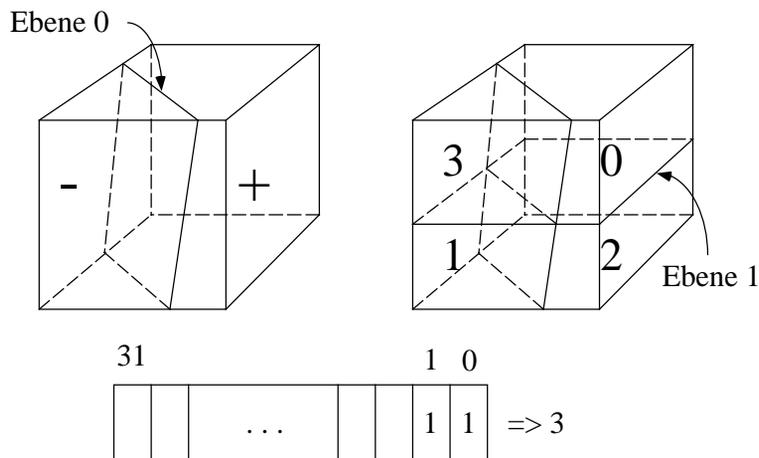


Abbildung 20: Erzeugung einer eindeutigen Regionenkennung: Der Wert 3 ergibt sich für alle Abtastpositionen, die sich auf der negativen Seite der Ebene 0 und der Ebene 1 befinden.

Die Regionenkennung muß für einen Strahlabschnitt, d. h. für den Bereich zwischen dem Startpunkt und der in Abtastrichtung am nächsten gelegenen Ebene nur einmal berechnet werden, nicht für jede Abtastposition. Es ist unmittelbar ersichtlich, daß es keine zwei verschiedenen Regionen mit derselben Kennung geben kann. Bei einer 32-Bit Rechnerarchitektur können so 32 Ebenen und 2^{32} Regionen unterschieden werden. Allerdings ist es nicht möglich, 32 Schnittebenen so anzuordnen, daß derartig viele Regionen entstehen.

3.8.2 Schneiden

Für jede Region wird eine Liste aller Objekte angelegt, die für jedes einzelne Objekt die modifizierbaren Parameter enthält wie die Transparenz oder die Farbe und ob das Objekt überhaupt

dargestellt werden soll oder geschnitten ist. Um ein Objekt in einer bestimmten Region zu entfernen, muß also lediglich das betreffende Objekt in der zugehörigen Objektliste als entfernt markiert werden.

Besondere Aufmerksamkeit erfordert die Bestimmung, ob ein Objekt an der Schnittstelle des Sehstrahls mit einer Schnittebene dargestellt werden soll oder nicht: Werden die Objekte auf den Schnittebenen unter Verwendung der Schwellwerte wie bei der Berechnung der Objektoberflächen bestimmt, können Löcher in der Darstellung der Schnittebene an den Stellen auftreten, wo Objekte mit disjunkten Schwellwerten benachbart sind (siehe hierzu Abbildung 12, Mitte) und kein Objekt gemäß dem Verfahren aus 3.2 bestimmt werden kann. An diesen Stellen wird der Strahl weiter bis zur nächsten Objektoberfläche verfolgt. Dies ist für radiologische Darstellungen der Schnittebene sehr störend. Verzichtet man dagegen auf die Schwellwerte, so entstehen voxelige Objektränder an den Schnittkanten.

Als Lösung für dieses Problem wurde folgende Methode entwickelt: Ist der Intensitätswert an der Schnittstelle des Abtaststrahls mit der Schnittebene keinem Objekt zuzuordnen, wird der Grauwert an dieser Stelle trotzdem dargestellt, es sei denn, mindestens eines der angrenzenden Voxel gehört zu einem deaktivierten Objekt, das also ebenfalls nicht dargestellt werden soll. Dadurch werden kleine Lücken zwischen benachbarten (aktiven) Objekten geschlossen, während die Übergänge zu deaktivierten Objekten und dem Hintergrund weiterhin im Subvoxelbereich bestimmt werden.

3.8.3 Realisierung von „Spanten“

Da das Segmentationsproblem für den allgemeinen Fall nicht gelöst ist, stellen die radiologischen Schnittbilder nach wie vor die wichtigste Informationsquelle für die Diagnostik dar. Ein Hilfsmittel, den mentalen Rekonstruktionsvorgang zu unterstützen, besteht in der Einbettung der Schichten in die 3D-Darstellung der segmentierbaren Objekte. Dazu werden einzelne Schnittebenen definiert und auf beiden Seiten der Ebene die Objekte entfernt, die den Blick auf den interessierenden Bereich verdecken. Die übrigen Objekte dienen der Orientierung zur Bestimmung der Betrachterposition und der Lage der Ebene im Raum (Abbildung 21).

Die Darstellung kann wahlweise im „radiologischen“ Modus erfolgen, bei dem auf der Schnittebene die tomographischen Grauwerte gezeigt werden, oder im „anatomischen“ Modus, bei dem die Schnittflächen durch Berechnung der Lichtreflexion ähnlich wie an den Objektoberflächen schattiert werden. Die Schattierungsparameter können unabhängig von denen für die Objektoberfläche spezifiziert werden, weil ein Objekt in der Regel im Inneren andere Reflexionseigenschaften besitzt als an der Oberfläche. Durch die Definition paralleler Schnitte kann die Dicke der Spanten variiert werden (Abbildung 52, Seite 72).

Auch für Darstellungsformen, die dem Betrachter keine Tiefeninformation geben, wie etwa die Maximum-Intensitäts-Projektion oder die durch Volume-Rendering erzeugten multiplen transparenten Oberflächen, ist es hilfreich, Spanten oder Schnittebenen zur besseren Orientierung in die Projektion zu integrieren. In den Regionen, die vor und hinter einer solchen Ebene liegen, werden die Objekte als MIP, künstliches Röntgenbild oder als transparente Oberflächen dargestellt, während auf der Ebene selbst die radiologischen Grauwerte gezeigt werden. Damit die Objekte auf der Ebene ihrerseits den Blick auf die dahinter liegenden Regionen nicht verdecken, müssen diese an- und abschaltbar sein bzw. transparent gemacht werden können. Das wird dadurch erreicht, daß die Visualisierungsparameter der Objekte in jeder Region und auf jeder Schnittebene unabhängig voneinander einstellbar sind.

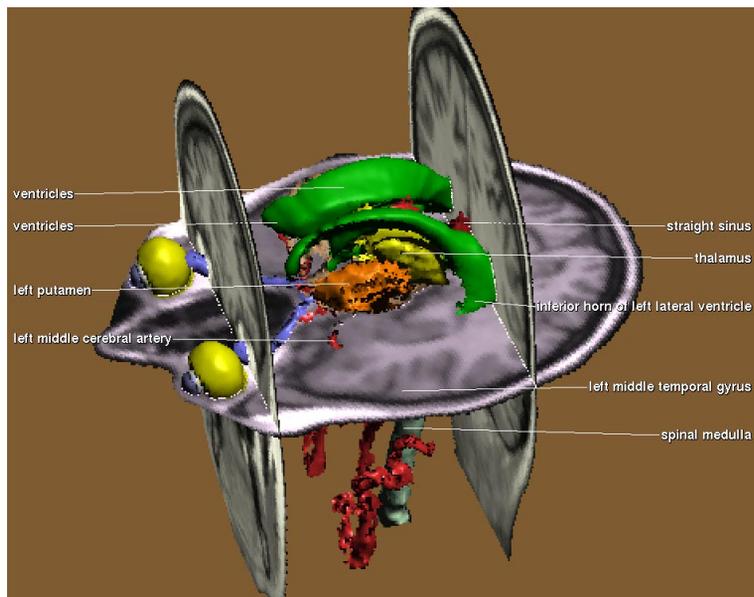


Abbildung 21: Zerlegung eines MR-Volumens in Spanten. Im „radiologischen“ Modus werden auf den Schnittebenen die tomographischen Grauwerte dargestellt.

3.8.4 Schneiden entlang des Sehstrahls

Neben der Definition frei orientierbarer Schnittebenen ist auch das Entfernen von Objekten entlang des Sehstrahls (Simulation eines Laserskalpells) ein wichtiges Hilfsmittel, um dreidimensionale Gegebenheiten zu analysieren und chirurgische Eingriffe zu simulieren [Pflesser *et al.* 1995]. Aus Sicht der Visualisierung kann ein solches Werkzeug durch ein weiteres Attributvolumen realisiert werden (vgl. Abschnitt 3.10). Dabei enthält das zusätzliche Volumen eine Marke für diejenigen Objektvoxel, die nicht dargestellt werden sollen. Diese Marke wird entlang des Sehstrahls objektabhängig und mit frei wählbarer Schnitttiefe gesetzt.

Um die als entfernt markierten Voxel (partiell) zu reaktivieren, muß lediglich die Markierung im Zusatzvolumen zurückgesetzt werden. Der Nachteil dieser Vorgehensweise besteht darin, daß für eine vergleichsweise einfache Funktion ein weiteres (binäres) Datenvolumen benötigt wird. Für einen durchschnittlichen Datensatz von 200^3 Voxeln erfordert das zusätzliche Attribut 0,95 MB mehr Speicher.

3.8.5 Freie Auswahl des Darstellungsverfahrens

Die regionenbasierte Visualisierung ermöglicht nicht nur das selektive Entfernen von Objekten, sondern erlaubt darüber hinaus, die Objekte in den verschiedenen Regionen auf unterschiedliche Weise darzustellen. So kann z. B. eine Region die Objekte als Oberflächen zeigen, eine andere als Maximum-Intensitäts-Projektion (MIP) oder simuliertes Röntgenbild und eine dritte in Form multipler transparenter Oberflächen (Volume-Rendering) (Abbildung 22). Dazu erhält jede Region einen Modus, der angibt, wie die betreffende Region dargestellt werden soll, sowie die für unterschiedlichen Darstellungsarten erforderlichen Visualisierungsparameter, wie den Kontrast für eine MIP oder Wichtungstabellen für die Grauwerte und Gradienten zur Berechnung der Opazität beim Volume-Rendering. Durch die Kopplung der Darstellungsparameter an die Objekte und Regionen können vielfältige unterschiedliche Abbildungen mit nur wenigen Grundfunktionen realisiert werden.

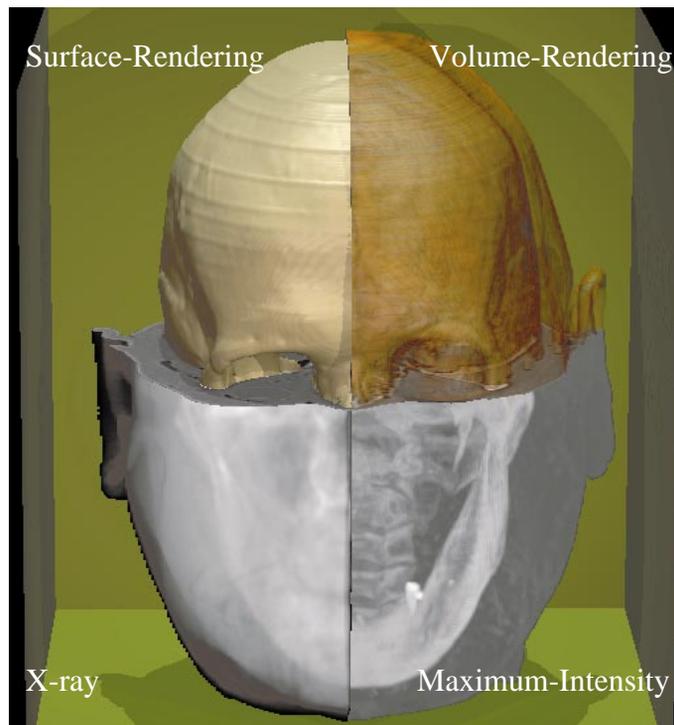


Abbildung 22: *Regionenbasierte Visualisierung: Die Objekte in den durch Schnittebenen definierten Regionen können als Oberflächen, Maximum-Intensitäts-Projektion, mit multiplen transparenten Oberflächen oder als simuliertes Röntgenbild dargestellt werden.*

3.9 Texturierung der Objektoberfläche

Ein wichtiges Hilfsmittel, um dreidimensionale Darstellungen realistischer erscheinen zu lassen, ist die Texturierung. Hierbei werden die verschiedenen Parameter, die in das Beleuchtungsmodell eingehen und den visuellen Eindruck beeinflussen, durch geeignete Funktionen so modelliert, daß ein möglichst naturgetreues Erscheinungsbild entsteht. Die wichtigsten Parameter eines Objektes, die sich für eine Texturierung eignen, sind:

- die Farbe eines Objektes;
- die Oberflächennormale, die die Reflexion des Lichts beeinflusst;
- die Transparenz, also die Lichtdurchlässigkeit der Oberfläche;
- der Ort der Oberfläche, mit dem die Geometrie des Objekts verändert wird.

Für künstliche Szenen und einfache, meist aus geometrischen Primitiven aufgebaute und mit Polygonen repräsentierte Objekte gibt es zahlreiche Verfahren, die Oberfläche wirklichkeitsgetreu zu modellieren. Hierzu gehören die Projektion zweidimensionaler Photographien auf die Objektoberfläche zur Bestimmung der Farbe oder auch dreidimensionale funktionale Texturen, mit denen sich Holzmaserungen u.ä. modellieren lassen ([Heckbert 1986; Perlin 1985; Watt 1992]). Auf natürliche Objekte in der Volumenrepräsentation sind diese Verfahren aber nicht ohne weiteres übertragbar. So kann z. B. die Gestalt der Oberfläche eines Objektes praktisch nicht modelliert werden, da sich diese erst aus dem Zusammenhang der benachbarten Voxel ergibt. Zudem sind medizinische Datensätze individuell, so daß für jeden neuen Datensatz die Texturparameter erneut bestimmt werden müßten ([Dahlmans 1997]).

Eine sinnvolle Anwendung für Volumendaten ergibt sich daher nur, falls deckungsreiche Texturvolumina zur Verfügung stehen, die zu jedem Volumenelement entsprechende Texturparameter enthalten. Das ist normalerweise nicht der Fall. Eine Ausnahme bilden die photographischen Farbschnittbilder des Visible-Human-Datensatzes (siehe Abschnitt 5.1.4). Anstatt einem Objekt eine willkürliche Farbe zuzuordnen, die über die gesamte Objektoberfläche konstant ist, steht hier für jedes Volumenelement ein eigener Farbwert zur Verfügung, der in das Beleuchtungsmodell eingeht und damit sehr realitätsnahe Darstellungen der Oberflächen ermöglicht. Wird der Farbwert aber direkt am Schnittpunkt des Sehstrahls mit der Objektoberfläche durch trilineare oder höhere Interpolation ermittelt, so entsteht eine Mischfarbe aus den benachbarten Objekten, die nicht der tatsächlichen Objektfarbe entspricht. Um diesen Effekt zu vermindern, wurde ein Verfahren entwickelt, das den Punkt zur Bestimmung der Objektfarbe entlang der Richtung der Oberflächennormale in das darzustellende Objekt hinein verschiebt.

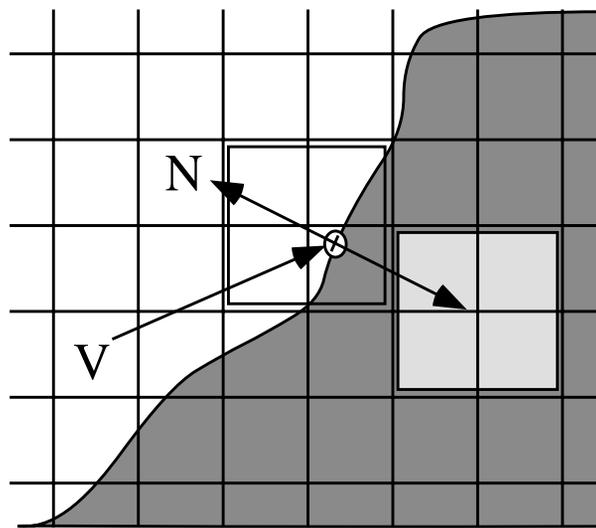


Abbildung 23: Bei der Interpolation der Objektfarbe direkt an der Objektoberfläche entsteht eine Mischfarbe aus den benachbarten Objekten (1). Durch Verschiebung des Interpolationszentrums entlang der Oberflächennormale N (nicht der Blickrichtung V) wird dieser Effekt reduziert (2).

Dieser heuristische Ansatz funktioniert nicht bei Strukturen, die kleiner als die für die verwendete Interpolation benötigte Umgebung sind. Bei der trilinearen Interpolation sind dies 2 Voxel, bei der kubischen Spline-Interpolation schon 4 Voxel, die das Objekt mindestens dick sein muß. Ist das Objekt dünner, würde der Berechnungspunkt durch die Verschiebung möglicherweise außerhalb des Objekts liegen, wodurch eine falsche Farbe berechnet würde. Für solche Objekte ist eine künstliche Farbe meist besser geeignet.

Das zuvor beschriebene Verfahren der Oberflächenfärbung mittels entsprechender Texturvolumina kann auch dazu verwendet werden, PET- oder SPECT-Werte auf der aus MR-Daten gewonnenen Hirnoberfläche mit Hilfe einer Farbtabelle darzustellen und so die Hirnaktivität an jedem Punkt sichtbar zu machen. Bei [Zuiderveld 1995] wird mit dieser Methode die Dicke des Knochens dargestellt. Dazu werden von der Knochenoberfläche aus die „Knochenvoxel“ entlang des Normalenvektors gezählt. Die Oberfläche wird dann entsprechend der Anzahl der gefundenen Voxel farblich markiert.

Neben der Modellierung der Objektfarbe wurde ein Verfahren entwickelt, den Normalenvektor einer Schnittebene zu variieren. Die normalerweise über die ganze Schnittebene konstante Ebenennormale erzeugt einen sehr glatten papierartigen Eindruck, der nicht einem realen Schnitt durch ein Objekt entspricht. Berechnet man den Grauwertgradienten auf der Schnittfläche, so entstehen an den Übergangsstellen zwischen verschiedenen Objekten stärkere Gradienten, während im homogenen inneren Bereich eines Objektes der Betrag des Gradienten verschwindend gering wird. Durch die gewichtete Ablenkung der Ebenennormale mit dem Grauwertgradienten an dem Schnittpunkt von Sehstrahl und Schnittebene entsteht ein realistischerer Eindruck der Schnittfläche (Abbildung 52, Seite 72). Die Modellierung der Komponenten des Normalenvektors \vec{N} an einem Punkt P durch den Grauwertgradienten $\vec{G}(P)$ auf der Ebene $E = (A, B, C, D)$ wird wie folgt berechnet:

$$\begin{aligned} N_x &= A + W \left(\left| \vec{G}(P) \right| \right) (G_x - A) \\ N_y &= B + W \left(\left| \vec{G}(P) \right| \right) (G_y - B) \\ N_z &= C + W \left(\left| \vec{G}(P) \right| \right) (G_z - C) \end{aligned}$$

Der endgültige Normalenvektor \vec{N} wird durch Normalisierung seiner Komponenten bestimmt. Die Berechnung des Grauwertgradienten erfolgt wie in Abschnitt 3.4 beschrieben. Die Komponenten A , B und C der Ebene und G_x , G_y und G_z des Gradienten seien zuvor normalisiert worden. Die Wichtung W des Betrags des Gradienten $\vec{G}(P)$ sei auf den Bereich 0 bis 1 beschränkt. Bei den in dieser Arbeit gezeigten Bildern wurden Werte zwischen 0.01 und 0.2 verwendet.

3.10 Visualisierung multiattributierter Volumina

Ein wesentlicher Gesichtspunkt der attributierten Volumenrepräsentation ist die Möglichkeit, jedem Voxel mehrere Attributwerte aus unterschiedlichen Domänen zuzuordnen. Ein multiattributiertes Voxel kann z. B. die Information beinhalten, zu welcher Hirnwindung es gehört, von welchem Blutgefäß es versorgt wird, welche Körperfunktion es steuert, ob es zu einem Risikoorgan gehört oder welche Strahlendosis es enthält. Jede Domäne erzeugt andersartige Voxelmengen. Innerhalb einer einzelnen Domäne können sich die Objekte nicht überschneiden, weil die Attributwerte disjunkt sind und ein Voxel nicht gleichzeitig zu verschiedenen Hirnwindungen gehören oder unterschiedliche Strahlendosen enthalten kann. Konzeptionell wurde diese Multiattributierbarkeit schon in [Stiehl 1987] erwähnt, bisher existiert jedoch kein System, das dieses Konzept konsistent umsetzt und eine entsprechende Visualisierung ermöglicht.

Die Objekte aus verschiedenen Domänen lassen sich mit den ihnen jeweils zugewiesenen Parametern für Farbe, Lichtreflexion, Transparenz etc. darstellen. Der volle Nutzen entsteht jedoch erst, wenn die Objekte aus mehreren Domänen miteinander verknüpft werden können. Dabei entstehen neue „zusammengesetzte“ Objekte aus den Schnittmengen der einzelnen Attributvolumina, weil die segmentierten Voxelmengen nicht deckungsgleich sind. So versorgt z. B. ein Blutgefäß mehrere Hirnwindungen, oder verschiedene Hirnareale sind an einer Körperfunktion beteiligt, oder die für eine Tumorbekämpfung erforderliche Strahlendosis überdeckt auch Risikoorgane, die nicht belastet werden dürfen. Bei einer generellen mengentheoretischen Verknüpfung der Objekte aus den unterschiedlichen Bereichen entstehen alle möglichen denkbaren

Teilmengen der ursprünglichen Objekte. Um diese differenziert visualisieren zu können, muß jede dieser Teilmengen einen eigenen vollständigen Satz von Darstellungsparametern erhalten, mit dem gesteuert werden kann, ob die Teilmenge aktiv oder inaktiv ist und ob sie farblich markiert oder transparent dargestellt werden soll. Eine Datenstruktur, die diese bisher nicht mögliche Funktionalität erlaubt, wird im Abschnitt 3.11 beschrieben.

Die Gestaltung einer graphischen Benutzeroberfläche für eine mengentheoretische Verknüpfung von Objekten aus beliebigen Domänen ist schwierig. Deshalb wurde hier ein einfacherer Weg beschritten: Der Anwender modifiziert (z. B. färbt oder entfernt) zunächst nur ein Objekt oder eine Objektgruppe aus einer Domäne (also keine zusammengesetzten Objekte), danach ein Objekt aus einer anderen Domäne, anschließend ein drittes und so fort. Als Darstellungsparameter werden jeweils diejenigen des zuletzt modifizierten Objekts benutzt. Die Darstellung hängt dann von der Reihenfolge ab, in der die Objekte z. B. eingefärbt wurden. Für den Benutzer ist der Vorgang intuitiv nachvollziehbar. Auf diese Weise lassen sich nicht alle Kombinationen realisieren, deshalb besteht daneben die Möglichkeit, über die im Anhang C beschriebene Spezifikationssprache jede beliebige Teilmenge der Objekte zu definieren und mit einem eigenen Satz von Parametern zu versehen.

Die potentielle Anzahl der neuen Voxelmengen nimmt mit der Anzahl der gleichzeitig verwendeten Domänen stark zu. Für drei Domänen mit jeweils 20 Objekten entstehen bereits $20^3 = 8000$ Schnittmengen mit dem erforderlichen Speicherbedarf. Tatsächlich überlappen sich aber nur ein Bruchteil der Objekte. Im Rahmen der Arbeit wurde deshalb ein Verfahren entwickelt, das nur die wirklich vorhandenen Teilmengen ermittelt und somit zu einer erheblichen Speicherreduktion führt und dadurch auch ein verbessertes Rechenzeitverhalten aufweist (siehe hierzu Abschnitt 4.2.3).

3.11 Datenstruktur für die multiattributierte und multiparametrische Visualisierung

In den vorangegangenen Abschnitten wurde eine Reihe neuer oder weiterentwickelter Algorithmen für die attributgesteuerte Visualisierung von Volumendaten beschrieben. Ein System für die objektbezogene freie Kombinierbarkeit dieser Verfahren in einem einzigen Bild existiert bisher nicht. Nachfolgend wird ein Datenstrukturkonzept beschrieben, das diese Funktionalität ermöglicht. Die Hauptkomponenten dieses Konzeptes bestehen aus Objekten, Domänen, Szenen, Regionen und Kameras, deren prinzipieller Zusammenhang in Abbildung 24 gezeigt ist.

- **Objekt**

Die zentrale Struktur ist das Objekt, eine durch die Segmentation erzeugte elementare Menge von Voxeln, die bezüglich ihrer Visualisierung dieselben Eigenschaften haben soll und nicht weiter differenziert werden kann. Neben Farbe, Reflexionsparameter und Texturinformation gehören auch der Klassifikator, mit dessen Hilfe die genaue Oberflächenposition bestimmt wird, und die Methode zur Berechnung der Oberflächennormalen zu diesen Eigenschaften.

- **Domäne**

In einer Domäne sind die zuvor beschriebenen elementaren Objekte zusammengefaßt. Eine notwendige Eigenschaft dieser Objekte ist es, daß sie disjunkt sind, d. h., sie dürfen

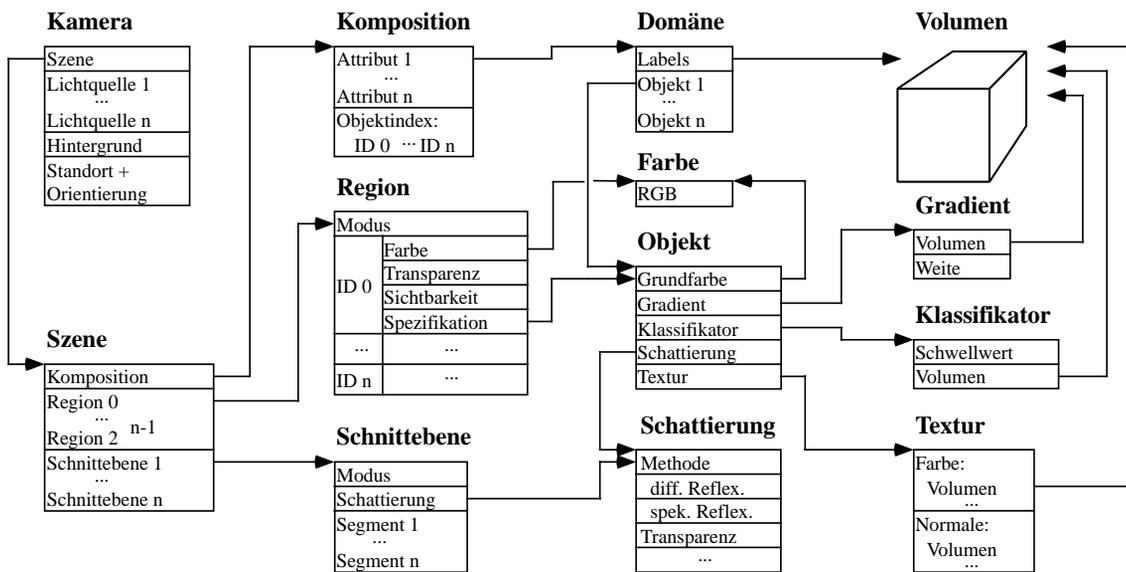


Abbildung 24: Datenstruktur für die 3D-Visualisierung multiattribuierter und multiparametrischer Volumendaten: Die Pfeile markieren Referenzen zu Instanzen der verschiedenen Komponenten. Von jeder Komponente können beliebig viele Instanzen existieren.

sich nicht überschneiden, damit ihnen ein eindeutiger Attributwert („Label“) zugeordnet werden kann. Beispiele für Domänen sind die Morphologie anatomischer Objekte wie Organe, Hirnwindungen, Knochensegmente und Blutgefäße, oder funktionale Areale der Großhirnrinde, Blutversorgungsgebiete der Arterien oder Risikoorgane in der Strahlentherapie. Aus praktischen Erwägungen wird bei der Segmentierung für jede Domäne ein eigenes Attributvolumen angelegt. Diese Vorgehensweise benötigt relativ viel Speicherplatz für die Volumina, weil in der Regel weniger Objekte segmentiert werden als Attributwerte zur Verfügung stehen. Dafür ist aber die Verwaltung der Objekte und Volumina erheblich einfacher, als wenn bei der Segmentierung bereits weitere (zukünftige) Domänen berücksichtigt werden müssen.

- **Komposition**

In einer Komposition werden Objekte aus den verschiedenen Domänen vereinigt. Durch die Überlappung der Objekte entstehen durch Schnittmengenbildung neue zusammengesetzte (komponierte) Objekte, deren megentheoretischer Wert dem Resultat einer „Und-Verknüpfung“ der einzelnen Attribute entspricht. Jedes dieser zusammengesetzten Objekte erhält eine eindeutige Kennung, um sie unterscheidbar zu machen (siehe auch Abschnitt 4.2.3). Dadurch sind alle in einer Komposition definierten (Teil-) Objekte in einer Szene unabhängig voneinander manipulierbar. Ein Beispiel für ein solchermaßen zusammengesetztes Objekt ist die Menge aller Voxel, die der linken unteren Stirnwindung des Gehirns **und** dem sekundären motorischen Cortex angehören **und** von den Endästen der linken mittleren Hirnarterie versorgt werden (Abbildung 25). Diese Voxelmenge könnte mit Hilfe der in dieser Arbeit entwickelten Beschreibungssprache (vgl. Abschnitt 4.1 und Anhang C.14) wie folgt spezifiziert werden:

```

set area *
  id=(( "Struktur"="linke untere Stirnwindung") and
    ("Funktion"="sekundärer motorischer Cortex") and
    ("Blutversorgung"=
      "Endäste der linken mittleren Hirnarterie"))
  color=yellow

```

Wurde ein Volumen nur hinsichtlich einer Domäne segmentiert, so existieren keine zusammengesetzten Objekte und die Kennung der Objekte in der Komposition ist identisch mit der Kennung in der Domäne.

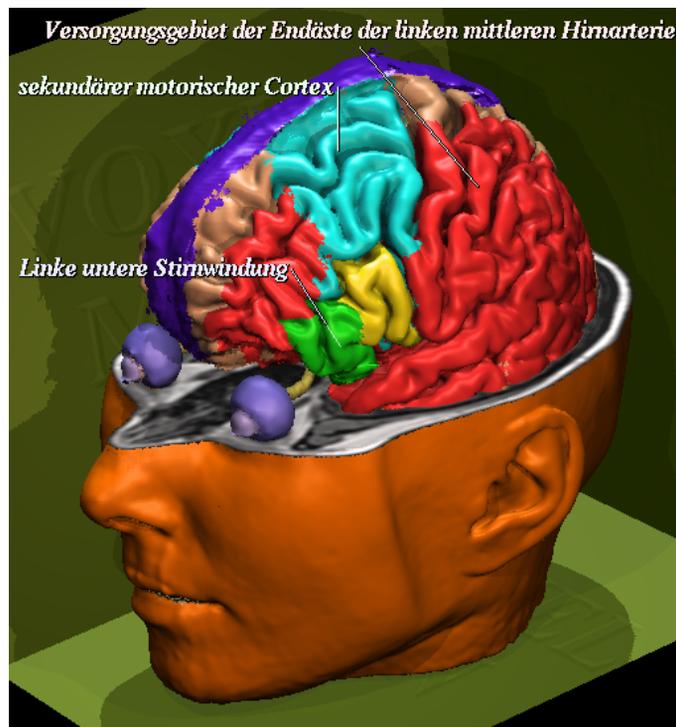


Abbildung 25: *Beispiel einer mengentheoretischen Verknüpfung aus verschiedenen Domänen: Die gelb markierte Voxelmengung ist Bestandteil der linken unteren Stirnwindung (grün) **und** wird von den Endästen der linken mittleren Hirnarterie (rot) versorgt **und** gehört zum sekundären motorischen Cortex (türkis).*

- **Szene, Regionen und Schnittebenen**

Eine Szene beschreibt den Zustand aller Objekte, also auch der Teilobjekte, die durch die Zusammenfassung verschiedener Domänen in einer Komposition entstanden sind. Der Raum, in dem sich die Objekte befinden, wird durch frei definierbare Schnittebenen in Sektoren bzw. Regionen unterteilt, in denen die Objekte individuell mit unterschiedlichen Visualisierungsmethoden dargestellt werden können. Dies wurde ausführlich in Abschnitt 3.8 behandelt. Schneiden sich mehrere Ebenen, so wird jede einzelne Ebene in verschiedene Segmente unterteilt. Um ein Objekt auf einem Segment sichtbar zu machen, auf einem anderen aber nicht, kann deshalb jedes Objekt auf jedem dieser Segmente unabhängig aktiviert bzw. deaktiviert werden.

- **Kamera**

Die Kamera ist ein Synonym für den Betrachter einer Szene. Die Abbildungsgeometrie wird durch Standort, Blickrichtung und Brennweite bestimmt. Da eine globale Ausleuchtung einer Szene, die aus jeder beliebigen Betrachtungsrichtung eine gleich gute Erkennbarkeit ermöglicht, praktisch nicht erreichbar ist, verfügt jede Kamera über eigene Lichtquellen und einen eigenen Szenenhintergrund, bestehend aus einer Hintergrundfarbe und künstlichen Objekten wie z. B. Seitenwände, die besonders für die Darstellung des Schlagschattens geeignet sind. Dadurch ist es möglich, dieselbe Szene gleichzeitig aus verschiedenen Richtungen, also mit mehreren Kameras, zu betrachten und dabei für jede Kamera eine optimale Beleuchtung der Szene und gegebenenfalls einen eigenen Hintergrund einzustellen.

Das Resultat des Projektionsvorgangs ist eine 2D-Bildmatrix. Neben den Farbwerten, die der Betrachter am Bildschirm sieht, werden zusätzliche Informationen an jeden Bildpunkt gebunden, um weitergehende Manipulationen zu ermöglichen (Abbildung 26). Dazu gehört der Attributwert des Objektes, so daß die sichtbaren Objekte mit der Maus direkt anwählbar sind und sich z. B. ohne erneutes Ray-Casting optisch hervorheben lassen. Durch Vertauschen der drei Farbwerte rot, grün und blau für alle Bildpunkte eines Objekts sind sechs unterschiedliche Einfärbungen des Objekts möglich, ohne daß sich die Reflexionseigenschaften wie etwa Glanzlichter ändern. (Voraussetzung ist, daß die Komponenten der Grundfarbe des Objektes alle verschieden sind. Ein rein weißes Objekt kann so nicht markiert werden.) Für eine beliebige Einfärbung eines Objekts muß die Schattierung aus dem Gradienten und den Beleuchtungsparametern erneut berechnet werden. Ist es hingegen gerade das Ziel, die Reflexionseigenschaften zu modifizieren, so kann das Beleuchtungsmodell mit den veränderten Parametern berechnet werden, weil die Oberflächennormale für jeden Bildpunkt gespeichert ist. Dies kann nur bei undurchsichtigen Objekten durchgeführt werden, weil jeweils nur ein Wert gespeichert wird. Für transparente Objekte ist ein erneutes Ray-Casting erforderlich, weil die Speicherung und Verwaltung einer unbekannt Anzahl transparenter Oberflächen an einem Bildpunkt sehr aufwendig wäre.

Der Z-Puffer, der für jeden Bildpunkt den Abstand der ersten sichtbaren Oberfläche zur Bildebene enthält, ermöglicht das schnelle Entfernen der sichtbaren Objekte, weil der Abtaststrahl nicht an der Grenze des Volumens beginnen muß, sondern erst an der Oberfläche des Objektes, das entfernt werden soll. Dadurch wird der Abtaststrahl erheblich kürzer, womit sich die Berechnungszeit reduziert. Daneben ermöglicht der Z-Puffer die kombinierte Darstellung der Volumenobjekte zusammen mit polygonalen Repräsentationen von künstlich modellierten Objekten oder Werkzeugen für die Spezifikation von Funktionen, die nicht in Echtzeit ausführbar sind, wie etwa die Berechnung einer neuen Projektion von einem anderen Betrachterstandort.

Nach Bedarf können weitere Informationen an jeden Bildpunkt gebunden sein, wie z. B. Begrenzungen für die Länge des Abtaststrahls, um eine schnellere Berechnung verschiedener Darstellungen aus derselben Blickrichtung zu ermöglichen (Abschnitt 4.2.2). Diese Struktur, die eine direkte Interaktion mit den dargestellten Objekten ermöglicht, wird als „intelligentes“ Bild bezeichnet [Höhne *et al.* 1995a].

Von jeder der vorgenannten Komponenten können beliebig viele Instanzen gleichzeitig existieren. Die einzige Einschränkung bildet der zur Verfügung stehende Speicher. So können z. B. mehrere Kameras existieren, die dieselbe oder verschiedene Szenen abbilden, oder für verschiedene Objekte werden dieselben Schattierungsparameter benutzt, jedoch mit unterschiedlichen Farben. Der Vorteil der Aufteilung verschiedener Parametergruppen zu eigenständigen Strukturen besteht darin, daß die Änderung eines Parameters sich auf alle Komponenten auswirkt, die die entsprechende Struktur referenzieren. Anstelle der Definition eigener Reflexionseigenschaften, Farben oder Schwellwerte für die verschiedenen Hirnwindungen existieren diese nur einmal, so daß z. B. eine Modifikation der Farbe die Darstellung aller Hirnwindungen beeinflußt. Durch diese Zusammenfassung der Parameter für eine ganze Objektgruppe wird zudem weniger Speicherplatz benötigt. Falls erforderlich können dennoch für einzelne Objekt individuelle Darstellungsparameter bereitgestellt werden.



Abbildung 26: „Intelligentes“ Bild: Jeder Bildpunkt enthält neben den Farbwerten auch die Information über das an der Stelle befindliche Objekt, den Tiefenpuffer und den Gradienten.

Der Hauptvorteil dieses Konzepts besteht in der großen Variabilität der Darstellung, die dadurch erreicht wird, daß jedes Objekt in jeder Region und auf jeder Schnittebene mit unterschiedlichen, für die jeweilige Anwendung oder Fragestellung am besten geeigneten Parametern visualisiert werden kann. Mit wenigen Funktionen entsteht so eine nahezu unbegrenzte Vielfalt verschiedenartiger Darstellungen. Ein extremes, medizinisch nicht unbedingt sinnvolles Beispiel ist in Abbildung 27 gezeigt.



Abbildung 27: *Beispiel für eine komplexe Komposition: Beliebige orientierte Schnittebenen, eingefärbte, partiell entfernte und transparente Oberflächen und verschiedene Darstellungsformen wie simulierte Röntgenbildprojektion und multiple Transparenz.*

4 Implementation

Die in Abschnitt 3 beschriebenen Verfahren wurden am Institut für Mathematik und Datenverarbeitung in der Medizin entwickelt und bilden den Visualisierungskern des VOXEL-MAN-Systems, das in Form eines Hirn- und Schädelatlases herausgegeben wurde [Höhne 1995]. Die Implementation erfolgte in ANSI-C unter Unix als Betriebssystem sowie mit OSF/Motif und X-Windows als graphischer Oberfläche, um eine möglichst weitgehende Hardwareunabhängigkeit zu erreichen. Das System ist zur Zeit auf Rechnern von DEC (Alpha und Mips Prozessoren), Sun, HP, SGI und PC-Systemen unter Linux lauffähig.

4.1 Benutzerschnittstelle

Als primäre Schnittstelle wurde eine Beschreibungssprache entwickelt, die die vollständige Steuerung und die Speicherung des Zustandes der Visualisierung in Textform ermöglicht (Abbildung 28). Diese Sprache ist durch folgende Eigenschaften charakterisiert:

1. Sie ist leicht implementierbar und erweiterbar für neue Funktionen.
2. Es ist die Unabhängigkeit von graphischen Oberflächen (z. B. OSF/Motif, SUN Openlook, Windows, Macintosh) gewährleistet, so daß eine Portierung auf andere Systeme leicht möglich ist.
3. Die Sprache ist skriptfähig, d. h., es können komplexe Anweisungen in Parameterdateien abgelegt und jederzeit zur Ausführung gebracht werden. Dies ist insbesondere bei der Erzeugung von Animationen wichtig, bei denen präzise numerische Angaben erforderlich sind, die mit einer graphischen Benutzeroberfläche nicht sinnvoll erstellt werden können.

4. Es gibt einen Hintergrundmodus, der es erlaubt, Filmsequenzen ohne jede Bildschirm-
ausgabe zu berechnen.
5. Viele semantische Abhängigkeiten können sicher und einfach behandelt werden. So ist
z. B. bei einer Objektspezifikation die Angabe einer nicht zuvor definierten Schattierungs-
methode unmöglich.
6. Der Zustand des Systems kann gesichert und wieder geladen werden. Die Speicherung
erfolgt in Klartext, weil bei einer binären Speicherung Abhängigkeiten von Maschinen
und Programmversionen entstehen, die eine spätere Modifikation unmöglich machen. Auf
diese Weise kann zu jedem berechneten Bild auch die Information gespeichert werden,
wie es erzeugt worden ist, so daß nach einem erneuten Programmstart alle Parameter
wie z. B. die Lage von Schnittebenen oder die Spezifikation von Beleuchtungsparamete-
ren nachträglich veränderbar sind. Das ist insbesondere für komplexe Kompositionen
wichtig, deren Spezifikation viele Schritte erfordern. Ebenso kann die Beschreibung ein-
es „alten“ Bildes mit jedem Texteditor an eine neue oder erweiterte Syntax angepaßt
werden.

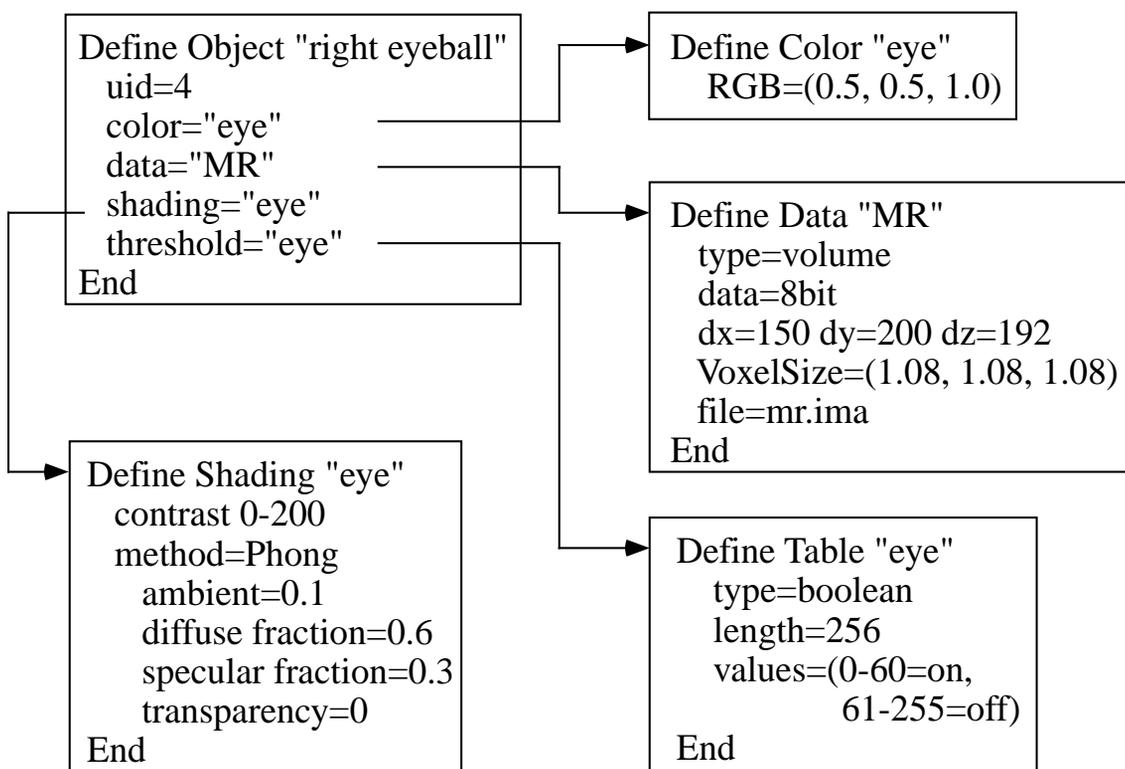


Abbildung 28: Ausschnitt aus der Definition eines Objektes mittels der Beschreibungssprache des Visualisierungskerns des VOXEL-MAN-Systems.

Die vollständige Syntax der Beschreibungssprache ist im Anhang C beschrieben. Für den normalen Anwender wurde im Rahmen des VOXEL-MAN-Projekts¹ von der Arbeitsgruppe eine graphische Benutzerschnittstelle konzipiert und implementiert, die den überwiegenden Teil der Steuerung der Visualisierung ermöglicht und eine Manipulation der Objekte sowohl im bildlichen als auch im symbolischen Kontext erlaubt (Abbildung 51, Seite 71). Diese graphische

¹VOXEL-MAN: <http://www.uke.uni-hamburg.de/institute/imdm/idv/forschung/vm/>

Schnittstelle kommuniziert ihrerseits über die Beschreibungssprache mit dem Visualisierungskern.

4.2 Performanz

Im folgenden Abschnitt wird auf drei Probleme eingegangen, die wesentlichen Einfluß auf die Geschwindigkeit der 3D-Darstellung haben:

- Der Zugriff auf die Randbereiche des Datenvolumens.
- Verfahren, um die Abtastung leerer Volumenbereiche zu vermeiden.
- Die Zusammenfassung mehrerer Attributvolumina zu einem Gesamtvolumen, um Speicherplatz zu sparen und die Anzahl der Zugriffe auf die Daten zu reduzieren.

4.2.1 Zugriff auf Randbereiche des Datenvolumens

Ein unangenehmer Effekt bei Ray-Casting-Algorithmen für Volumendaten entsteht beim Zugriff auf Voxel, die am Rand des Datenvolumens liegen. Wird für solche Voxel eine trilineare Interpolation durchgeführt (z. B. bei der Berechnung des Intensitätswertes für die Schwellwertklassifikation), so hat dies einen Zugriff auf eine nicht gültige Speicheradresse zur Folge, weil die Nachbarn eines solchen Randvoxels außerhalb des Volumens liegen (Abbildung 29). Dieses Problem ist nicht durch das normale Ray-Clipping zu beheben.

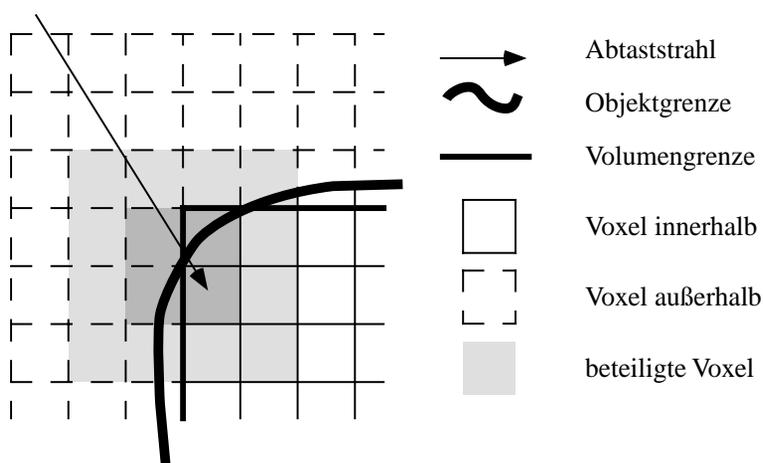


Abbildung 29: Bei der Interpolation der Grauwerte oder Gradienten im Randbereich des Volumens müssen ungültige Speicherzugriffe außerhalb des Volumens verhindert werden.

Wird der Grauwertgradient für ein solches Randvoxel berechnet und werden die Nachbarn selbst wieder durch trilineare Interpolation erzeugt, so ergeben sich Zugriffe, die maximal 2 Voxel von der Volumengrenze entfernt sind (bei einer Gradienten-Weite von 1, für größere Weiten ergeben sich entsprechend noch höhere Werte). Wie bereits gezeigt wurde, kann die genaue Objektgrenze ebenso außerhalb des Attributvoxels liegen. Es muß also berücksichtigt werden, daß Zugriffe mit mindestens drei Voxeln in jede Richtung außerhalb des Volumens korrekt behandelt werden.

Dazu bieten sich folgende Lösungen an:

1. Es wird generell vor jedem Datenzugriff geprüft, ob die 3D-Koordinate innerhalb des Volumens liegt. Ist das der Fall, kann der Wert an dieser Stelle benutzt werden, andernfalls wird die „0“ benutzt. Bei dieser Methode müssen 6 Tests (jeweils 2 für jede Dimension) vor jedem Datenzugriff durchgeführt werden.

Für eine beliebige Strahlposition $P(x,y,z)$ und ein Volumen V der Größe $W \times H \times D$ ergibt sich folgender Pseudo-C-Code:

```

/* Runden der Koordinate auf „Nearest-Neighbour“ */
X = (int)(P.x + 0.5);
Y = (int)(P.y + 0.5);
Z = (int)(P.z + 0.5);

/* Testen, ob die Position innerhalb des Volumens liegt */
if ( (0<=X) and (X<(W-1)) and
      (0<=Y) and (Y<(H-1)) and
      (0<=Z) and (Z<(D-1)) ) {
  Wert=V[(Z*H + Y)*W + X];
}
else {
  /* Die Position liegt außerhalb des Volumens */
  Wert=0;
}

```

Das Rechenzeitverhalten dieses Verfahrens ist ungünstig, weil die meisten Voxel keine Randvoxel und die Tests somit überflüssig sind.

2. Das Datenvolumen wird nicht vollständig abgetastet, d. h., daß der Strahlbegrenzungsalgorithmus nicht das vollständige Datenvolumen benutzt, sondern an jeder der sechs Seiten jeweils drei Schichten abschneidet. In diesem Fall kann auf die Tests ganz verzichtet werden, was sich in einer deutlich höheren Geschwindigkeit bemerkbar macht, mit der Einschränkung, daß nun die Objekte in den Randschichten nicht mehr visualisiert werden können. Für klinische Datensätze ist diese Vorgehensweise nicht vertretbar.
3. Um den Nachteil der vorgenannten Methode zu vermeiden, kann das Volumen im Rechner auch entsprechend vergrößert werden. Dabei wird entsprechend mehr Speicher angefordert und die Daten werden dann so eingelesen, daß die zusätzlichen Randschichten leer bleiben. Auf diese Weise können die eigentlichen Nutzdaten vollständig visualisiert werden, ohne die aufwendigen Tests durchführen zu müssen. Für die lineare Interpolation der Grauwerte und einer Gradientenweite von einem Voxel erfordert dies eine Hülle von drei Voxeln an jeder Seite des Volumens, so daß ein zusätzlicher Speicherbedarf von 3.6% (bei 512^3 Voxel) bis zu 14.7% (bei 128^3 Voxel) entsteht. Dieses Verfahren ist aber nur zum Zeitpunkt des Ladens der Daten anwendbar. Jede dynamische Veränderung des Randbereichs, etwa eine Vergrößerung der Gradientenweite oder ein anderes Interpolationsverfahren mit größerer Nachbarschaft wie z. B. Splines, kann nur durch Reorganisation der Daten im Speicher realisiert werden.

4. Ein Kompromiß aus den ersten beiden Verfahren wird durch eine Erweiterung des Strahlbegrenzungsalgorithmus erreicht, der nicht nur das vollständige Volumen benutzt, sondern gleichzeitig auch die Parameter für die Schnittpunkte zum unkritischen Innenbereich liefert (Abbildung 30). Mittels dieser zusätzlichen Parameter kann auf einfache Weise festgestellt werden, ob sich die Strahlposition im Randbereich befindet. Beim Abtasten des Strahls müssen dann in den Randbereichen die sechs Tests aus Verfahren 1 durchgeführt werden, während im Inneren des Volumens auf die Tests verzichtet werden kann. Diese Methode ist etwas langsamer als das unter 3. vorgestellte Verfahren, benötigt aber keinen zusätzlichen Speicher.

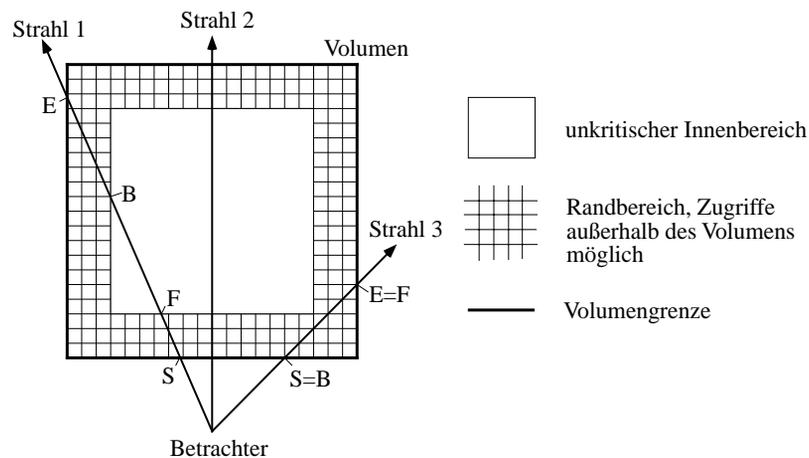


Abbildung 30: Der erweiterte Clipping-Algorithmus liefert neben den Parametern S und E für den Start- bzw. Endpunkt eines Strahls auch die Werte F und B , die die Schnittpunkte mit dem kritischen Randbereich kennzeichnen.

Der Pseudo-C-Code für das Abtasten des Strahls mit dem erweiterten Clipping-Algorithmus sieht wie folgt aus:

```

for (t=S; t<=E; t=t+1) {
  if (t<F or t>B) {
    → Die Position P(t) des Strahls liegt im Randbereich
  }
  else {
    → P(t) ist im sicheren Bereich
  }
}

```

Noch effizienter wird das Programm, wenn der Test, ob die Position $P(t)$ im unkritischen inneren Bereich liegt, durch 3 Teilschleifen ersetzt wird. In diesem Fall kommt die im Mittel längste 2. Teilschleife ganz ohne Tests aus:

```

for (t=S; t<F; t=t+1) {
  → P(t) ist im Randbereich
}
for (; t<=B; t=t+1) {
  → P(t) ist im sicheren Bereich
}
for (; t<E; t=t+1) {
  → P(t) ist im Randbereich
}

```

Die größte Flexibilität wird mit dem zuletzt genannten Verfahren erreicht. Steht jedoch ausreichend Hauptspeicher zur Verfügung, und sind die Maximalwerte der Gradientenweite und die benötigte Nachbarschaft des verwendeten Interpolationsverfahrens bekannt, so ist die unter 3. genannte Speichervergrößerung etwas schneller.

4.2.2 Bounding-Volumes

Ein wesentlicher Nachteil der Volumenrepräsentation von Objekten besteht darin, daß es keinen unmittelbaren Zugriff auf nur diejenigen Voxel gibt, die zu einem bestimmten Objekt gehören. Vielmehr muß für die Projektion eines Objektes jeweils das gesamte Volumen abgetastet werden. Um diesen Nachteil auszugleichen, sind eine ganze Reihe von Verfahren entwickelt worden, die eine möglichst enge, einfach zu berechnende und zu transformierende Hülle um die Objekte erzeugen. Die wichtigsten Verfahren sind:

- „Bounding Boxes“: Hierbei wird für jedes Objekt der minimal umschließende Quader bestimmt. Bei der Projektion des entsprechenden Objektes wird der Umriß dieses Quaders auf der Bildebene bestimmt. Nur Bildpunkte, die innerhalb der Projektion des Quaders liegen, müssen berechnet werden [Blinn 1996]. Für spezielle Objekte, wie etwa diagonal durch das Volumen verlaufende Gefäße, ist die Methode nicht effizient, weil der Quader überwiegend leer ist.
- „Space Subdivision“: Das Volumen wird mit „Octrees“ ([Meagher 1982; Samet, Webber 1988]) oder „Adaptiven Gittern“ ([Klimaszewski, Sederberg 1997]) in Subvolumina unterteilt, so daß leere Bereiche schnell übersprungen werden können. Nachteil bei diesen Verfahren ist der Aufwand für die Berechnung, in welchem Subvolumen sich der Abtaststrahl gerade befindet, insbesondere bei perspektivischen Darstellungen.
- „Minimum Distance“: Zu jedem Voxel wird der kleinste Abstand zu den angrenzenden Objekten in einem zusätzlichen Volumen eingetragen. Auf diese Weise kann man von jeder Position aus mit der jeweils minimalen Schrittweite den Strahl weiter verfolgen, ohne Objekte zu überspringen [Šrámek 1994; Zuiderveld 1995]. Die Nachteile sind, daß ein weiteres Volumen benötigt wird, die Berechnung der minimalen Abstände aufwendig ist und keine Richtungsinformation vorhanden ist. Verläuft der Abtaststrahl parallel zu einer Objektkante, so bildet diese Objektkante den minimalen Abstand, nicht jedoch das Objekt, das in Strahlrichtung möglicherweise weit voraus liegt.

In der vorliegenden Arbeit wurde daher ein Verfahren entwickelt und implementiert, das zu jedem beliebigen Objekt die minimal umschließende Voxelhülle erzeugt. Diese Hülle liegt unmittelbar am Objekt an, so daß keine leeren Zwischenräume entstehen und die Abtaststrahlen somit minimale Länge erhalten. Die Berechnung der Hülle erfolgt nach folgendem Schema:

1. Markieren aller Objektvoxel, die innerhalb der Hülle liegen sollen.
2. Füllen dieser Maske, um Hohlräume innerhalb der Hülle zu vermeiden.
3. Dilation der Maske und Differenzbildung mit der ursprünglichen in Schritt 2 erzeugten Maske.
4. Die verbleibenden Voxel bilden das begrenzende Volumen und werden in Form einer Koordinatenliste gespeichert.

Ein Problem dieses Verfahrens ist, daß bei komplexer Oberflächenstruktur eine sehr große Koordinatenliste entsteht, die entsprechend mehr Zeit für die Berechnung der Start- und Endpunkte der Abtaststrahlen erfordert. Außerdem ist die Projektion der Voxel bei perspektivischen Abbildungen schwierig, weil im Prinzip für jedes Voxel der Hülle die Region auf der Projektionsebene berechnet werden muß, die dieses Voxel überdeckt. Aus Performanzgründen wurde deshalb die Projektion der Hülle mit einem normalen Strahlbegrenzungsalgorithmus für Quader kombiniert. Abbildungen 31 und 32 zeigen einen Quader bzw. eine Voxelhülle als Strahlbegrenzung. Zum leichteren Verständnis sind die Abstände von der Bildebene invertiert dargestellt, d. h., kleine Entfernungen erscheinen hell, und große Abstände sind dunkel dargestellt. Für die schwarzen Bereiche werden überhaupt keine Strahlen berechnet. Bei der Voxelhülle ist deutlich zu erkennen, daß nur für Bildpunkte, die das Objekt berühren, Strahlen erzeugt werden, die zudem wesentlich kürzer sind als bei einem umschließenden Quader.

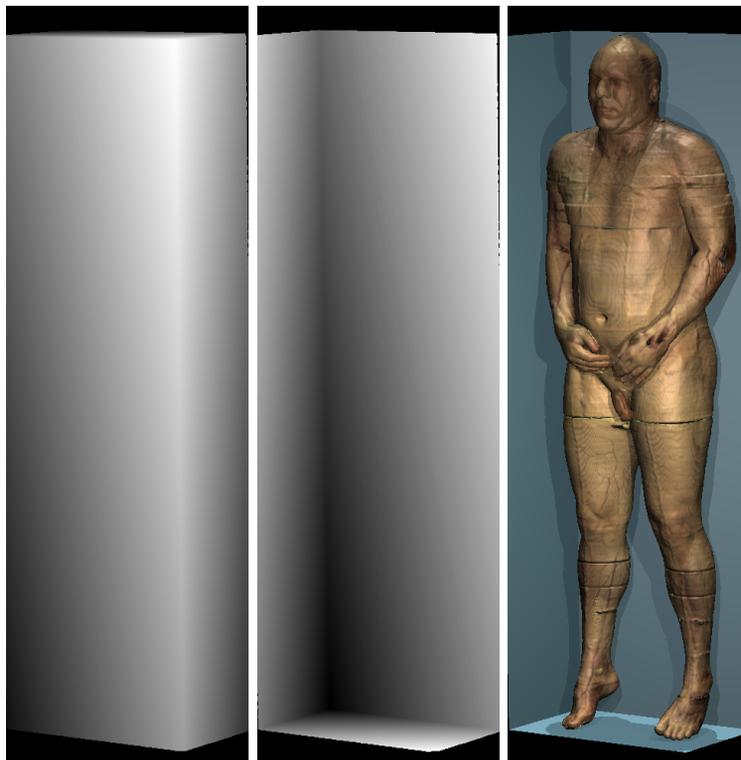


Abbildung 31: *Strahlverfolgung ohne Bounding-Volume: Der Datensatz begrenzt die Strahlen. Man beachte die durch die perspektivische Projektion unterschiedliche Geometrie der vorderen (links) und der hinteren (Mitte) Strahlbegrenzung. Zur leichteren Erkennbarkeit sind kleine Abstände von der Projektionsebene hell und große Abstände dunkel dargestellt.*

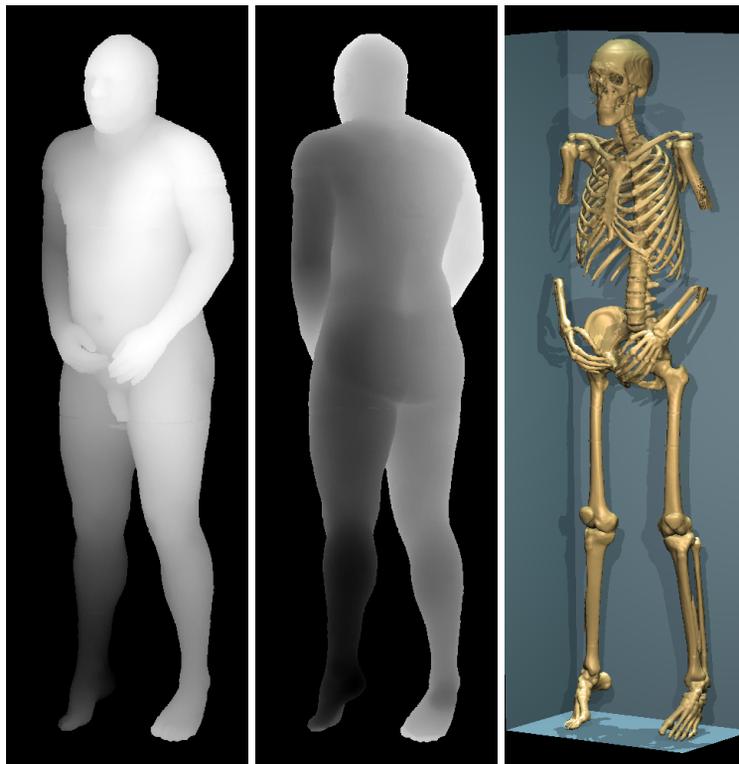


Abbildung 32: *Strahlverfolgung mit Bounding-Volume: Die Strahlen beginnen und enden an der Hautoberfläche. Die fehlenden Armknochen (rechts) sind ein Artefakt in den CT-Daten des Visible-Human, siehe Abschnitt 5.1.4.*

In Tabelle 2 sind beispielhaft die Rechenzeiten für die Abbildungen 31 und 32 zusammengefaßt. Dabei wurde die Zeit für die Berechnung des Schattenwurfs nicht berücksichtigt. Als Bounding-Volume wurde die Hautoberfläche benutzt. Die kürzeren Rechenzeiten für die Skelettdarstellung gegenüber der Hautoberfläche sind durch die fehlende Texturierung der Knochenoberfläche bedingt.

Abb.	Bildgröße	BV	Clipping [s]	Ray-Casting [s]	Faktor
31	325×900	ohne	1.4	81	1
		mit	2.3	40	0.51
32	325×900	ohne	1.4	76	1
		mit	2.3	31	0.43

Tabelle 2: *Rechenzeiten für die 3D-Darstellungen in Abbildung 31 und 32, ermittelt auf einer DEC-station 250 4/266 (Alpha 21064 CPU, 266 MHz Prozessortakt, 512 MB Hauptspeicher). Die Datensatzgröße beträgt 194×113×620 Voxel (65 MB). Das Bounding-Volume besteht aus 334593 Voxeln. Die Berechnungszeit für die Darstellung wird durch Verwendung von Bounding-Volumes in etwa halbiert.*

In der Anwendung hat es sich als praktisch erwiesen, für das äußere Objekt, meistens die Haut, eine Voxelhülle zu definieren und für die innen liegenden Strukturen Quader zu verwenden. Dadurch wird der Hintergrund optimal übersprungen und für die Objekte im Inneren eine besonders schnelle Methode der Strahlbegrenzung angewendet.

4.2.3 Generierung eines multiattributierten Labelvolumens

Ein wesentlicher Gesichtspunkt bei der Entwicklung des in dieser Arbeit beschriebenen Systems war es, jedem Voxel verschiedenartige Attribute zuordnen zu können. Aus Vereinfachungsgründen, insbesondere bei der Segmentation, wird dabei jedes Attribut in einem eigenen Datenvolumen gehalten. Bei der Visualisierung muß dann für jede Position entlang der Sehstrahlen auf jedes dieser Volumina zugegriffen werden, um den jeweiligen Attributwert zu ermitteln. Mehrere Zugriffe verlangsamten aber die Darstellungsgeschwindigkeit. Außerdem ist nur selten der volle Wertebereich für ein Attribut ausgenutzt, so daß ungenutzter Speicherplatz entsteht. Bei dem anatomischen Hirnatlas aus Abschnitt 5.2.4 wurden beispielsweise 136 morphologische Strukturen segmentiert, dazu 28 funktionale Areale und 21 Blutversorgungsgebiete. Für jedes dieser Attribute wird ein Byte benötigt, so daß bei einer Volumengröße von $150 \times 200 \times 192$ Voxel ein Speicherbedarf von 16.5 MB entsteht. Die mengentheoretische Verknüpfung der verschiedenen Attribute, etwa „zeige alle Hirnwindungen, die an einer bestimmten Funktion beteiligt sind **und** von einem bestimmten Blutgefäß versorgt werden“, führt zu theoretisch $136 \times 28 \times 21 = 79968$ verschiedenen Teilmengen. Bei voller Ausnutzung der 8 Bit pro Attribut entstehen sogar $256^3 = 16777216$ Teilmengen. Für jede Region eines „geschnittenen“ Volumens müßten also jeweils fast 17 Millionen Darstellungsparameter vorgehalten werden. Es ist sofort einsehbar, daß man hier schnell jede Speichergrenze sprengen würde. Es wurde deshalb ein Verfahren entwickelt, das die verschiedenen Attributvolumina, nachdem die Segmentation abgeschlossen ist, zu einem neuen Gesamtvolumen zusammenfaßt. Dabei muß es möglich sein, die ursprünglichen Attributwerte der Einzelvolumina aus dem neuen Volumen abzuleiten, damit der Zugriff auf die Originalobjekte erhalten bleibt.

Das hier entwickelte und in Abbildung 33 dargestellte Verfahren arbeitet wie folgt: Zunächst wird eine Tabelle der Länge „Anzahl der Objekte im 1. Attributvolumen \times Anzahl der Objekte im 2. Volumen $\times \dots \times$ Anzahl der Objekte im n. Volumen“ angelegt. Diese Tabelle nimmt die neuen Attributwerte auf. Im zweiten Schritt wird ein neues Volumen erzeugt, in dem alle Schnittmengen der Einzelvolumina eindeutig gekennzeichnet sind. Dazu werden die Volumina sequentiell durchlaufen und für jede Voxelposition wird aus den einzelnen Attributwerten eine eindeutige Teilmengenummer T nach dem Horner-Schema errechnet: Seien A_1 bis A_n die Attributwerte und O_1 bis O_n die Anzahl der Objekte der Volumina 1 bis n, so ist

$$T = (\dots((A_n O_{n-1} + A_{n-1}) O_{n-2} + A_{n-2}) \dots) O_1 + A_1$$

Existiert in der Tabelle an der Stelle T bereits ein Eintrag, so wird dieser in das neue Attributvolumen an der augenblicklichen Position eingesetzt. Im anderen Fall ist die Teilmenge T bisher nicht aufgetreten. Der Zähler für die bisher gefundenen Teilmengen wird erhöht und an der Position T in die Tabelle eingetragen. Gleichzeitig wird in einer „inversen“ Tabelle an der Stelle des neuen Teilmengezählers die Teilmengenummer T eingetragen. Durch diese Vorwärts- und Rückwärts-Indizierung ist es später möglich, auf die ursprünglichen Attributwerte der Einzelvolumina zurückzuschließen. Im dritten Schritt wird das neue Volumen mit den dazugehörigen Tabellen gesichert. Nachfolgend werden dann die ursprünglichen Attributvolumina nicht mehr benötigt.

Für den oben erwähnten Hirnatlas wurden 813 verschiedene Teilmengen gefunden, die in einem 16-Bit-Volumen untergebracht werden können, so daß neben der Reduktion des Speicherbedarfs von 5,5 MByte auch noch eine Zeitersparnis bei der Attributermittlung erreicht wird.

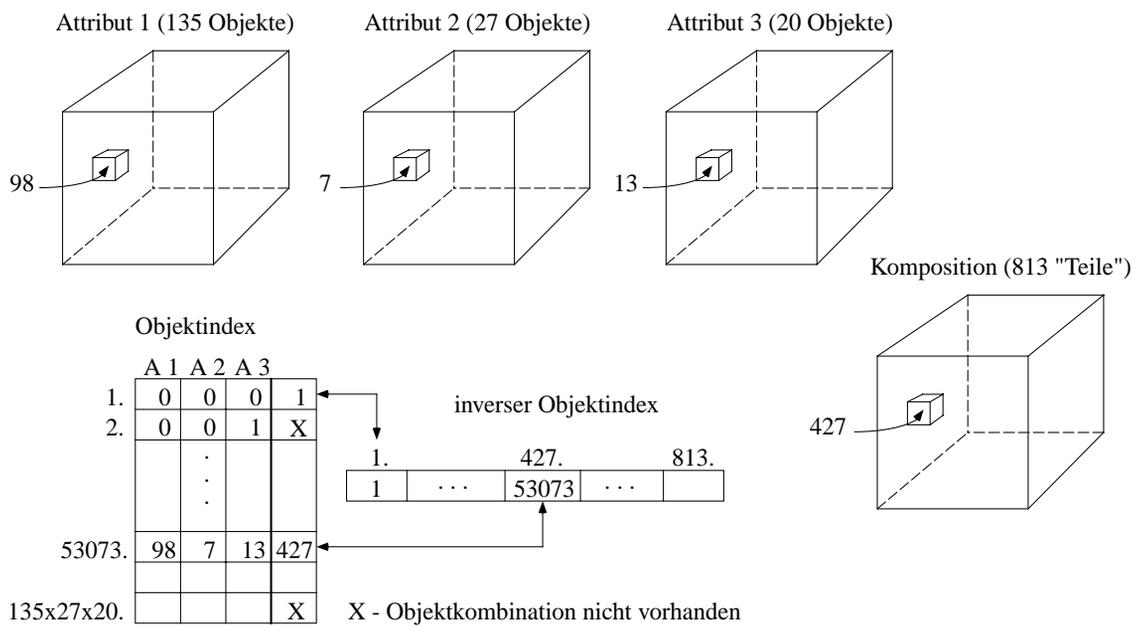


Abbildung 33: Zusammenfassung der einzelnen Attributvolumina zu einem Gesamtvolumen. Die Umrechnung der Attributwerte in das neue Kompositionsvolumen und umgekehrt erfolgt über entsprechende Tabellen.

Anstelle von drei Datenzugriffen auf die Einzelvolumina ist nur noch ein Zugriff auf das kombinierte Volumen erforderlich.

5 Anwendung und Ergebnisse

5.1 Vergleich der Qualität der Oberflächendarstellung

5.1.1 Demonstration an einem künstlichen Objekt

Für einen qualitativen Vergleich des herkömmlichen Strahlverfolgungsverfahrens und der neu entwickelten Methode der klassifikationsgesteuerten Bestimmung der Attributwerte im Subvoxelbereich wurde ein künstliches Volumen von 32^3 Voxeln erzeugt. Es enthält als einziges Objekt eine 1 Voxel dünne Spirale. Die Objektvoxel sind „26-connected“, d. h., sie sind über gemeinsame Flächen, Kanten und Eckpunkte miteinander verbunden. Dieses Objekt dient als Modell für sehr kleine und dünne Strukturen, wie etwa die Blutgefäße. Um eine manuelle Segmentation zu simulieren, wurde der Partial-Volumen-Effekt ausgeschlossen, indem eine binäre Repräsentation gewählt wurde. Es sind also nur zwei Grauwerte, einer für den Hintergrund und einer für das Objekt, vorhanden. Abbildung 34 (oben links) zeigt das Ergebnis eines „brute-force“ Verfahrens, bei dem das Volumen mit konstanter Schrittweite abgetastet und der Gradient jeweils an der Abtastposition ohne Interpolation berechnet wird. Es sind zahlreiche Artefakte sichtbar, die zum einen durch das Verfehlen der Oberfläche durch die konstante Schrittweite entstehen und zum anderen durch eine falsche Gradientenberechnung zu Beleuchtungsfehlern führen. Demgegenüber bringt ein klassischer 3D-Algorithmus, der das Volumen an den diskreten Voxelpositionen abtastet, die voxelige Struktur der Spirale richtig hervor (Abbildung 34, oben rechts). Diese Darstellung ist im mathematischen Sinne korrekt, weil keinerlei weitere Annahmen über die zugrundeliegenden Daten gemacht werden. Allerdings entspricht diese Darstellung nicht der Erwartungshaltung für natürliche Objekte, insbesondere stört die gerasterte Oberfläche. In Abbildung 34 (unten links) wurde die Objektoberfläche klassifikationsgesteuert im Subvoxelbereich bestimmt. Da kein Partial-Volumen-Effekt vorhanden ist, so daß eine Klassifikation auf Grauwertbasis nicht berechnet werden kann, wurde die im Abschnitt 3.5 beschriebene trilineare Interpolation der Attributwerte benutzt. Es ist eine deutlich glattere und besser zusammenhängende Oberfläche zu erkennen. Die Abbildung 34 (unten rechts) wurde mittels B-Spline-Interpolation der Attributwerte und einem Schwellwert von 0.25 (statt 0.5) berechnet, wodurch sich das Objekt geringfügig ausdehnt und die Verbindung über die Kanten der Voxel glatter erscheint. Diese Darstellung kommt der Erwartungshaltung an natürliche Objekte sehr nahe. Das Beispiel verdeutlicht auch, daß eine glatte Darstellung binärer Objekte ohne aufwendige Hilfsmittel, wie etwa der Gauß-Filterung der Attributvolumina, möglich ist. Dieses künstliche Abrunden der Oberfläche ist nur unter der Annahme zulässig, daß natürliche Objekte wie die menschlichen Organe keine Voxelstruktur besitzen, sondern sich eher weich als scharfkantig voneinander abgrenzen.

5.1.2 Auswirkung bei starken Vergrößerungen

Besonders deutlich zeigen sich die Unterschiede zwischen herkömmlichem Ray-Casting und dem neu entwickelten Verfahren bei vergrößerten Darstellungen, bei denen ein Bildpunkt im Vergleich zu einem Voxel sehr klein wird, so daß die Projektionsfläche eines Voxels viele Bildpunkte umfaßt. Die Abbildung 35 zeigt eine vergrößerte Darstellung der Rekonstruktion eines Auges aus dem Visible-Human-Datensatz. Auf diesen speziellen Datensatz wird in Abschnitt 5.1.4 noch näher eingegangen.

Abbildung 36 zeigt den Qualitätsunterschied der 3D-Darstellung an einem Ausschnitt einer realen Blutgefäßstruktur, die mit Schwellwerten aus einem MR-Angiographie-Datensatz [Urban *et al.* 1997] segmentiert wurde.

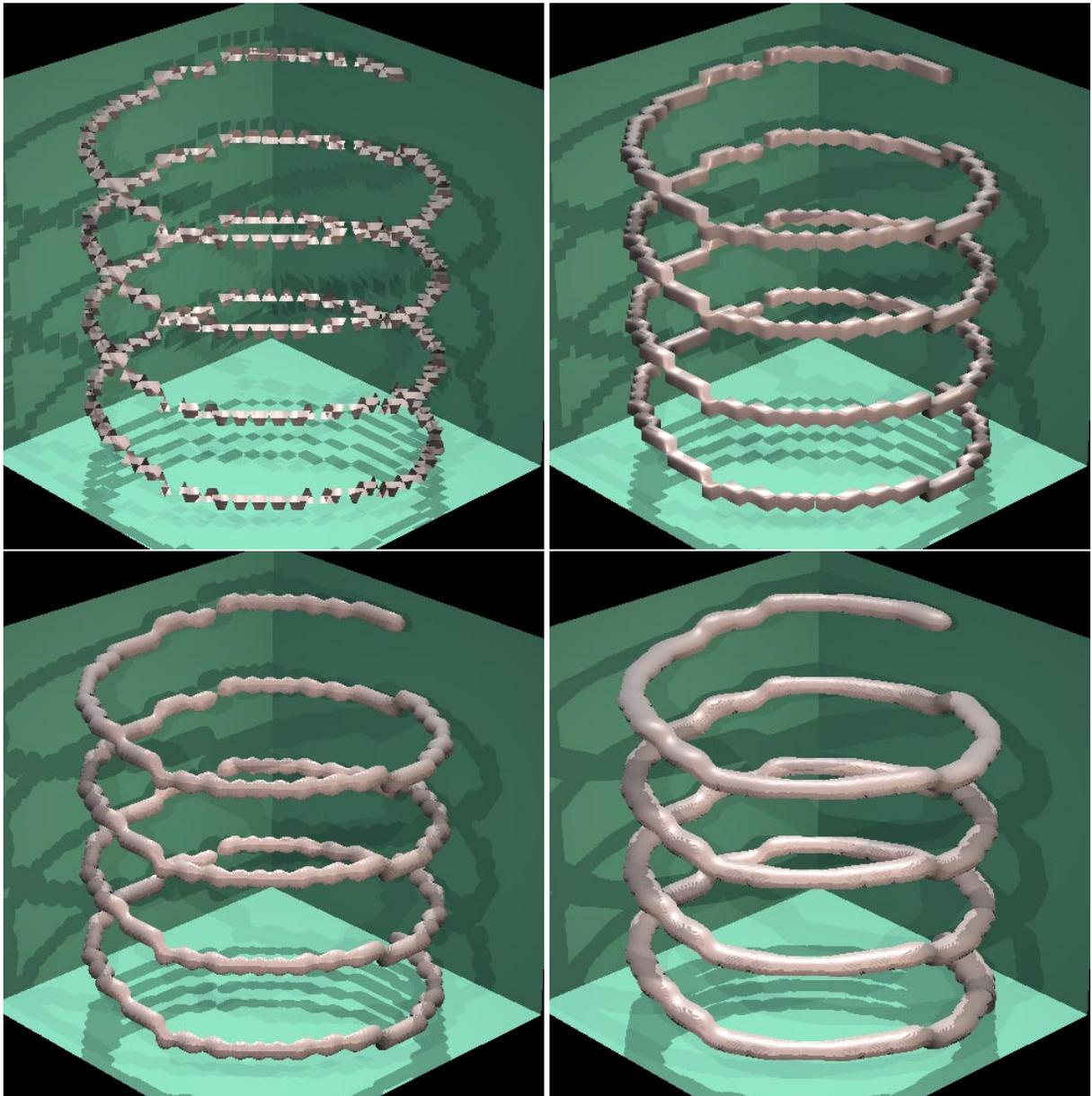


Abbildung 34: Darstellung einer 1 Voxel dünnen Spirale: „Brute-force“ Algorithmus (oben links), korrekte Strahlverfolgung (oben rechts), klassifikationsgesteuerte Oberflächenbestimmung im Subvoxelbereich mit trilinearere Interpolation der Attributwerte und einem Schwellwert von 0.5 (unten links) und mit B-Spline-Interpolation und einem Schwellwert von 0.25 (unten rechts).

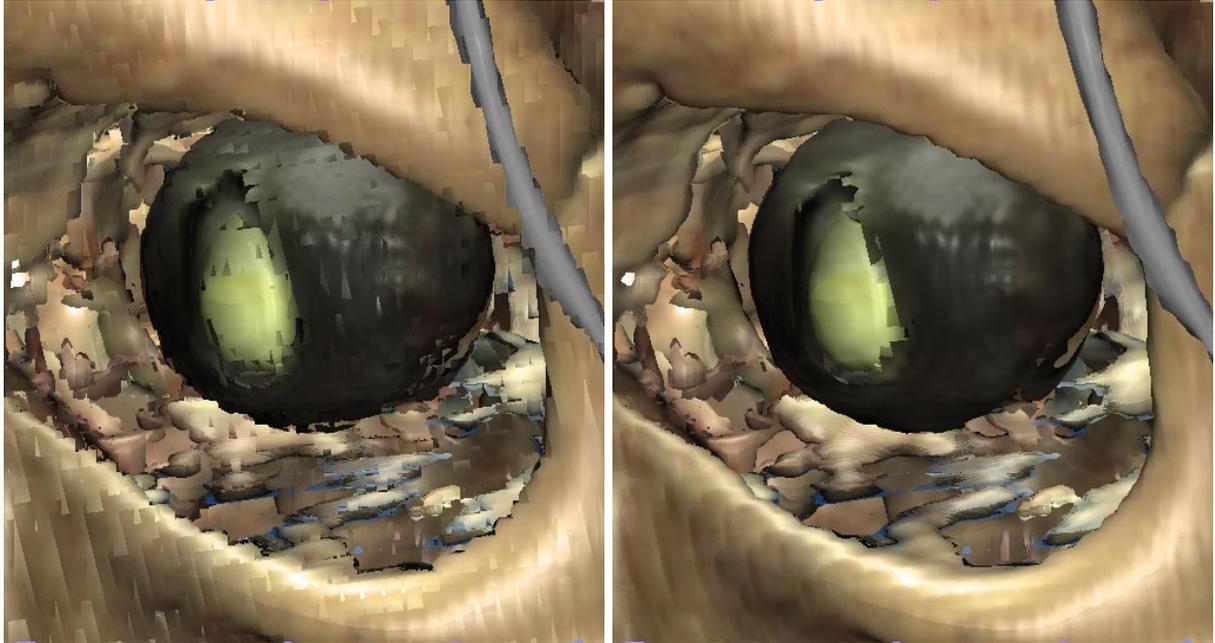


Abbildung 35: Vergleich der Qualität der Darstellung eines Auges bei Berechnung der Attributwerte unter Berücksichtigung des Schwellwerts an der Abtastposition (links) und durch klassifikationsgesteuertes Suchen des besten Wertes (rechts). Die Oberflächen und Objektränder sind wesentlich glatter.

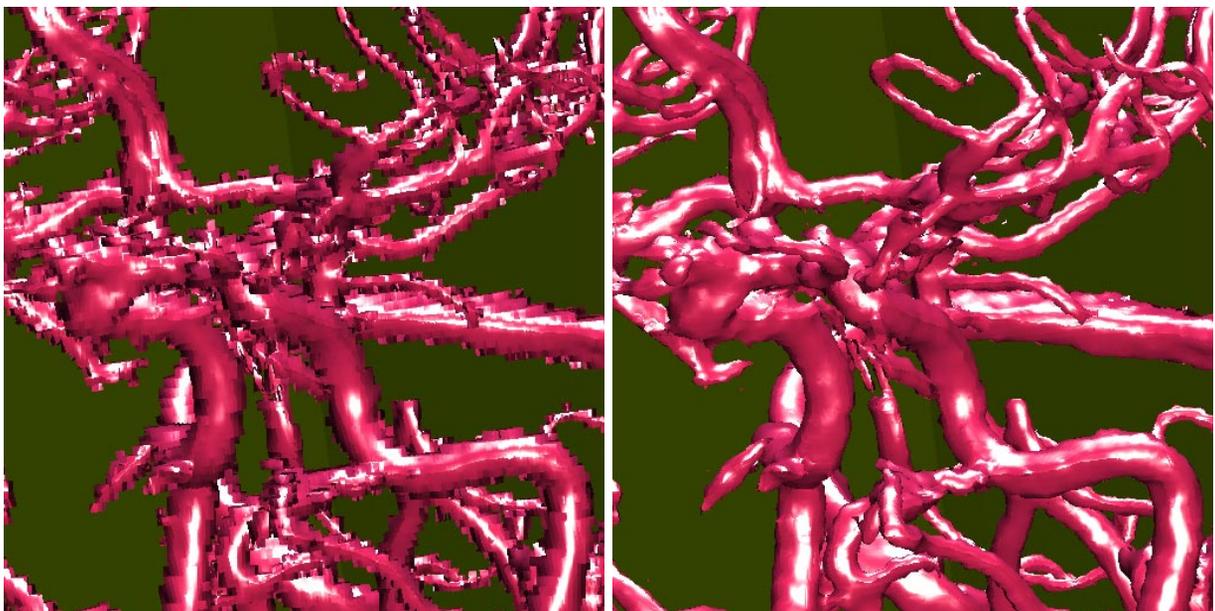


Abbildung 36: Ausschnitt aus einer Blutgefäßstruktur: Bestimmung der Attributwerte wie in Abbildung 35. Insbesondere die Erkennbarkeit kleiner Gefäße ist deutlich verbessert.

Für eine realistische 3D-Darstellung ist neben der reinen Oberflächendarstellung auch die Berücksichtigung des Schattenwurfs notwendig, der durch die Verdeckung der Lichtquellen durch undurchsichtige Objekte zustande kommt. Bei einem einfachen Schlagschattenmodell wird dazu ausgehend von der gerade ermittelten Objektoberfläche ein Strahl in Richtung der Lichtquelle erzeugt. Dieser Strahl wird nach dem gleichen Prinzip abgetastet wie die Bestimmung der sichtbaren Oberfläche mit dem Unterschied, daß der Strahl abbricht, sobald ein undurchsichtiges Objekt getroffen wird. Das bedeutet, daß die Attributwerte entlang des Schattenstrahls ebenfalls mit dem aufwendigen Suchverfahren berechnet werden müssen. Abbildung 37 zeigt die deutlich verbesserte Qualität bei der Berechnung von Schlagschatten mit dem neuen Verfahren.

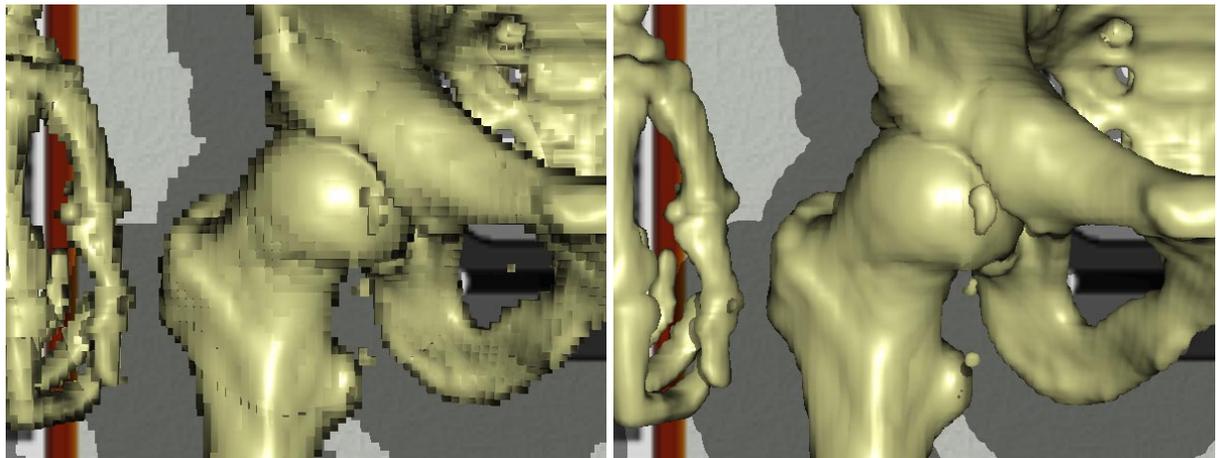


Abbildung 37: *Rekonstruktion eines Hüftgelenks aus einem CT-Datensatz einer Leiche: Man beachte den Qualitätsgewinn auch des Schattenwurfs auf der Wand im Hintergrund durch die klassifikationsgesteuerte Attributbestimmung.*

5.1.3 Qualitätsvergleich der berechneten Attributwerte

Für den folgenden Qualitätsvergleich wurde eine segmentierte MR-Sequenz verwendet, bei der die Objekte überwiegend durch eine Grauwertkante getrennt sind (siehe auch Abschnitt 5.2.4). Für die Darstellung der Attributwerte wurden diese bei der Projektion durch das Strahlverfolgungsverfahren auf drei Arten bestimmt:

1. Mittels „nearest-neighbour“ Zugriff, wie er bei der herkömmlichen Strahlverfolgung verwendet wird,
2. unter zusätzlicher Berücksichtigung des Schwellwerts an der Abtastposition,
3. durch das neu entwickelte klassifikationsgesteuerte Suchverfahren.

In Abbildung 38 (obere Reihe) sind die farblich markierten Attributwerte verschiedener Objekte auf einem transversalen Schnitt durch den Kopf in Höhe der Augen gezeigt. Bei einer Abbildungsgröße, bei der die Kantenlänge eines Bildpunktes in etwa der Kantenlänge eines Voxels entspricht, sind zwischen den 3 Arten der Attributbestimmung kaum Unterschiede sichtbar. Beim Heranzoomen an einen Ausschnitt wird die voxelige Struktur der Attribute jedoch zunehmend dominanter, wodurch die Qualität der Darstellung entsprechend leidet (Abbildung 38,

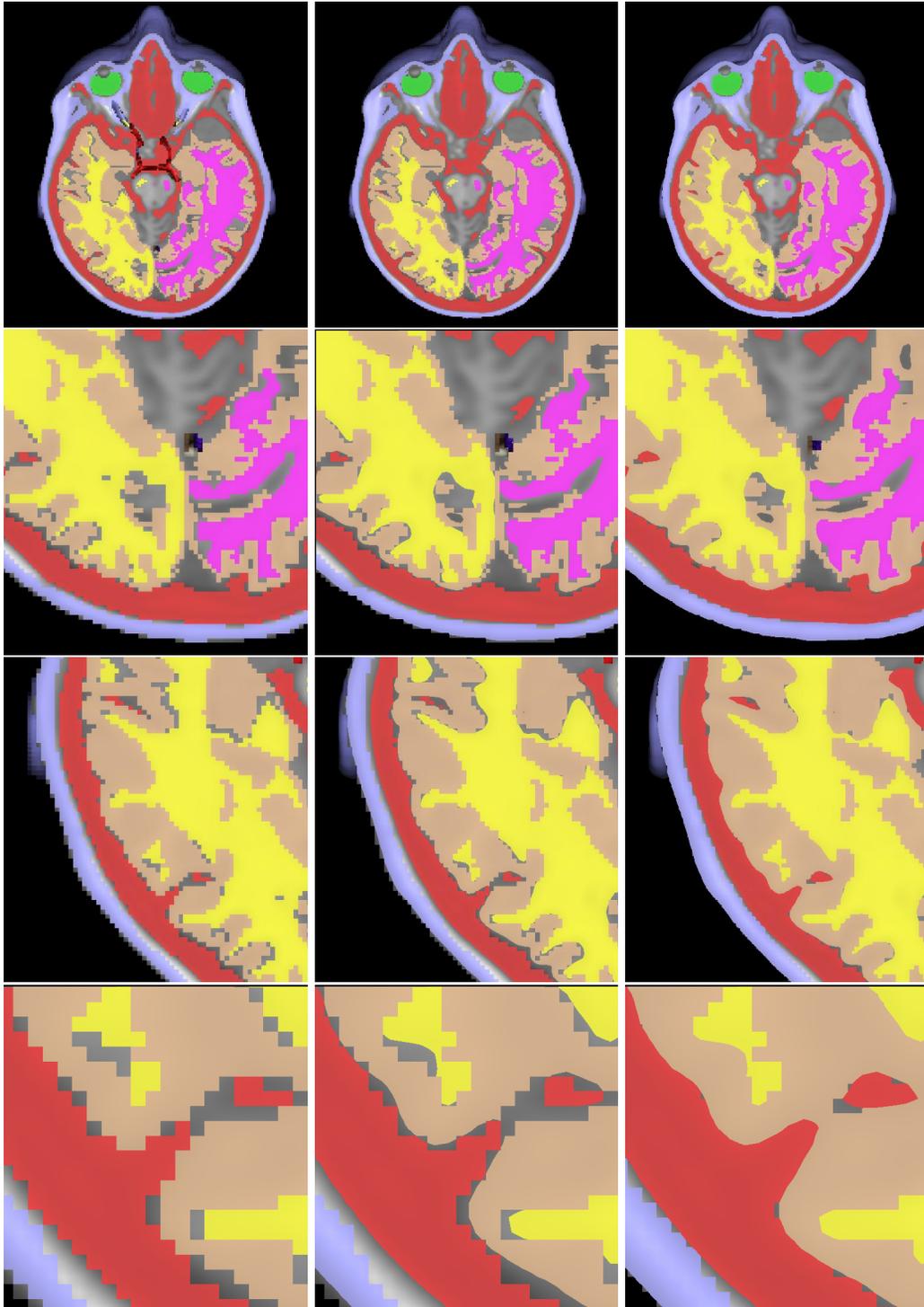


Abbildung 38: *Bestimmung der Attributwerte: Mit nearest-neighbour Zugriff (linke Spalte), zusätzlich mit dem Schwellwert an der Abtastposition (mittlere Spalte) und klassifikationsgesteuert im Subvoxelbereich (rechte Spalte). Je stärker die berechnete Vergrößerung ist, desto deutlicher wird der Qualitätsgewinn durch die klassifikationsgesteuerte Attributberechnung, sichtbar an den glatten Objektübergängen.*

linke Spalte). Es ist auch zu sehen, daß die zusätzliche Berücksichtigung von Schwellwerten allein nicht ausreicht (mittlere Spalte). Erst die Verwendung des im Abschnitt 3.2 beschriebenen klassifikationsgesteuerten Suchverfahrens erzeugt die im Bezug auf die verwendete Schwellwertsegmentation korrekte Zuordnung der Abtastposition zu einem Objekt im Subvoxelbereich und damit bei jeder Vergrößerung einen glatten Verlauf der Oberfläche zwischen benachbarten Objekten (rechte Spalte).

5.1.4 Visible-Human-Projekt

1988 startete die National Library of Medicine das Visible-Human-Projekt mit dem Ziel, einen vollständigen Satz digital photographierter Farbschnittbilder der menschlichen Anatomie zu erzeugen [National Library of Medicine 1996]. Eine entsprechende Ausschreibung wurde von der medizinischen Fakultät der Universität von Colorado gewonnen. Es wurden 1994 ein gefrorener männlicher und 1996 ein weiblicher Leichnam zerschnitten [Spitzer *et al.* 1996]. Zusätzlich wurden CT- und MR-Aufnahmen von den frischen Leichen angefertigt und beim Mann außerdem CT-Aufnahmen des gefrorenen Leichnams:

- 1871 anatomische Schnittbilder mit einer Auflösung von 0.33 mm, 2048×1216 Pixel, 24-Bit Farbe, 1 mm Abstand (Abbildung 39)
- 1871 CT-Schichtbilder der gefrorenen Leiche mit einer Auflösung von ca. 0,53 mm, 512×512 Pixel, 12-Bit Grauwerte, 1 mm Abstand
- verschiedene CT- und MR-Sequenzen der frischen Leiche mit 256×256 Pixel, 12-Bit Grauwerte, 4 mm Abstand

Die weibliche Leiche wurde in 1/3 mm dünne Schichten geschnitten. Das Datenmaterial umfaßt beim Mann ungefähr 15 GB und bei der Frau etwa 40 GB.

Die Datensätze bieten gegenüber radiologischen Daten einige Vorteile. Zum einen haben sie eine wesentlich verbesserte räumliche Auflösung und zum anderen stehen nicht nur Grauwerte, sondern auch Farbinformationen der menschlichen Anatomie volumenmäßig zur Verfügung. Eine direkte 3D-Visualisierung dieser großen Datenmengen ist mit heutigen Workstations nicht möglich. Die folgenden Resultate beschränken sich daher auf den Kopf des Mannes, der zusätzlich in der Schnittelebene um den Faktor 3 verkleinert wurde, so daß die Gesamtvolumengröße ungefähr 97 MB beträgt. Die Registrierung der gefrorenen CT-Daten wurde interaktiv auf eine Genauigkeit von etwa 1 Voxel in jede Dimension verbessert, während die MR- und CT-Schichten der frischen Leiche nur recht grob in Übereinstimmung gebracht werden konnten, da hier nichtlineare Verzerrungen auftraten. Für die Segmentierung waren diese Daten deshalb nicht verwendbar. Die Segmentierung der Knochenstrukturen erfolgte durch Schwellwerte aus den CT-Daten, während verschiedene Weichteile des Kopfes durch Ellipsoide (Tabelle 3) im RGB-Farbvolumen charakterisiert wurden [Schiemann *et al.* 1997]. Für manche Strukturen ergaben sich erhebliche Überschneidungen in der Klassifikation, so daß diese Objekte mit dieser Methode nicht trennbar waren. Die 3D-Rekonstruktion der Objektoberfläche wurde wie im Abschnitt 3.6 beschrieben unter Berücksichtigung der Ellipsoide auf den RGB-Werten berechnet. Der Gradient wurde als Mittelwert der Gradienten der drei Einzelkomponenten bestimmt [Tiede *et al.* 1996]. Abbildung 40 zeigt eine Rekonstruktion der Hautoberfläche mit den Originalfarbwerten auf den Schnittflächen.

Für eine realistische Darstellung der Objektoberflächen wurden die Farbwerte der anatomischen Daten an der Oberfläche durch Interpolation berechnet und in der Phong-Schattierung

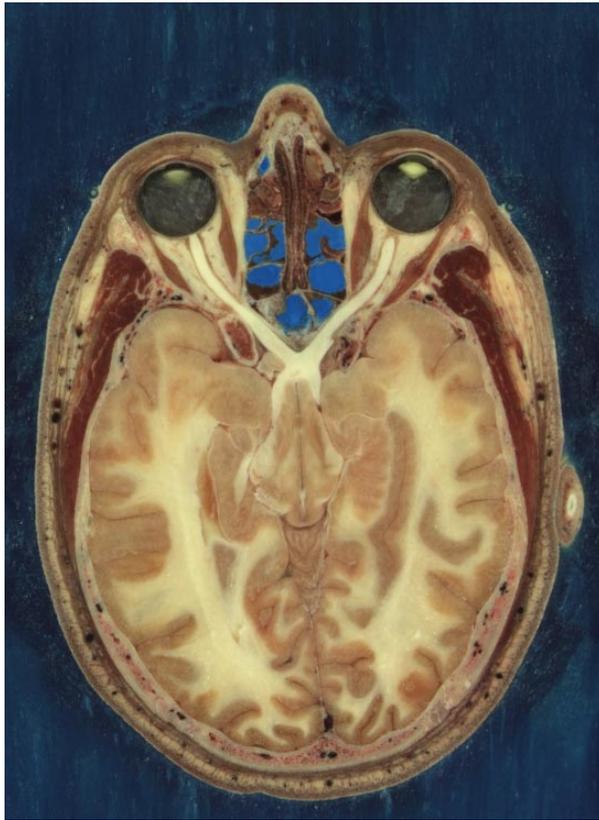


Abbildung 39: *Digital photographierter Kryotomschnitt des Visible-Human.*



Abbildung 40: *Rekonstruktion des Visible-Human: Die Schnittflächen zeigen die Originalfarben der anatomischen Schnitte.*

<i>Objekt</i>		<i>Zentrum</i>	<i>Länge</i>	<i>Basis</i>		
Muskeln	R	67.34	86.35	0.88	0.38	0.29
	G	39.47	21.11	-0.48	0.70	0.53
	B	31.36	8.02	0.00	-0.60	0.80
Augen	R	74.16	83.72	0.53	0.63	0.56
	G	76.36	16.25	0.84	-0.30	-0.46
	B	65.23	7.04	0.12	-0.71	0.69
Augenmuskeln	R	139.32	47.63	0.84	0.52	0.18
	G	82.44	17.81	-0.53	0.70	0.49
	B	54.40	7.15	0.13	-0.50	0.85
Linse	R	152.02	101.62	0.58	0.64	0.50
	G	160.79	17.65	-0.39	-0.31	0.87
	B	97.67	8.98	0.71	-0.70	0.07
Graue Hirnsubstanz	R	182,75	54.18	0.44	0.68	0.58
	G	154.72	10.20	0.76	0.06	-0.65
	B	103.33	2.73	0.48	-0.73	0.49
Weiße Hirnsubstanz	R	214.00	84.89	0.46	0.64	0.62
	G	202.24	13.77	0.63	0.26	-0.73
	B	141.41	6.81	0.63	-0.72	0.28
Sehnerv	R	219.38	33.49	0.48	0.68	0.56
	G	216.89	12.24	0.68	0.11	-0.72
	B	165.11	5.52	0.55	-0.73	0.41
Weichteile	R	146.36	174.55	0.62	0.65	0.44
	G	122.75	46.00	0.74	-0.30	-0.60
	B	90.54	24.14	0.26	-0.70	0.67
Nebenhöhlen	R	63.43	79.96	0.82	0.34	-0.46
	G	91.34	38.63	0.18	0.60	0.78
	B	121.63	11.95	-0.54	0.72	-0.43

Tabelle 3: *Klassifikation verschiedener Objekte im Kopf des Visible-Human durch Ellipsoide im RGB-Merkmalraum. Eine bildliche Darstellung findet sich in Abbildung 16.*

als Grundfarbe verwendet. Dabei hat sich gezeigt, daß besonders bei der Hautoberfläche eine Mischung mit dem blauen Hintergrund die Farbe stark verfälscht. Eine wesentlich bessere Färbung wurde mit der in Abschnitt 3.9 beschriebenen Verschiebung des Berechnungsortes für die Interpolation um ein bis zwei Voxel unter die Objektoberfläche erzielt. Besonders augenscheinlich ist dieser Effekt bei den Lippen, die bei einer Berechnung der Farbe direkt an der Objektoberfläche nicht von der umgebenden Haut unterschieden werden können. Eine Interpolation der Farbe zwei Voxel unterhalb der Oberfläche erzeugt eine klare Differenzierung (Abbildung 41). Es entstehen nuancenreiche, plastische und sehr realistische Abbildungen der menschlichen Anatomie (Abbildung 42), wie sie zuvor mit computergestützten Verfahren nicht möglich waren.

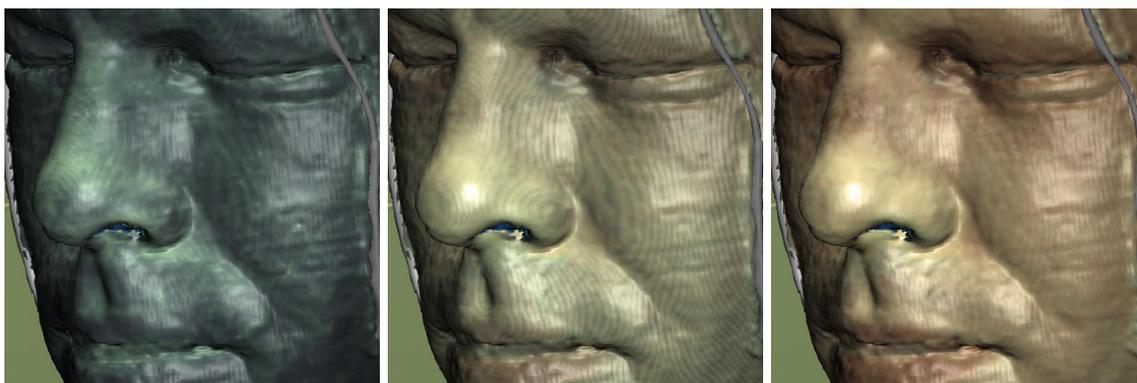


Abbildung 41: *Variation des Interpolationspunkts für die Berechnung der Objektfarbe: Direkt an der Objektoberfläche (links), 1 bzw. 2 Voxel entlang der Oberflächennormalen unter die Oberfläche (Mitte und rechts). Man beachte die Farbe der Lippen.*

Mit dem Konzept der individuellen Objektparametrisierung ist es möglich, Oberfläche und Gradienten für den Schädel aus den CT-Daten zu berechnen, während die Farbe für den Knochen aus den RGB-Daten ermittelt wird. Abbildung 43 zeigt die kombinierte Darstellung von Knochen und Muskelgewebe. Die bräunlich-rot gefärbten Regionen im Bereich der Schädeldecke deuten auf kleine Registrierungsfehler hin, bei denen die Knochen das umgebende Muskelgewebe geringfügig durchdringen. Der Ausschnitt der Abbildung zeigt die Detailtreue, die mit den reduzierten Daten erreichbar ist. Die flexible Parametrisierung ermöglicht auch die gleichzeitige Darstellung verschiedener Aspekte aus den verfügbaren unterschiedlichen Bildmodalitäten. So können z. B. auf den Schnittflächen die anatomischen Farbwerte oder die entsprechenden CT- und MR-Werte dargestellt werden. Die Objekte lassen sich oberflächenschattiert darstellen oder einzelne Regionen können als simuliertes Röntgenbild oder mit multiplen transparenten Oberflächen abgebildet werden (Abbildung 44). Für Lehr- und Ausbildungszwecke ist dieser vergleichende Aspekt von besonderer Bedeutung.

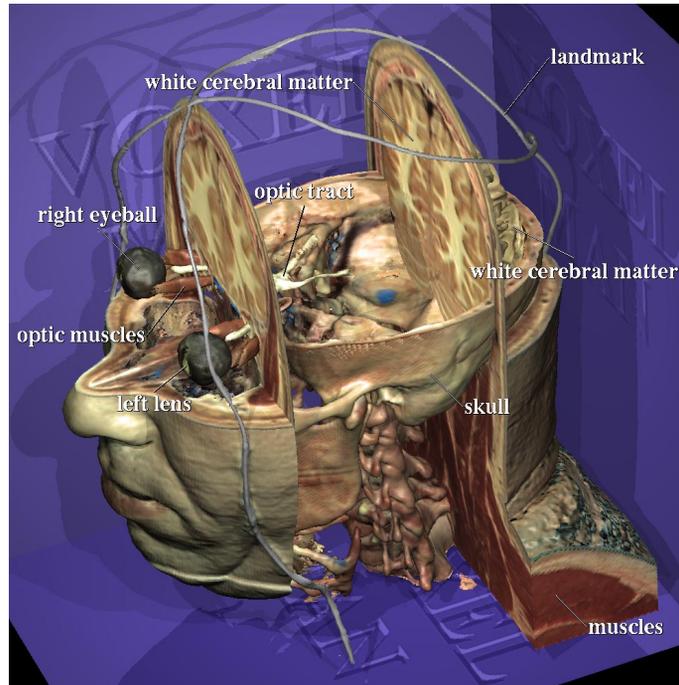


Abbildung 42: Komplexe Komposition aus verschiedenen Oberflächen, Schnittflächen und Spanten. Die knöchernen Strukturen wurden aus den CT-Daten segmentiert, alle anderen Organe aus den Kryotomschnitten. Die Objektflächen und Schnittebenen wurden mit den Originalfarbwerten texturiert.

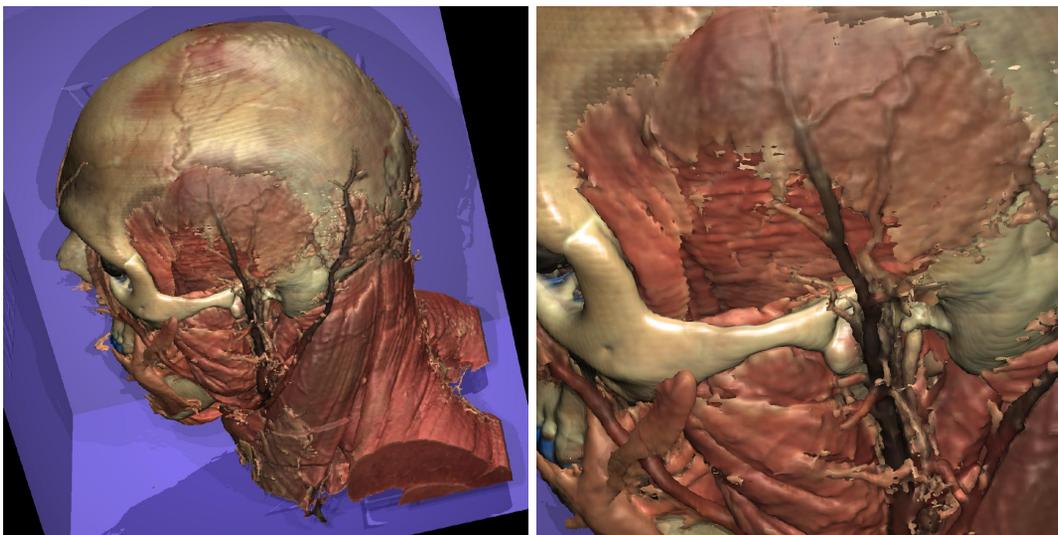


Abbildung 43: Rekonstruktion des Schädels aus den CT-Daten, Muskeln und Gefäße wurden aus den anatomischen Schnitten rekonstruiert.

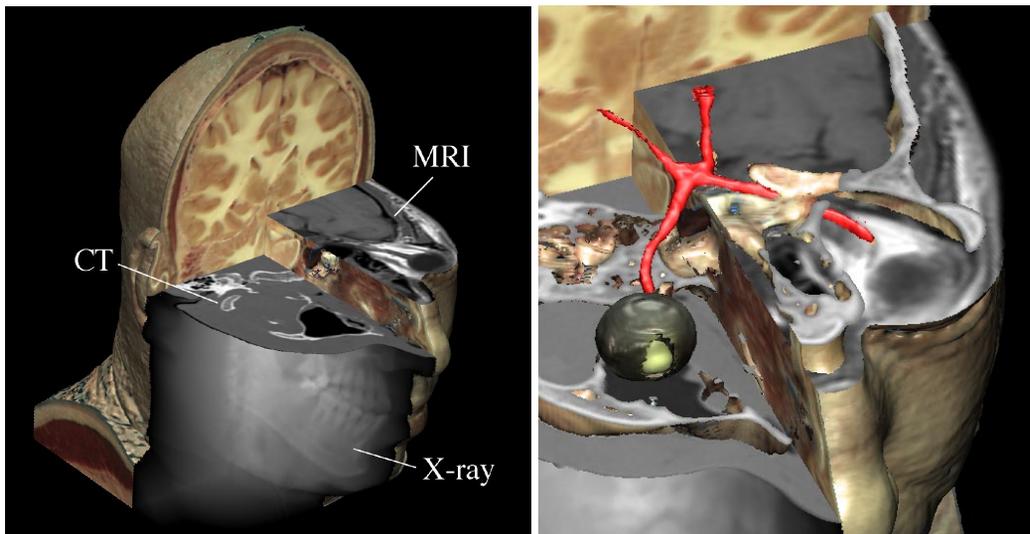


Abbildung 44: Gleichzeitige Darstellung verschiedener Modalitäten: Simuliertes Röntgenbild aus den CT-Daten. Die Schnitte zeigen CT-, MR- und die anatomischen Farbwerte.

5.2 Eigenschaften der Visualisierungsverfahren in verschiedenen Anwendungen

5.2.1 Gefäßdarstellung

In der konventionellen CT- und MR-Angiographie werden die Blutgefäßstrukturen häufig als Maximum-Intensitäts-Projektionen (MIP) dargestellt, weil sich der Verlauf der oft nur 1 Voxel dünnen Gefäße aus Punkten und kurzen Liniensegmenten in den Schnittbildern mental praktisch nicht rekonstruieren läßt. Aus dieser Darstellung kann die räumliche Beziehung der Gefäße zueinander aber nur schwer abgeleitet werden (vgl. Abbildung 6). Die Integration einzelner Schnittebenen ermöglicht die Berechnung von Verdeckungen, so daß die Überlagerung der Strukturen reduziert wird und die Orientierung in der MIP erleichtert wird (Abbildung 45). Eine Oberflächendarstellung unter Berücksichtigung von Schlagschatten der schwelwertsegmentierten Gefäße wurde in Abbildung 45 (rechts) berechnet. Die Lage der Strukturen wird so auch für den weniger Geübten leicht erkennbar. Das Problem besteht hier in der Wahl des richtigen Schwellwerts. Ein zu hoher Schwellwert erzeugt Lücken zwischen den Gefäßen und künstliche Gefäßverengungen (Stenosen), während durch zu niedrige Schwellwerte das Rauschen erhöht wird und nicht differenzierbare Gefäßklumpen entstehen.

Unter Verwendung des in Abschnitt 3.1 beschriebenen punktförmigen Kameramodells wurden auch endoskopische Darstellungen der Gefäße berechnet, die mit parallel projizierenden Verfahren nicht möglich wären. Da die Gefäße nur wenige Voxel Durchmesser haben, erzeugen Standardverfahren keine befriedigenden Resultate. Erst die Berechnung der Objektgrenze im Subvoxelbereich ermöglicht auch bei sehr kleinen Gefäßen glatte Innenansichten (Abbildung 46).

5.2.2 Simulation dynamischer Vorgänge

Neben der Visualisierung statischer Zustände nimmt die Bedeutung der Simulation dynamischer Vorgänge zu. Dynamische Prozesse lassen sich prinzipiell in zwei Kategorien einteilen:

1. Vorgänge, bei denen sich die Geometrie der beteiligten Objekte nicht verändert. Hierzu

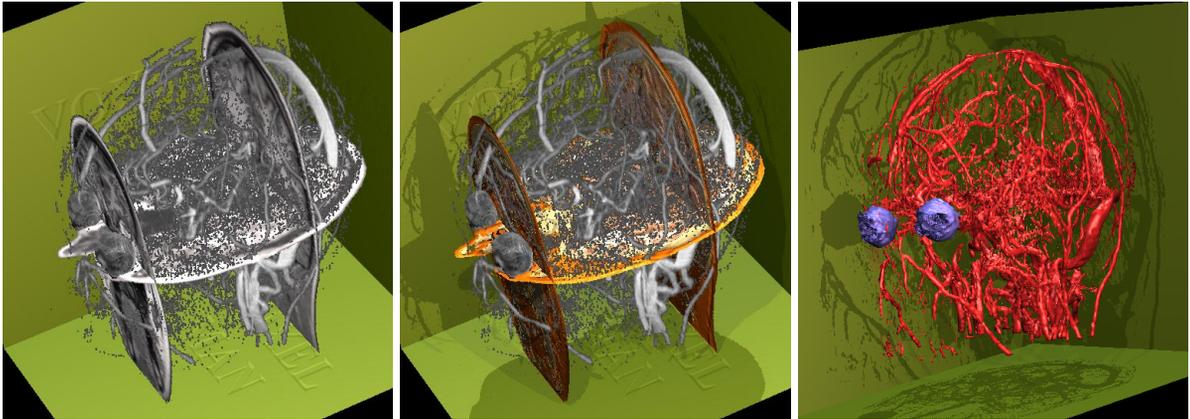


Abbildung 45: Die Orientierung in der Maximum-Intensitäts-Projektion der Blutgefäße aus einer MR-Angiographie wird durch das Einfügen von Schnittebenen erleichtert (links). Darstellung der Schnittebenen im anatomischen Modus mit Schlagschatten (Mitte). Schwellwertbasierte Segmentation und Darstellung der Blutgefäße mit schattierten Oberflächen (rechts).

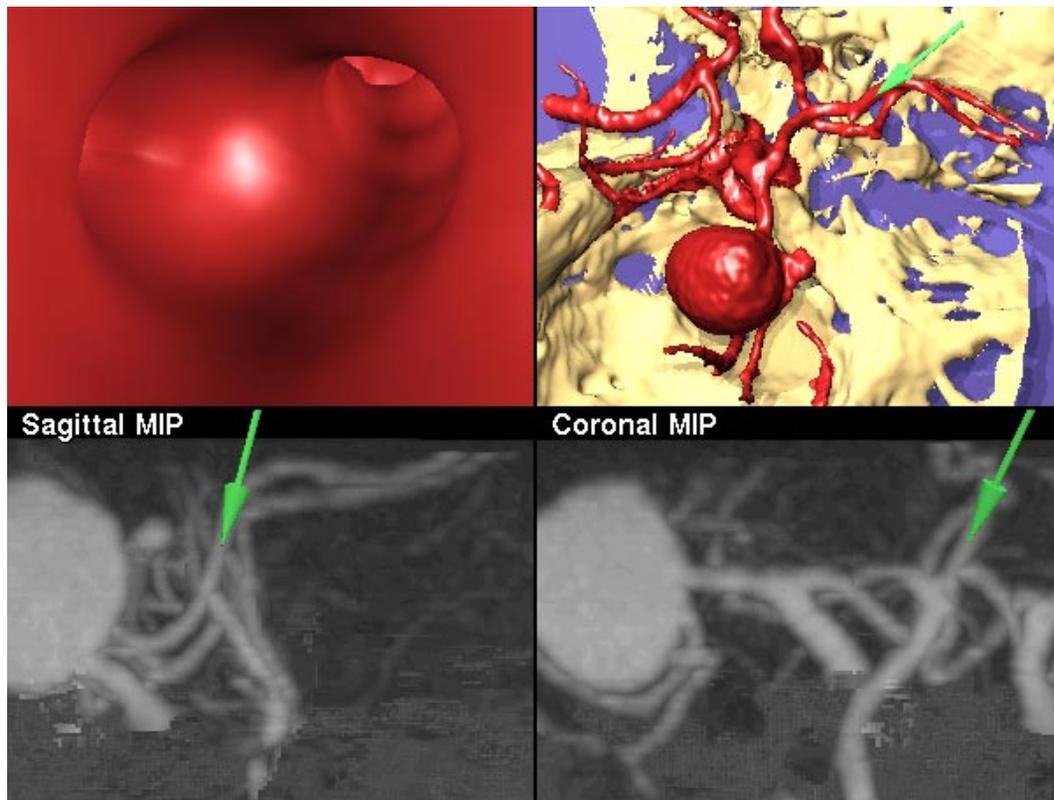


Abbildung 46: Endoskopische Gefäßdarstellung: Der Betrachterstandort befindet sich in einem Blutgefäß (links oben). Die übrigen Darstellungen dienen der Orientierung über Standort und Blickrichtung (grüne Pfeile).

gehören z. B. die Fortpflanzung von Nervenimpulsen oder in erster Näherung auch der Blutfluß in den Gefäßen. (Eine stärkere Verformung der Gefäße tritt nur bei den großen Arterien auf.)

2. Vorgänge, bei denen starke (nichtlineare) Verformungen der Objekte auftreten, wie z. B. bei Muskelkontraktionen, Gelenkbewegungen oder der Atmung.

Während die Berechnung nichtlinearer Verformungen und die Anwendung auf Volumendaten enorm rechenzeitaufwendig sind [Schiemann, Höhne 1997], ist die Visualisierung des Blutflusses mit dem Konzept der multiattributierten Volumina einfach zu realisieren. Als Beispiel wurde für den CT-Angiographie-Datensatz aus Abbildung 46 ein Attributvolumen für die Ankunftszeit, und damit für die Geschwindigkeit, eines Kontrastmittels in den Blutgefäßen folgendermaßen erzeugt:

1. Initialisierung des Volumens auf „0“
2. Definition von drei Startpunkten mit dem Zeitwert $t_0 = 1$. Als Startpunkte wurden die Eintrittspunkte der Basilararterie sowie der linken und rechten inneren Kopfschlagader in das Datenvolumen gewählt.
3. In einem rekursiven Prozeß erhielten die zu den Arterien gehörenden und noch mit keinem Zeitwert versehenen Nachbarvoxel der Voxel mit dem Zeitwert t_n den Zeitwert $t_{n+1} = t_n + 1$.

Auf diese Weise erhält jedes Voxel, das zu einer Arterie gehört, zusätzlich einen Wert, der der kürzesten Entfernung von einem der drei Startpunkte entspricht. In erster Näherung ist die Entfernung vom Startpunkt proportional zur Ankunftszeit des Blutes an einem bestimmten Ort. Die Simulation der Ausbreitung des Kontrastmittels erfolgt durch Einfärben und Entfärben aller Voxel, deren Zeitwert innerhalb eines jeweils fortlaufenden Zeitintervalls t_m bis t_n liegt.

Die Berechnung der Bildsequenz in Abbildung 47 wurde mit dem nachfolgenden Unix-Shellskript realisiert, das entsprechende Kommandos der im Anhang C beschriebenen Sprache für die Steuerung der Visualisierung erzeugt:

```
#!/bin/csh -f

@ t0 = 1          # Startzeitpunkt
@ t1 = $t0 + 15  # Laenge des Bolus
@ dt = 5         # Geschwindigkeit des Bolus

while ($t1 <= 255)
  @ t = $t0

  echo "SET AREA * ID=(BOLUS=($t - $t0)) COLOR=NONE"      # Bolus loeschen
  echo "SET AREA * ID=(BOLUS=($t0 - $t1)) COLOR=PAINT_5" # Bolus markieren
  echo "UPDATE CAMERA CURRENT"                            # Kame-
ra aktualisieren

  @ t0 = $t0 + $dt
  @ t1 = $t1 + $dt
end
```

Dieses einfache Modell gibt nicht den exakten zeitlichen Verlauf eines Kontrastmittels wieder, dazu müßten die Viskosität des Blutes, der Gefäßquerschnitt, der Blutdruck und andere hämodynamische Eigenschaften in das Modell eingehen. Es zeigt aber die potentiellen Möglichkeiten

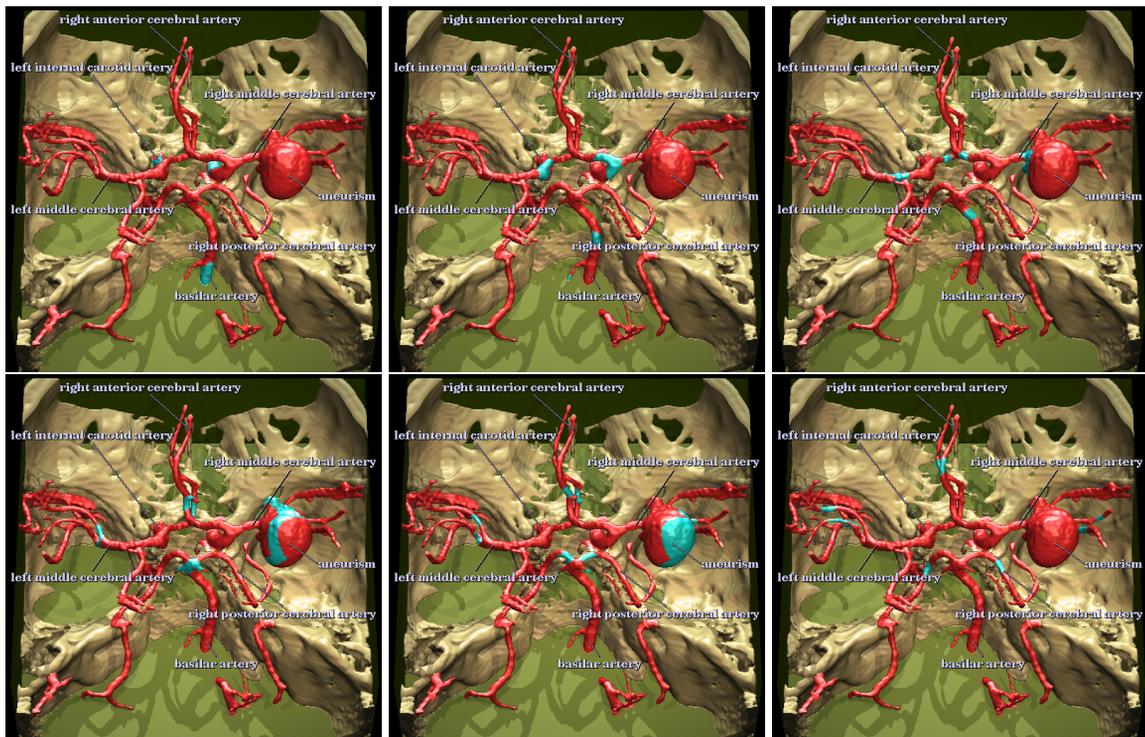


Abbildung 47: Ausschnitt aus einer Bildsequenz zur Simulation des Blutflusses im Bereich der Schädelbasis: Ein Kontrastmittelbolus (türkis) fließt durch das aus einem CT-Angiographie-Datensatz rekonstruierte Gefäßsystem und markiert die Ankunftszeit des Blutes an einem bestimmten Ort (von links oben nach rechts unten).

der hier entwickelten Verfahren für die Volumenvisualisierung dynamischer Vorgänge, wenn entsprechende Attributvolumina zur Verfügung stehen. Auf ähnliche Weise ließe sich z. B. die Fortpflanzung eines Reizes durch die Nervenbahnen simulieren.

5.2.3 Multiparametrische Daten

In zunehmendem Maße werden je nach Fragestellung mehrere Bildsequenzen von demselben Patienten erzeugt, um verschiedene morphologische oder funktionale Aspekte besser sichtbar und beurteilbar machen zu können. Dies können Akquisitionen derselben Modalität wie z. B. mehrere MR-Aufnahmen mit unterschiedlichen Parametereinstellungen (T1- oder T2-gewichtet) sein oder Aufnahmen mit verschiedenen Modalitäten wie CT und MR oder PET und MR. Ziel der Visualisierung solcher Daten ist es, die multiparametrische Information gleichzeitig nutzbar zu machen. Zur Zeit ist eine räumliche Übereinstimmung der Daten nur bei MR-Akquisitionen gegeben, die mit demselben Gerät in unmittelbarem zeitlichen Zusammenhang erfolgen, also ohne daß der Patient zwischen den Aufnahmen seine Lage verändert. Bei zeitlich und räumlich getrennt durchgeführten Aufnahmen oder solchen mit verschiedenen Modalitäten müssen zunächst die Daten nachträglich registriert werden, d. h., die Volumina müssen zur Deckung gebracht werden, so daß jedes Voxel des einen Datensatzes derselben räumlichen Lage im anderen Volumen entspricht. Dabei muß zwischen linearen und nichtlinearen Verzerrungen unterschieden werden. Eine verbreitete Technik zur Kompensation linearer Verzerrungen besteht in der Verwendung externer Landmarken oder der Definition gemeinsamer morphologischer Strukturen in den verschiedenen Volumina, um daraus die entsprechende affine Transformation abzuleiten. Für nichtlineare Verzerrungen, wie sie z. B. durch Wachstumspro-

zesse oder Gelenkbewegungen entstehen, existiert eine generelle Lösung, die eine voxelgenaue Übereinstimmung erzeugt, bisher nicht. Auf die spezielle Problematik der Registrierung multiparametrischer Datenvolumina soll hier nicht weiter eingegangen werden.

5.2.3.1 Visualisierung multiparametrischer MR-Volumina

Abbildung 48 zeigt einen klinischen Fall, bei dem drei MR-Volumina erzeugt wurden, von denen je einer den arteriellen bzw. den venösen Blutfluß zeigt und der dritte die übrigen Weichteilstrukturen. Die Segmentation der Venen (blau) und Arterien (rot) sowie der anderen Organe erfolgte mit Schwellwerten auf dem jeweiligen Volumen. Für die gemeinsame Visualisierung ist es erforderlich, alle Volumina während des Ray-Castings im Speicher zu halten, weil die Schwellwerte und Gradienten zur Laufzeit objektabhängig jeweils aus dem entsprechenden Volumen berechnet werden müssen.

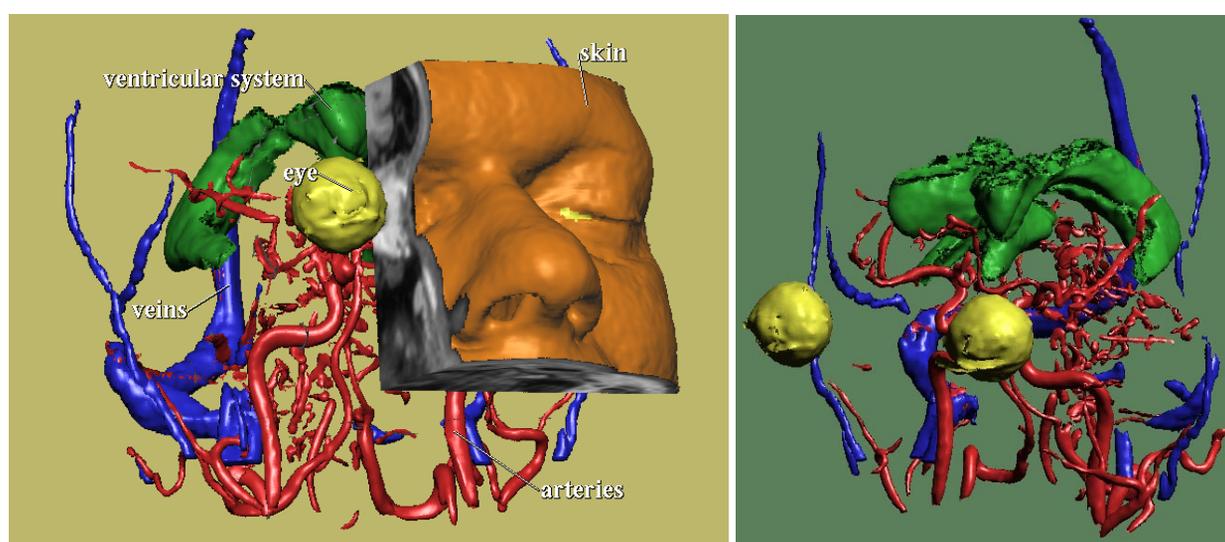


Abbildung 48: *Darstellung eines aus drei verschiedenen MR-Akquisitionen erzeugten Datensatzes: Die Segmentation und die Berechnung der Oberfläche und des Gradienten für Venen (blau), Arterien (rot) und die übrigen Strukturen erfolgte jeweils aus eigenen Volumina.*

5.2.3.2 Kombinierte 3D-Darstellung von PET und MR

Die Positronen-Emissions-Tomographie (PET) macht Stoffwechselfvorgänge im Körper durch Messung der Anreicherung bestimmter radioaktiver Isotope sichtbar. Da PET-Daten nur eine geringe räumliche Auflösung haben und praktisch keine morphologischen Strukturen zeigen, werden in der Regel zusätzlich MR-Schichtbilder erzeugt. Abbildung 49 zeigt die kombinierte Visualisierung aus MR und PET, wobei die Registrierung mit Hilfe externer Landmarken, die in beiden Modalitäten ein Signal erzeugen, erfolgte [Pietrzyk 1994]. Aus den MR-Daten wurden Haut, Augen, Gehirn und Ventrikelsystem segmentiert. Die Darstellung der Hirnaktivität aus den PET-Daten, die den Glukosestoffwechsel zeigen, erfolgte mit der in Abschnitt 3.9 beschriebenen Technik der Volumentexturierung. Hierbei wurde für jeden aus den MR-Daten berechneten Punkt der Hirnoberfläche der korrespondierende Intensitätswert im PET-Volumen bestimmt. Dieser Wert diente als Index in eine für die PET-Visualisierung gebräuchliche Farbtabelle, die den unterschiedlich starken Glukoseverbrauch in Farbwerte umsetzt. Die entsprechenden Farbwerte wurden dann als Grundfarbe für die Berechnung der Schattierung mit dem Phongsehen

Beleuchtungsmodell verwendet. Durch dieses Verfahren werden die morphologischen Strukturen des Gehirns sichtbar, und gleichzeitig ist die Aktivität an jedem Volumenpunkt ablesbar (Abbildung 49).

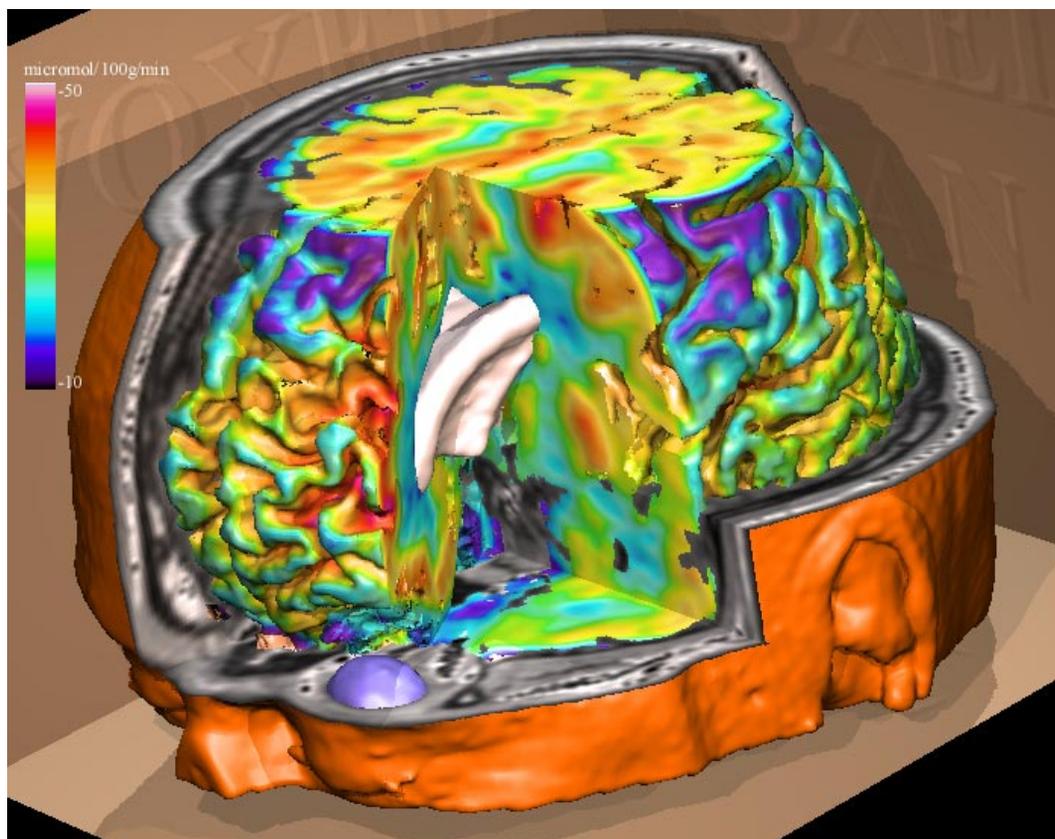


Abbildung 49: *Kombinierte Visualisierung von PET und MR: Das aus MR-Daten segmentierte Hirnvolumen wurde mit den korrespondierenden PET-Werten, die den Glukosestoffwechsel zeigen, texturiert. Um die Aktivität zu verdeutlichen, wurden die PET-Werte mit Hilfe einer Standardfarbtabelle eingefärbt.*

5.2.4 Entwicklung intelligenter dreidimensionaler anatomischer Atlanten

Eine wesentliche Einschränkung für die Anwendbarkeit der 3D-Visualisierung in der klinischen Routine liegt nach wie vor in der ungelösten automatischen Segmentierung der Volumendaten. In der Regel steht nur eine begrenzte Zeit für die Datenaufbereitung zur Verfügung, die eine detaillierte Segmentierung nicht zulässt. Anders ist die Situation für die Lehre und Ausbildung in der Anatomie. Hier genügt es, bezüglich einer Domäne ein einzelnes möglichst vollständiges Modell wie etwa vom Kopf oder Bauch zu erzeugen. Vor diesem Hintergrund werden am Institut für Mathematik und Datenverarbeitung in der Medizin des Universitäts-Krankenhauses Eppendorf seit 1990 Forschungsarbeiten für die Entwicklung virtueller dreidimensionaler Atlanten auf der Basis tomographischer Volumendaten durchgeführt [Höhne *et al.* 1992; Tiede *et al.* 1993]. Die Grundidee dabei ist die Erzeugung eines Modells, in dem räumliches Wissen und symbolische Beschreibung des menschlichen Körpers verknüpft sind. Die räumliche Repräsentation erfolgt durch bestmögliche Segmentierung der Volumendaten hinsichtlich verschiedener Domänen wie z. B. struktureller oder funktionaler Anatomie, um eine möglichst realistische 3D-Visualisierung zu erreichen. Diese Segmentierung kann von einem Experten mit dem für einen

bestimmten Aspekt notwendigen Fachwissen durchgeführt werden und darf auch längere Zeit in Anspruch nehmen, weil das einmal erstellte Modell immer wieder verwendet werden kann und nicht wie bei den klinischen Anwendungen die spezifischen Besonderheiten des individuellen Patienten berücksichtigen muß. Die segmentierten Objekte erhalten Eigenschaften für die Visualisierung wie Farbe, Textur und Lichtreflexion, oder biomechanische Eigenschaften wie Verformbarkeit. Die gegenseitigen Wechselbeziehungen werden in Form eines semantischen Netzwerkes modelliert, das die Verknüpfung von Objekten mit Hilfe von Relationen mit gegebener Semantik ermöglicht [Schubert *et al.* 1992; Pommert *et al.* 1994; Schubert *et al.* 1995; Höhne *et al.* 1995b]. Die aus der Verbindung von symbolischer Beschreibung mit einem konkreten Datensatz eines Patienten oder Probanden resultierende Struktur wird als „intelligentes Volumen“ bezeichnet [Höhne *et al.* 1996a] (Abbildung 50).

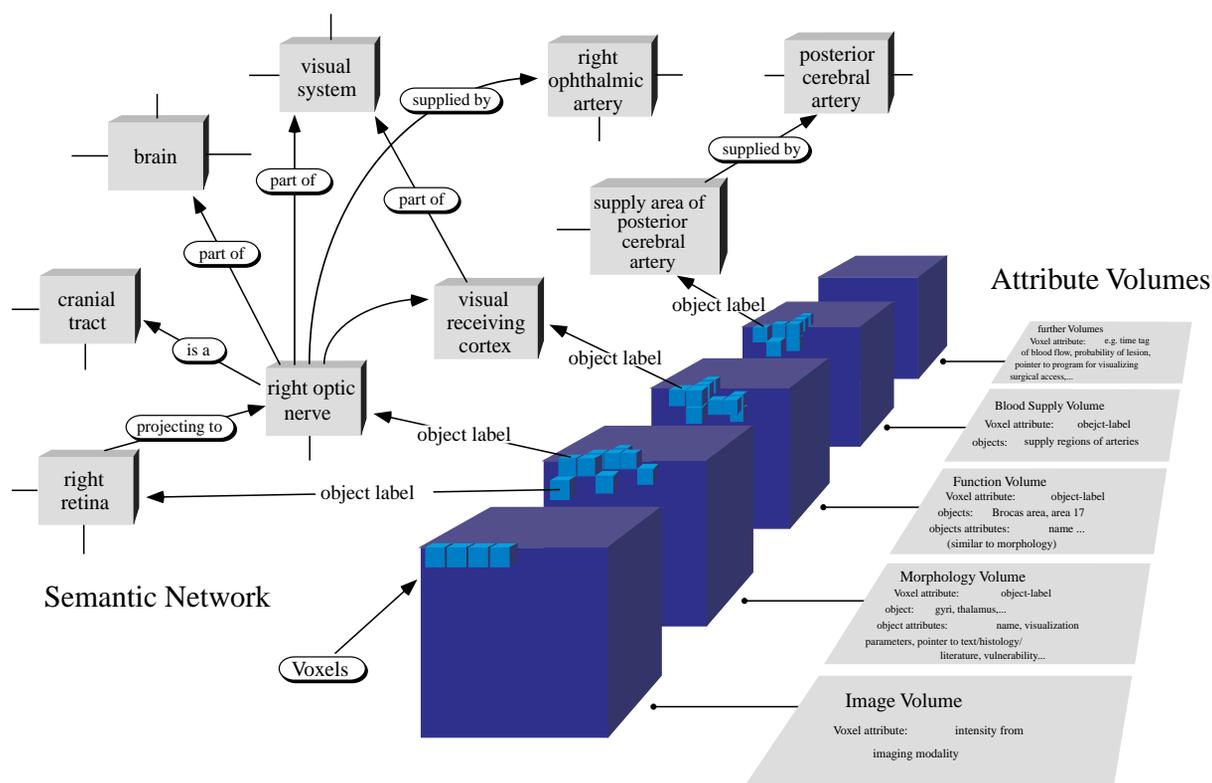


Abbildung 50: Verknüpfung von räumlichem Wissen und semantischem Netzwerk zu einem „intelligenten“ Volumen

Es wurden verschiedene Atlanten erzeugt. Besonders aufwendig wurden ein Hirnatlas aus einem MR-Datensatz (150×200×192 Voxel), ein Atlas des knöchernen Schädels aus einem CT-Volumen (240×222×246 Voxel) [Höhne 1995] und ein Atlas eines Fötus aus MR-Daten (256×256×108 Voxel) [Krämer 1997] segmentiert. Für den Schädel wurden 84 Knochensegmente definiert und der Hirnatlas bezüglich der Domänen Morphologie, funktionale Anatomie und Blutversorgung in 156, 26 beziehungsweise 20 Objekte segmentiert, die jeweils in einem eigenen Attributvolumen gespeichert wurden. In Tabelle 4 sind die interaktiv bestimmten Intensitätsbereiche der Hauptbestandteile des Hirnatlasses aufgeführt. Die Feinsegmentation, z. B. die Trennung der verschiedenen Hirnwindungen, die sich im Grauwertprofil nicht unterscheiden, wurde manuell durchgeführt.

Aus den Attributvolumina der drei Domänen ergaben sich insgesamt 813 unterschiedliche Voxelmengen. Das semantische Netzwerk besteht aus etwa 1000 Objekten mit 2500 Verknüpfungen in den verschiedenen Domänen. Durch entsprechende Verknüpfungen innerhalb

Objekt	Schwellwert	
	unterer	oberer
Augen	0	60
Blutgefäße	31	255
graue Hirnsubstanz	61	255
weiße Hirnsubstanz	91	255
Hornhaut	20	59
Knochen	0	60
Ventrikelsystem	0	90
Weichteile	45	255

Tabelle 4: Schwellwerte für verschiedene Objekte eines MR-Datensatzes. Aus Speicherplatzgründen wurden die Grauwerte zuvor linear von 12 Bit auf 8 Bit skaliert.

des Netzwerkes können alle möglichen Objektkombinationen aus den unterschiedlichen Domänen generiert werden, die auf verschiedenartige Weise darstellbar sind, weil jeder dieser multiattributierten Voxelmengen eigene Visualisierungsparameter zugeordnet werden können. (vgl. Abschnitt 3.8). Abbildung 51 zeigt die Benutzeroberfläche dieses Atlases, der als VOXEL-MAN/brain [Höhne 1995] veröffentlicht wurde.

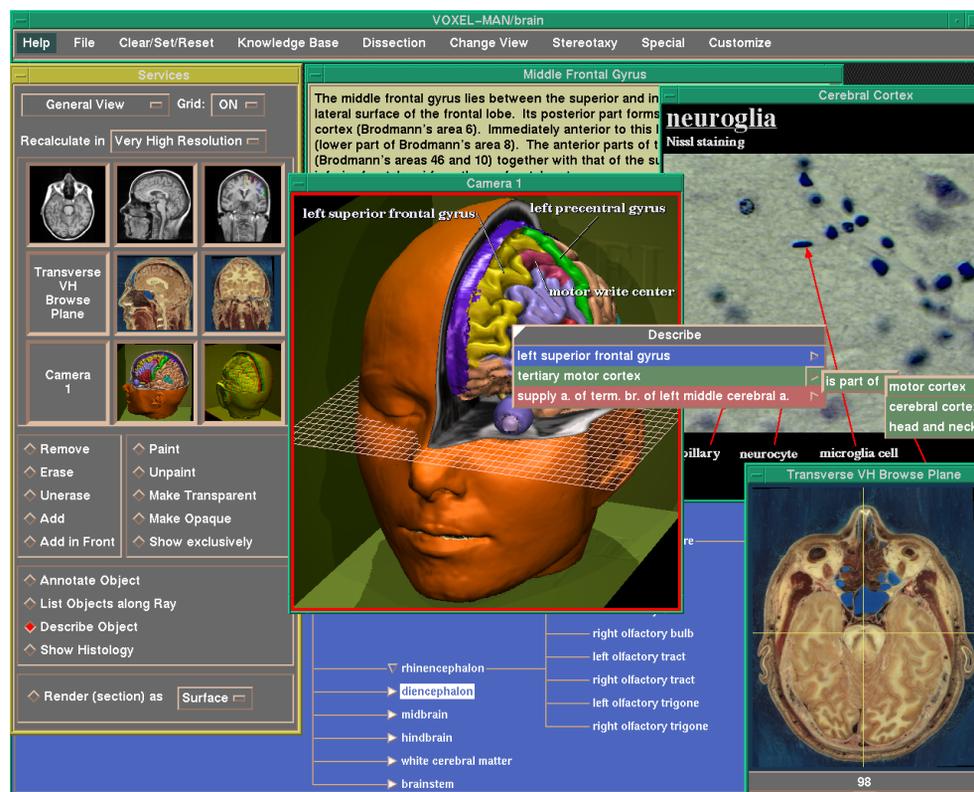


Abbildung 51: Benutzeroberfläche des VOXEL-MAN/brain Atlases.

5.2.4.1 Präparation durch selektives Schneiden

Abbildung 52 zeigt eine Spantendarstellung mit unterschiedlicher Breite der Schnittsegmente im anatomischen Modus. Hierbei sind die Schnittflächen wie die Organoberflächen schattiert und texturiert, um so einen realen anatomischen Schnitt zu simulieren. Die Präparation eines Fötens, der aus einem MR-Datensatz in über 200 Objekte segmentiert wurde, ist in Abbildung 53 gezeigt.

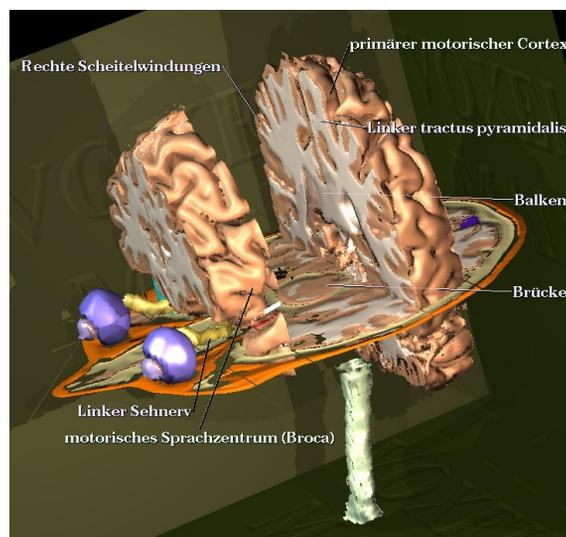


Abbildung 52: Darstellung von Spanten und Schnittflächen im anatomischen Modus, ähnlich wie sie bei einer realen Präparation entstehen würden.

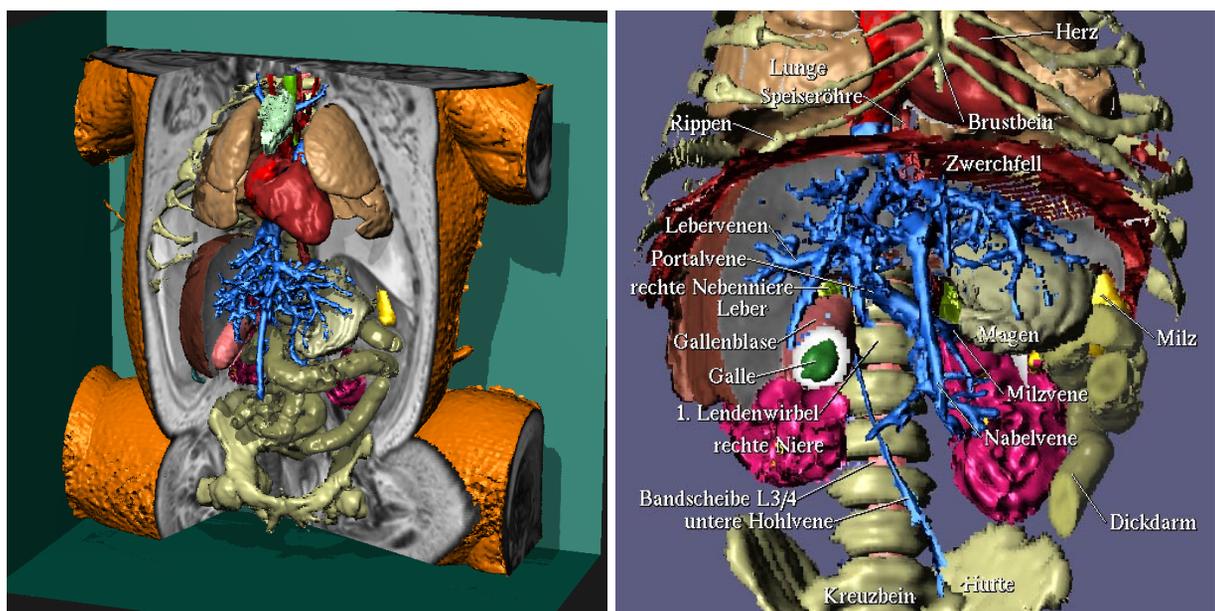


Abbildung 53: Präparation eines Fötens aus [Krämer 1997].

In Abbildung 54 (links) wurden durch die Definition einer transversalen Schnittebene Haut und Knochen im oberen Bereich des Schädels entfernt, um die farblich markierten Hirnwindungen der äußeren Hirnoberfläche sichtbar zu machen. Die rechte Darstellung zeigt den Abdruck derselben Hirnwindungen auf der darunter angrenzenden weißen Hirnsubstanz.

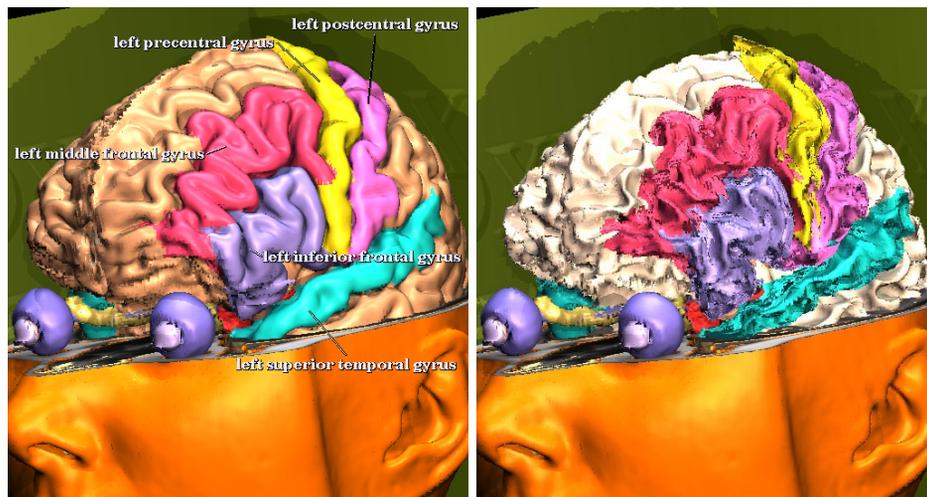


Abbildung 54: Links: Darstellung der Hirnoberfläche mit farblich markierten Hirnwindungen. Rechts: Abdruck derselben Hirnwindungen auf der darunter liegenden weißen Hirnsubstanz.

5.2.4.2 Simulation eines stereotaktischen Eingriffs

Die Simulation chirurgischer Eingriffe gewinnt insbesondere in der Ausbildung zunehmend an Bedeutung. In der Stereotaxie wird z. B. eine Kanüle in das Gehirn geschoben, um eine Gewebeprobe zu entnehmen oder um ein radioaktives Präparat an einer bestimmten Stelle zur Tumorbekämpfung zu plazieren. Das Problem ist dabei, den optimalen Weg durch das Gehirn zu finden, ohne Gefäße zu verletzen oder vitale Bereiche wie das motorische oder visuelle Zentrum zu beschädigen. Abbildung 55 zeigt die Simulation dieser speziellen Operation. Dabei wird zunächst auf einer Schnittebene das Zielgebiet, in diesem Fall der Thalamus, markiert und die Spitze des Instrumentes dorthin positioniert. Anschließend wird der äußere Einstichpunkt durch Bewegung des Instrumentes um die fixierte Spitze festgelegt. Dabei werden zu jeder Zeit die von der Nadel durchstoßenen Objekte angezeigt. In der 3D-Darstellung kann der Einstichpunkt an der Hautoberfläche abgelesen werden.

5.2.4.3 Simulation einer „intelligenten“ Röntgenbilddarstellung

Die klassische Röntgenaufnahme ist nach wie vor die am meisten benutzte Methode für die Diagnose von knöchernen Strukturen. Die richtige Interpretation der Bilder erfordert jedoch jahrelanges Training. Mittels der simulierten Röntgenbilddarstellung auf der Basis von CT-Daten, wie sie in Abbildung 56 gezeigt ist, kann für jede Strahlgeometrie bestimmt werden, welche Strukturen wieviel Röntgenstrahlen absorbieren und damit zur Intensität an jedem beliebigen Bildpunkt beitragen. Die objektbezogene freie Wahl der Darstellungsform ermöglicht es, in der Röntgenbilddarstellung einzelne Knochenglieder als Oberflächen darzustellen, andere farblich zu markieren oder zu entfernen.

5.2.4.4 Integration modellierter Gefäße

Eine Segmentation der Blutgefäße aus dem im Rahmen des Hirnatlasses verwendeten MR-Datensatz war wegen eines zu schwachen Kontrastes praktisch nicht möglich. Es wurde deshalb ausgehend von den MR-Schichten ein Gefäßbaum mit insgesamt 23.000 Polygonen modelliert

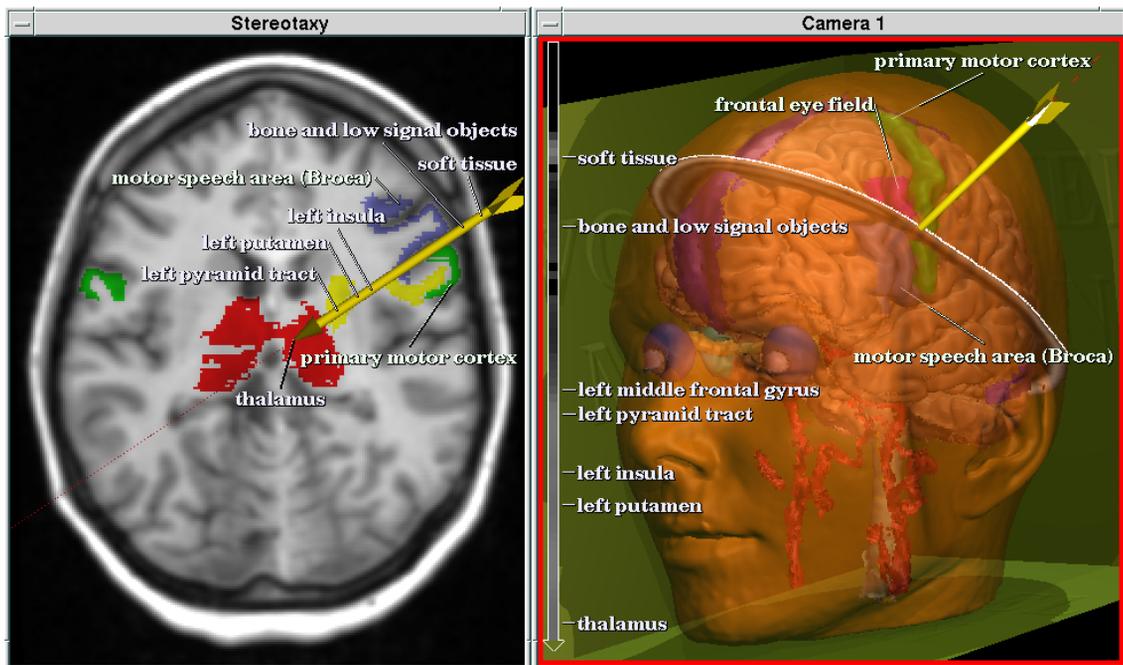


Abbildung 55: Simulation einer Stereotaxie: Die Lage der Ebene, in der sich das stereotaktische Instrument befindet, ist durch die transparente Haut hindurch sichtbar.

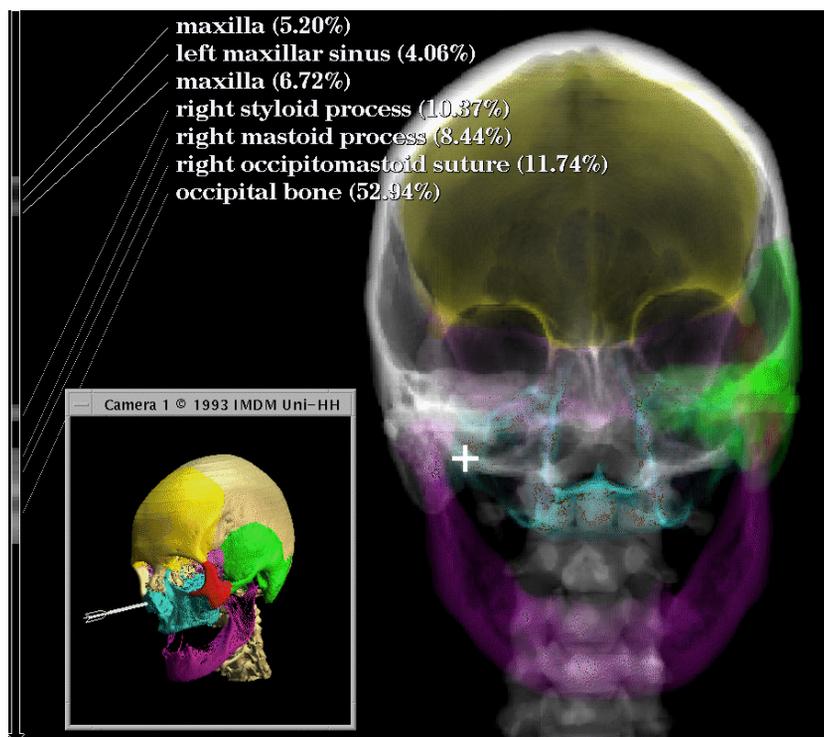


Abbildung 56: Simulation eines Röntgenbildes: Für jeden beliebigen Bildpunkt (weißes Kreuz) kann ermittelt werden, welches Objekt wieviel zur Helligkeit beiträgt.

[Rheinwald 1996]. Für die Darstellung ergaben sich zwei Möglichkeiten:

1. Hybride Darstellung der Volumenobjekte und der Polygone für die Gefäße.
2. Die Rückführung der polygonalen Beschreibung in die Volumenrepräsentation.

Abbildung 57 zeigt einen Vergleich der segmentierbaren und der modellierten Gefäße, für die die hybride Darstellung gewählt wurde. Dazu enthält die Objektbeschreibung der Gefäße anstelle der Volumendaten einen Verweis auf eine entsprechende Polygonliste. Eine ähnliche Darstellungsqualität der Gefäße in der Volumenrepräsentation kann nur durch eine Glättung, z. B. mittels einer Gaußfilterung, des Volumens der „voxelisierten“ Gefäße erzielt werden, weil keine Schwellwerte existieren, um die Oberfläche im Subvoxelbereich zu bestimmen. Für kleine polygonale Objekte ist der Speicheraufwand für ein zusätzliches Volumen kaum zu rechtfertigen. Außerdem können Strukturen, die kleiner als ein Voxel sind, nicht in der Volumenwelt repräsentiert werden.

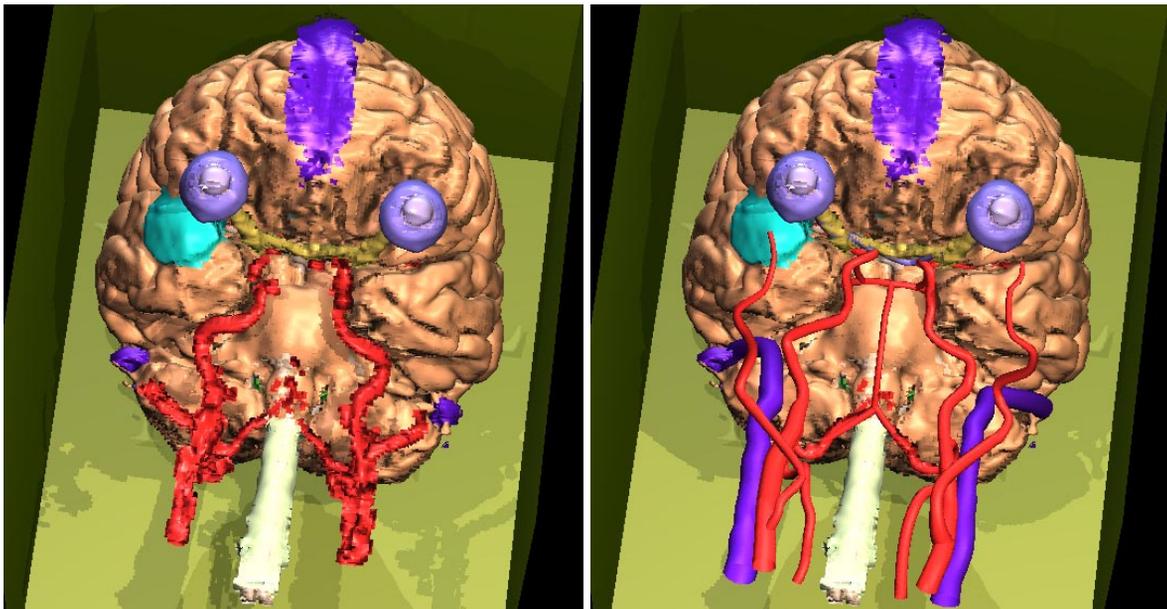


Abbildung 57: Die segmentierbaren Gefäßstrukturen in Volumenrepräsentation (links). Hybride Darstellung von Volumenobjekten und eines modellierten Gefäßbaumes mit polygonaler Repräsentation (rechts).

Eine funktional vollständige Integration, die auch ineinander verzahnte transparente Polygon- und Volumenoberflächen sowie korrekten Schattenwurf ermöglicht, ist extrem rechenzeitaufwendig, weil es für solche kombinierten Datenstrukturen keine Hardwarebeschleuniger gibt. Aus diesem Grunde wurde ein Zwei-Phasen-Verfahren implementiert, das für die polygonalen Objekte keine Transparenz und keinen Schattenwurf berechnet. Zunächst werden alle Polygone sequentiell mit Hilfe eines „Scan-Conversion“-Algorithmus ([Foley *et al.* 1990, Seite 883 ff]) in einen Z-Puffer projiziert. Dieses Verfahren ist sehr schnell, weil keine Strahlen erzeugt werden und somit die aufwendige Schnittpunktberechnung entfällt. Eine Berechnung der Schattierung braucht an dieser Stelle noch nicht durchgeführt zu werden, weil in der anschließenden zweiten Phase Teile der Polygone möglicherweise durch Volumenobjekte verdeckt werden. In der zweiten Phase wird das beschriebene Ray-Casting des Volumens durchgeführt. Einzige Änderung ist, daß der Strahl abbricht, sobald die Länge des Strahls den Wert im Z-Puffer erreicht

hat. An dieser Stelle befindet sich das aus der ersten Phase berechnete polygonale Objekt. Für dieses Objekt wird nun der Schattierungswert berechnet und gegebenenfalls bestimmt, ob auf dieses Element durch ein Volumenobjekt ein Schlagschatten geworfen wird. Im Gegensatz zu dem Verfahren von [Kaufman *et al.* 1990] wurde hier die Reihenfolge der Phasen vertauscht, so daß die Polygone auch durch transparente volumenbasierte Objekte hindurch sichtbar gemacht werden können. Dies ist bei dem Kaufmanschen Verfahren nicht möglich.

5.2.4.5 Verwendung von Texturen

Einen großen Einfluß auf die Wiedererkennbarkeit von Organen und Strukturen hat das Erscheinungsbild der äußeren Oberfläche. Die Geometrie kann zwar mit den beschriebenen Methoden hinreichend genau rekonstruiert werden, aber Farbe und Feinstruktur der Oberflächen lassen sich aus klinischen Daten nicht gewinnen. Eine Alternative bietet die Texturierung der Organe durch Photographien. Besonders einfach läßt sich die Texturierung der Hautoberfläche mittels Diaprojektion erreichen. Dabei wird eine Photographie des Objekts zwischen dem virtuellen Betrachter und dem Objekt in den Sehstrahlengang positioniert. Bei richtiger Skalierung und Drehung werden Photographie und Objekt deckungsgleich. Anstelle der künstlichen Objektfarbe kann nun die Farbe des projizierten Bildes in das Beleuchtungsmodell eingehen. Die Abbildungen 58 und 59 zeigen Beispiele für diese Methode.

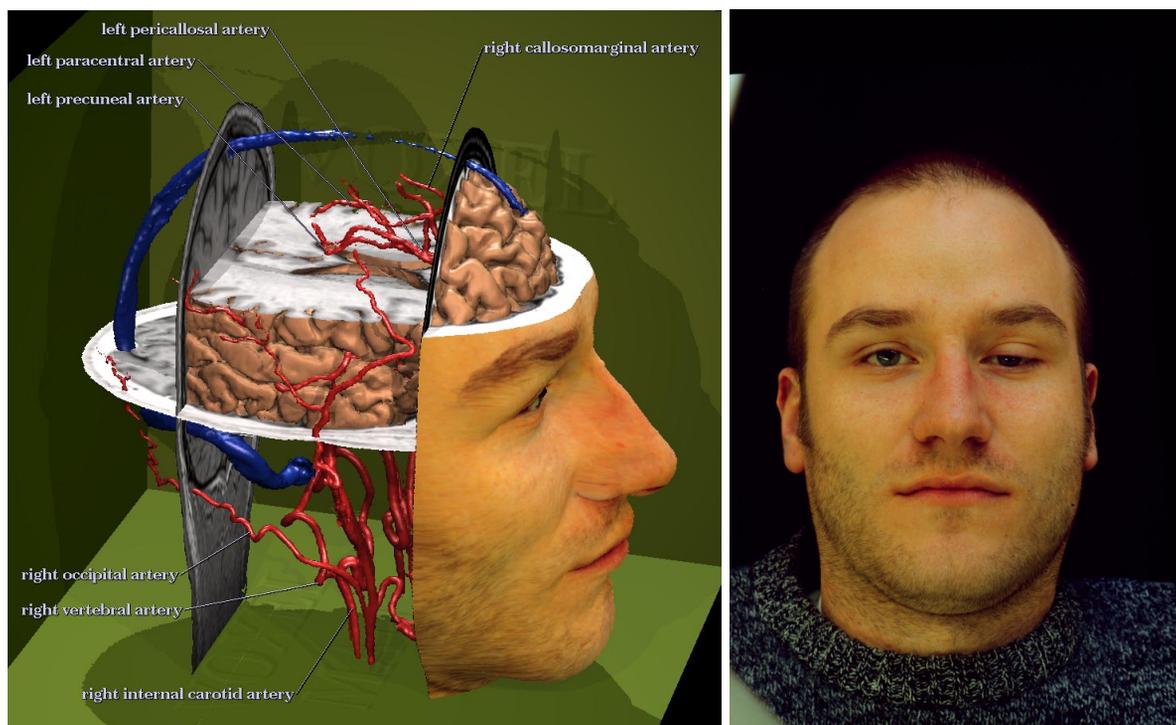


Abbildung 58: Rekonstruktion verschiedener Organe aus einem multiparametrischen MR-Datensatz. Die Hautoberfläche wurde mit einer frontalen Photographie des Probanden texturiert [Urban *et al.* 1997].

5.2.4.6 Erzeugung von Lehrfilmen

In Tabelle 5 sind die Gesamtrechnenzeiten für die Strahlabtastung, Berechnung der Oberfläche im Subvoxelbereich, Bestimmung der Oberflächennormalen aus dem Grauwertgradienten,



Abbildung 59: Rekonstruktion des Kopfes einer ägyptischen Mumie aus CT-Daten. Die Umhüllung der Mumie wurde wie in Abbildung 58 mit einer frontalen Photographie texturiert [Höhne et al. 1997b].

Berechnung der Lichtreflexion und gegebenenfalls die Berechnung des Schlagschattens von einigen der in dieser Arbeit gezeigten 3D-Bilder zusammengestellt. Daraus ist zu erkennen, daß für echtes interaktives Verhalten die Rechnerleistung gegenüber einer heutigen Standardworkstation oder einem Personal-Computer etwa 10 bis 100 mal größer sein müßte. Dagegen lassen sich die vorgestellten Verfahren aber ideal für die Erzeugung von Einzelbildern und Filmen zu Demonstrations- und Lehrzwecken nutzen [Höhne et al. 1994]. Dieses Material wird nur einmal hergestellt und dann vielfach verwendet, so daß auch längere Berechnungszeiten gerechtfertigt sind.

Abb.	Bildgröße [Pixel]	Datensatzgröße [Voxel, MB]	Rechenzeit [s]	Bemerkung
57	538×558	150×200×192×5, 27.5	113	ohne Schatten
57	538×558	”	538	mit Schatten
49	700×545	170×201×150×3, 14.7	595	Textur + Schatten
44	512×545	183×246×277×4, 97.1	76	mit Bounding-Volume
45	512×545	208×256×242×3, 36.9	85	MIP
45	512×545	”	545	Oberfläche + Schatten

Tabelle 5: Rechenzeiten: Ermittelt auf einer DEC-station 5000/300 (Alpha 21064 CPU, 150 MHz Prozessortakt, 256 MB Hauptspeicher)

Das Konzept der „intelligenten“ Bilder ermöglicht die Erzeugung von Bildsequenzen mit erweiterter Funktionalität. Hierbei werden neben den eigentlichen Bildmatrizen weitere Informationen wie die Objektkennung und der Z-Puffer für jedes Einzelbild mitgespeichert. Mit diesen Zusatzinformationen lassen sich verschiedene Operationen, z. B. das anatomisch kor-

rekte Beschriften und Einfärben der Objekte, nachträglich realisieren, ohne die Originalvolumina bereithalten und eine Bildberechnung durchführen zu müssen [Höhne *et al.* 1996b]. Die Speicherung erfolgt in einer Erweiterung des von Apple entwickelten QuickTimeVR (Virtual Reality) Formats, das dem Benutzer zwei Freiheitsgrade für die Navigation zur Verfügung stellt. Diese beiden Freiheitsgrade können für unterschiedliche Funktionen genutzt werden, z. B. um ein Objekt, das sich innerhalb einer virtuellen Kugel befindet, entlang der Längengrade und Breitenkreise ähnlich wie bei einem Globus zu drehen. Dazu wird jeweils am Schnittpunkt der gewählten Rastereinteilung von Längen- und Breitengrad die entsprechende Ansicht berechnet und gespeichert. Der erste Freiheitsgrad kann auch eine Rotationsbewegung parametrisieren, während mit dem zweiten Freiheitsgrad eine Schnittebene durch das Volumen bewegt wird oder die Objekte Schicht für Schicht entfernt werden. Mittels sogenannter „hot spots“, das sind speziell markierte Regionen in einem Bild, lassen sich Verzweigungen realisieren, so daß z. B. der Benutzer bei einer Fahrt durch das Blutgefäßsystem entscheiden kann, welchen Weg er an einer Gefäßteilung weiter verfolgen will. Die Erzeugung solcher Filme wird durch die Skriptfähigkeit der Beschreibungssprache realisiert, mit der komplexe Szenen und Abläufe programmierbar sind. Die Parametrisierung ermöglicht auch eine Berechnung mit verminderter Auflösung für eine schnelle Vorschau, bevor der endgültige Film gerechnet wird. Eine Sammlung interaktiver intelligenter Filme für die Ausbildung in Anatomie und Radiologie ist unter dem Namen VOXEL-MAN Jr. verfügbar [Höhne 1998].

5.2.5 Strahlentherapieplanung

In der Strahlentherapie geht es um die Zerstörung von entartetem Gewebe mit Hilfe hochenergetischer Strahlung. Die Planung einer solchen Therapie ist schwierig, weil zum einen der Tumor mit einer möglichst hohen Dosis bestrahlt werden soll, andererseits aber angrenzende lebenswichtige Organe keine oder nur eine geringe Dosis erhalten dürfen. Die Visualisierung eines Bestrahlungsplans auf der Basis von CT-Daten erfordert die gleichzeitige Darstellung verschiedener Strukturen:

1. die Organe, soweit sie aus den CT-Daten segmentierbar sind, als Orientierungshilfe;
2. die Risikoorgane, die nicht bestrahlt werden dürfen;
3. das sogenannte Zielvolumen, d. h. also die Region, die eine möglichst hohe Dosis erhalten soll. Dieses Zielvolumen stimmt häufig nicht mit Organ- oder Tumorgrenzen überein und wird in der Regel manuell definiert;
4. die konkrete Dosisverteilung, die aufgrund einer bestimmten Parametrisierung der Strahlenquellen durch Simulation errechnet wird.

Die Beurteilung, ob eine bestimmte Parametrisierung optimal ist, wird heute in der Regel noch in 2D durchgeführt. Mit dem hier vorgestellten Konzept für multiattributierte Volumendaten wurde eine kombinierte 3D-Darstellung aller Informationen möglich. Ähnlich wie bei dem zuvor beschriebenen Atlas wurden hier verschiedene Domänen definiert, um Organe, Dosisverteilungen und Zielvolumina zu beschreiben. Zusätzlich wurde ähnlich wie im Beispiel der PET-Visualisierung eine Farbtabelle benutzt, um unterschiedlich starke Strahlenkonzentrationen visuell besser unterscheiden zu können. Abbildung 60 zeigt die Visualisierung für eine bestimmte Einstellung der Bestrahlungsgeräte. Deutlich ist zu erkennen, daß die kritische Isodosislinie (rot) außerhalb des Zielvolumens (violett) liegt und das Herz als Risikoorgan teilweise überdeckt. Eine Beurteilung, ob der Plan optimal ist oder geändert werden sollte,

wird mit dieser Darstellungsart deutlich erleichtert.

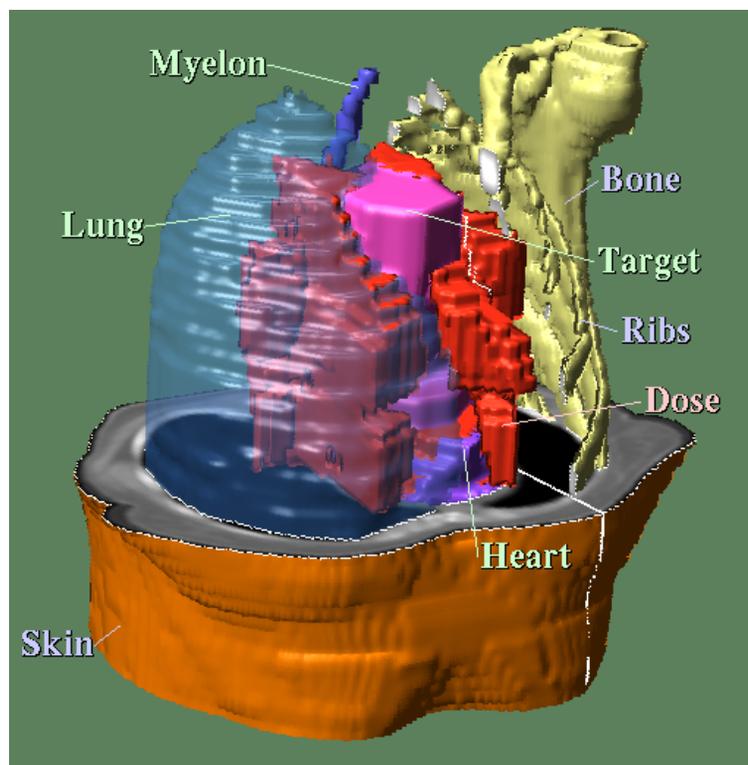


Abbildung 60: *Strahlentherapieplanung: Kombinierte Darstellung von Organen, 90% Dosisbelastung (rot) und Zielvolumen (violett) (aus [Frenzel 1996]).*

Abbildung 61 zeigt eine Therapieplanung, bei der CT- und MR-Volumina erzeugt und registriert wurden. Auf der Basis der CT-Daten wurde die Dosisverteilung berechnet und der knöcherne Schädel segmentiert. Für Gehirn und Augen wurde das MR-Volumen benutzt. In einem fünften Volumen wurden zusätzlich verschiedene Regionen, sogenannte „volumes of interest“ (VOI), definiert. Das Gehirn sowie die sagittale und transversale Schnittebene wurden mit den Dosisdaten texturiert, wobei die Dosiswerte mittels einer Tabelle in Farbwerte umgesetzt wurden.

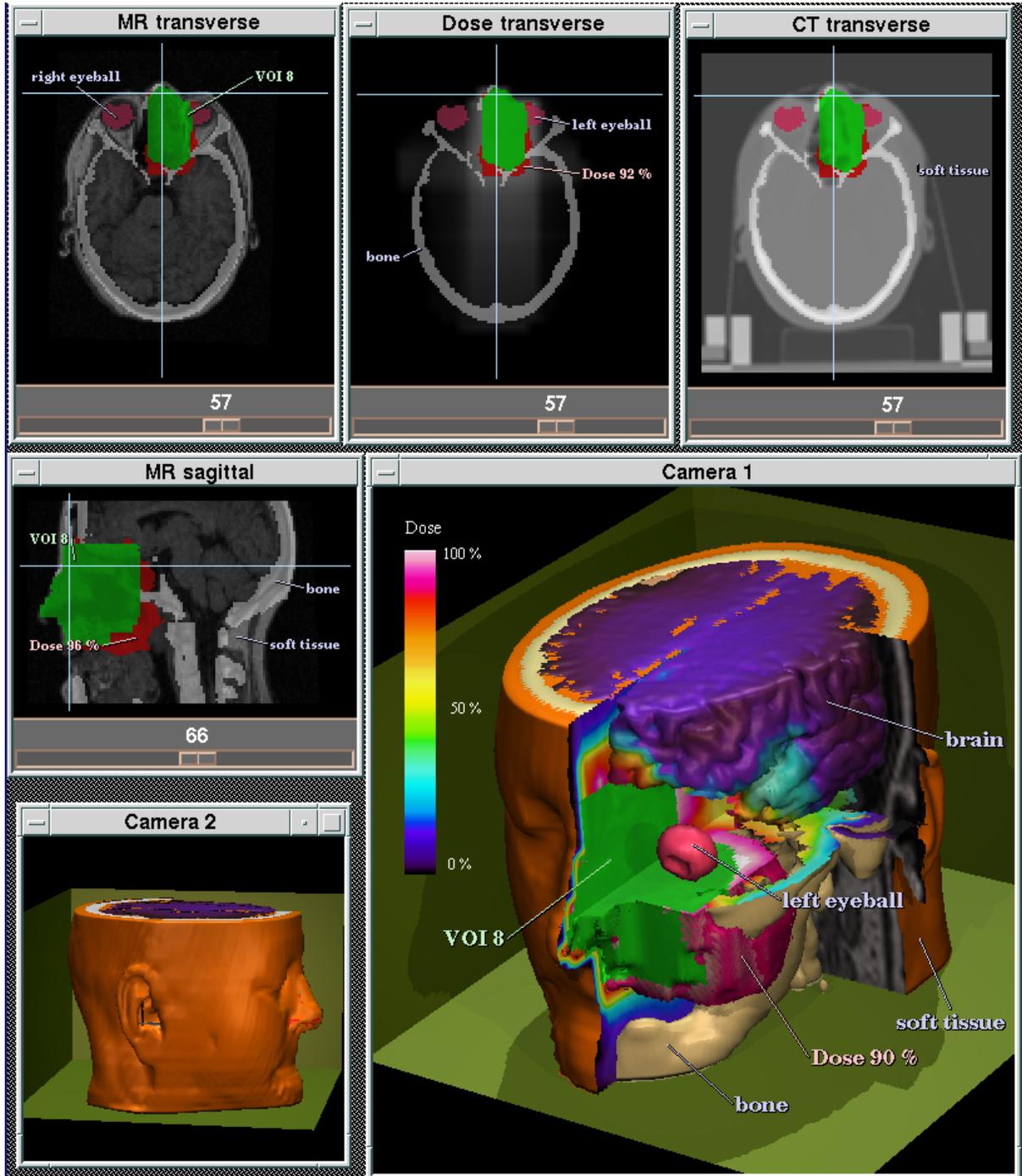


Abbildung 61: Strahlentherapieplanung: Texturierung des Gehirns und der Schnittebenen in der 3D-Darstellung mit der Dosisverteilung: An jedem Ort kann mit Hilfe der Farbtabelle die jeweilige Strahlendosis abgelesen werden.

6 Schlußfolgerungen

Kern der vorliegenden Arbeit war die Entwicklung und Implementation eines erweiterten Strahlverfolgungsverfahrens für die Bestimmung von Objektgrenzen im Subvoxelbereich für die Visualisierung von tomographischen Volumendaten. Die Methode basiert auf der Untersuchung der Umgebung der Abtastposition, um den besten Attributwert anhand der bei der Segmentation definierten Klassifikatoren wie Schwellwerten für skalare Bildvolumina oder Ellipsoide für RGB-Daten zu bestimmen. Die Lokalisation der Grenzflächen benachbarter Objekte erfolgt durch Interpolation oder Binärsuche entlang des Abtaststrahls ebenfalls unter Verwendung der jeweiligen Klassifikation. Diese wesentlich verbesserte Bestimmung der Oberfläche im Subvoxelbereich ermöglichte auch eine genauere Berechnung des Grauwertgradienten, der für die Oberflächennormale und damit für die visuelle Erscheinung der 3D-Darstellung von entscheidender Bedeutung ist. An verschiedenen Beispielen wurde die Überlegenheit des Verfahrens, insbesondere bei starken Vergrößerungen, gegenüber klassischen Ray-Casting-Algorithmen demonstriert. Anhand eines künstlichen Objekts und eines Blutgefäßdatensatzes wurde gezeigt, daß auch sehr kleine, nur voxelgroße Objekte ohne Voxelisierungsartefakte dargestellt werden können. Aus den Farbschnittbildern des Visible-Human-Datensatzes wurden mit diesem Verfahren nahezu photorealistische Abbildungen erzeugt.

Die neue Qualität der 3D-Darstellung erfordert einen deutlich höheren Rechenaufwand, der den Einsatz vor allem für die Erzeugung von Demonstrationsbildern und Animationen für Ausbildungszwecke als sinnvoll erscheinen läßt. Um die Visualisierung sehr großer Datensätze wie z. B. vom Visible-Human mit der hier aufgezeigten Qualität in Echtzeit durchführen zu können, sind etwa 1000 mal schnellere Rechner notwendig. Die rasche Entwicklung der Rechnertechnologie und die hervorragende Eignung der Strahlverfolgungsmethode für eine Parallelisierung [Whitman *et al.* 1994] machen dieses Ziel jedoch in wenigen Jahren erreichbar.

Eine Untersuchung, die die Genauigkeit der Methode gegenüber herkömmlichen Verfahren quantitativ erfaßt, wurde nicht durchgeführt. Dazu ist die Entwicklung geeigneter Testobjekte und Meßverfahren notwendig. Dieses Thema ist unter anderem Gegenstand einer weiteren Untersuchung [Pommert 1999].

In dieser Arbeit wurde darüberhinaus ein Konzept entwickelt, das eine flexible Kombination verschiedenartiger Visualisierungsformen von multiattributierten und multiparametrischen tomographischen Volumendaten innerhalb einer 3D-Darstellung ermöglicht. Das Konzept erlaubt die Zuordnung mehrerer beliebiger Attribute zu den Voxeln und deren mengentheoretische Verknüpfung.

Im Fall eines dreidimensionalen Hirnatlasses auf der Basis eines MR-Volumens wurde dies mit den Domänen Morphologie, Funktionsareale und Blutversorgung demonstriert. Für die Anwendung in der Strahlentherapieplanung wurden multiple Objekte aus den Domänen Morphologie, Dosisverteilung, Zielvolumina und Risikoorgane gleichzeitig visualisiert. Aus den multiparametrischen Visible-Human-Daten wurde die Lage und Orientierung der Knochenoberfläche aus den CT-Daten berechnet, während die Oberflächenfarbe und die Weichteile aus den anatomischen Farbschnittbildern gewonnen wurden.

Mit der individuellen Parametrisierung der Darstellung der Objekte, die durch frei wählbare

Schnittebenen regionenabhängig variiert werden kann, wurde eine bisher nicht mögliche gleichzeitige Darstellung verschiedener Aspekte in einem Bild erreicht. So konnten z. B. Oberflächen von Objekten zusammen mit künstlichen Röntgenbilddarstellungen, Maximum-Intensitäts-Projektionen oder multiplen transparenten Oberflächen erzeugt werden.

Durch die Integration polygonbasierter Objekte konnten auch nicht aus den Datenvolumina segmentierbare oder nicht in den Daten enthaltene Strukturen wie z. B. künstliche Gefäße und Werkzeuge dargestellt werden.

Die beschriebenen Methoden bilden den Visualisierungskern des VOXEL-MAN-Systems, das in Form eines dreidimensionalen Atlases des Kopfes publiziert ist [Höhne 1995]. Die praktische Benutzbarkeit des komplexen Systems wird durch die in dieser Arbeit entwickelte Beschreibungssprache gewährleistet, die eine konsistente Spezifikation komplexer Szenen mit Objekten, beliebig orientierten Schnittebenen und unterschiedlichen Darstellungsverfahren ermöglicht. Die Skriptfähigkeit dieser Sprache bildete die Voraussetzung für die Herstellung verschiedener Animationen wie z. B. die Simulation des zeitlichen Verlaufs einer Kontrastmittelgabe in einem Blutgefäßbaum oder die Fahrt durch die Speiseröhre in den Magen des Visible-Human. Weitere Beispiele sowie ein Lehrfilm über die Entwicklung von Röntgenbildern hin zu virtuellen Körpermodellen, der vollständig mit dem System erzeugt wurde, sind als Video verfügbar [Höhne *et al.* 1997a].

Eine Weiterentwicklung der beschriebenen volumenbasierten Visualisierungsverfahren erscheint in verschiedene Richtungen sinnvoll:

1. Die Darstellung sehr großer Datensätze wie etwa des Visible-Human:
Hierbei ist die Auflösung der Daten größer als die des Bildschirms, so daß immer nur ein Teil der Daten sichtbar ist. Erst durch „Zoomen“ wird der jeweils interessierende Ausschnitt aus den Daten bis hin zur vollen Auflösung extrahiert. Hierfür scheinen besonders „Multiresolution-Datenstrukturen“ wie die Wavelet-Transformation geeignet zu sein [Stollnitz *et al.* 1995].
2. Verbesserte Schattierungsmodelle:
Für noch realistischere Darstellungen sind komplexere, aber auch rechenzeitaufwendigere Reflexionsmodelle notwendig, die die spezifischen Materialeigenschaften der Objekte besser modellieren als das Phong'sche Beleuchtungsmodell. Methoden wie das Torrance/Sparrow-Modell oder Radiosity-Verfahren erfordern die Kenntnis dieser Eigenschaften, die bisher nur für wenige in der Regel metallische Stoffe bekannt sind. Für natürliche Objekte wie die menschlichen Organe (z. B. Leber, Knochen, Gefäße, Hautoberflächen) bietet sich eher die Texturierung mit geeigneten Vorlagen an.
3. Erweiterte Manipulationsmöglichkeiten:
Eine Einschränkung des derzeitigen Systems liegt in der „starr“ Objekthandhabung, d. h., einzelne Objekte können nicht gegeneinander bewegt werden, um z. B. eine Gelenkbewegung zu simulieren. Im Gegensatz zur polygonalen Repräsentation ist eine solche Funktion im Volumenmodell nicht trivial zu realisieren [Pfleger 1992]. Für weitergehende realistischere Manipulationen sind neben der Objektbewegung auch die Entwicklung endoskopischer Werkzeuge und die Modellierung von Objektdeformationen [Schiemann 1998] für die Visualisierung von Wachstums- bzw. Verdrängungsprozessen denkbar.

Literaturverzeichnis

- [Badouel 1990] Didier Badouel. An efficient ray-polygon intersection. In Andrew S. Glassner (Ed.), *Graphic Gems*, 390–393. Academic Press, Inc., 1990. 8, 32
- [Ballard, Brown 1982] D. H. Ballard, C. M. Brown. *Computer Vision*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1982. 24
- [Blinn 1991] James F. Blinn. A trip down the graphics pipeline: Line clipping. *IEEE Comput. Graphics & Appl.*, 11(1):98–105, 1991. 8
- [Blinn 1993] James F. Blinn. A trip down the graphics pipeline: The homogenous perspective transform. *IEEE Comput. Graphics & Appl.*, 13(3):75–80, 1993. 17, 95
- [Blinn 1996] James F. Blinn. Calculating screen coverage. *IEEE Comput. Graphics & Appl.*, 16(3):84–88, 1996. 49
- [Bomans 1994] Michael Bomans. *Segmentationsverfahren zur 3D-Visualisierung von Kernspintomogrammen des Kopfes: Evaluierung der Standardverfahren und Entwurf und Realisierung eines klinisch einsetzbaren Systems*. Dissertation, Fachbereich Informatik, Universität Hamburg, 1994. 1, 6, 18, 26
- [Dahlmanns 1997] Thomas Dahlmanns. Entwicklung eines Texture-Mapping-Programmsystems für medizinische Anwendungen der Volumenvisualisierung. Diplomarbeit, Fachbereich Informatik, Universität Hamburg, 1997. 36
- [Drebin *et al.* 1988] Robert A. Drebin, Loren Carpenter, Pat Hanrahan. Volume rendering. *Comput. Graphics*, 22(4):65–74, 1988. 12, 15
- [Foley *et al.* 1990] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes (Eds.). *Computer Graphics: Principles and Practice, Second Edition*. The System Programming Series. Addison-Wesley Publishing Company, 1990. 8, 8, 9, 9, 75
- [Frenzel 1996] Thorsten Frenzel. Dosimetrie und Qualitätssicherung bei der Verwendung von Modifikatoren. Diplomarbeit, Fachbereich Physik, Universität Hamburg, 1996. 79
- [Frieder *et al.* 1985] Gideon Frieder, Dan Gordon, R. Anthony Reynolds. Back-to-front display of voxel-based objects. *IEEE Comput. Graphics Appl.*, 5(1):52–59, 1985. 10
- [Fuchs *et al.* 1977] Henry Fuchs, Z. M. Kedem, S. P. Uselton. Optimal surface reconstruction from planar contours. *Commun. ACM*, 20(10):693–702, 1977. 10
- [Gibson 1998] Sarah F. F. Gibson. Using distance maps for accurate surface representation in sampled volumes. In *IEEE Symposium on Volume Visualization, Oct. 19-20, 1998*, 23–30, 1998. 26
- [Goldwasser, Reynolds 1987] S. M. Goldwasser, R. A. Reynolds. Real-time display and manipulation of 3d-medical objects: The voxel processor architecture. *Computer Vision, Graphics and Image Processing*, 39(1):1–27, 1987. 10
- [Greenberg *et al.* 1986] Donald P. Greenberg, Michael F. Cohen, Kenneth E. Torrance. Radiosity: A method for computing global illumination. *The Visual Computer*, 2(5):291–297, 1986. 9

- [Grimstead 1996] Ian Grimstead. Ray tracing. World Wide Web: <http://www.cm.cf.ac.uk/Ray.Tracing/>, 1996. 8
- [Hall 1989] Roy Hall. *Illumination and Color in Computer Generated Imagery*. Springer-Verlag, New York, 1989. 9
- [Heckbert 1986] Paul S. Heckbert. Survey of texture mapping. *IEEE Comput. Graphics & Appl.*, 6(11):56–67, 1986. 36
- [Hendrick 1993] R. Edward Hendrick (Ed.). *MRI: principles and artifacts*. The Raven MRI teaching file. Raven Press, 1993. 1
- [Herman, Liu 1979] Gabor T. Herman, Hsun Kao Liu. Three-dimensional display of human organs from computed tomograms. *Comput. Graphics Image Process.*, 9:1–21, 1979. 10
- [Höhne, Bernstein 1986] Karl Heinz Höhne, Ralph Bernstein. Shading 3D-images from CT using gray level gradients. *IEEE Trans. Med. Imaging*, MI-5(1):45–47, 1986. 11, 20
- [Höhne *et al.* 1987] Karl Heinz Höhne, Martin Riemer, Ulf Tiede. Viewing operations for 3D-tomographic gray level data. In Heinz U. Lemke, Michael L. Rhodes, C. Carl Jaffe, Roland Felix (Eds.), *Computer Assisted Radiology, Proc. CAR '87*, 599–609. Springer-Verlag, Berlin, 1987. 15, 30
- [Höhne *et al.* 1988] Karl Heinz Höhne, Michael Bomans, Ulf Tiede, Martin Riemer. Display of multiple 3D-objects using the generalized voxel-model. In R. H. Schneider, S. J. Dwyer (Eds.), *Medical Imaging II, Part B, Proc. SPIE 914*, 850–854, Newport Beach, 1988. 15
- [Höhne *et al.* 1992] Karl Heinz Höhne, Michael Bomans, Martin Riemer, Rainer Schubert, Ulf Tiede, Werner Lierse. A 3D anatomical atlas based on a volume model. *IEEE Comput. Graphics Appl.*, 12(4):72–78, 1992. 69
- [Höhne *et al.* 1994] Karl Heinz Höhne, Bernhard Pflesser, Andreas Pommert, Rainer Schubert, Thomas Schiemann, Ulf Tiede. Anatomic realism in radiologic imaging. *Radiology*, 193(P):430, 1994. (abstract). 77
- [Höhne *et al.* 1995a] Karl Heinz Höhne, Bernhard Pflesser, Andreas Pommert, Martin Riemer, Thomas Schiemann, Rainer Schubert, Ulf Tiede. Radiological imaging in the second century after Röntgen: Realistic and intelligent. In Heinz U. Lemke, Kiyonari Inamura, C. Carl Jaffe, Michael W. Vannier (Eds.), *Computer Assisted Radiology, Proc. CAR '95*, 119–124. Springer-Verlag, Berlin, 1995. 42
- [Höhne *et al.* 1995b] Karl Heinz Höhne, Bernhard Pflesser, Andreas Pommert, Martin Riemer, Thomas Schiemann, Rainer Schubert, Ulf Tiede. A new representation of knowledge concerning human anatomy and function. *Nature Med.*, 1(6):506–511, 1995. 70
- [Höhne *et al.* 1996a] Karl Heinz Höhne, Bernhard Pflesser, Andreas Pommert, Martin Riemer, Thomas Schiemann, Rainer Schubert, Ulf Tiede. A virtual body model for surgical education and rehearsal. *IEEE Computer*, 29(1):25–31, 1996. 70
- [Höhne *et al.* 1996b] Karl Heinz Höhne, Kay Priesmeyer, Martin Riemer, Thomas Schiemann, Rainer Schubert, Ulf Tiede, Andreas Pommert, Hans-Christian Wulf. Exploring the Visible Human via “intelligent movies”. *Radiology*, 201(P):564, 1996. (abstract). 78

- [Höhne *et al.* 1997a] Karl Heinz Höhne, Michael Bomans, Thomas Dahlmanns, Zdenek Halata, Nils Hausig, Wolf-Joachim Hölzje, Christoph Koch, Henning Krämer, Werner Lierse, Rainer Maas, Jochen Nuthmann, Bernhard Pflesser, Andreas Pommert, Kay Priesmeyer, Uwe Rehder, Martin Riemer, Thomas Schiemann, Rainer Schubert, Christian Seebode, Ulf Tiede, Volker Wening. From X-ray imaging to virtual body models. Video, Institute of Mathematics and Computer Science in Medicine, University of Hamburg, 1997. 82
- [Höhne *et al.* 1997b] Karl Heinz Höhne, Kay Priesmeyer, Maike Süverkrübbe, Ulf Tiede. Die Virtuelle Mumie: Auswickeln einer Mumie per Mausclick. World Wide Web: <http://www.uke.uni-hamburg.de/institute/imdm/idv/forschung/mumie/mumie.de.html>, 1997. 77
- [Höhne, Hanson 1992] Karl Heinz Höhne, W. A. Hanson. Interactive 3D-segmentation of MRI and CT volumes using morphological operations. *J. Comput. Assist. Tomogr.*, 16(2):285–294, 1992. 1, 8
- [Höhne, Pommert 1996] Karl Heinz Höhne, Andreas Pommert. Volume visualization. In Arthur W. Toga, John C. Mazziotta (Eds.), *Brain Mapping*, chapter 17, 423–443. Academic Press, San Diego, CA, 1996. 16
- [Höhne 1995] Karl Heinz Höhne (Ed.). *VOXEL-MAN, Part 1: Brain and Skull, Version 1.0*. Springer-Verlag Electronic Media, Heidelberg, 1995. (CD-ROM, ISBN 3-540-14517-6). 44, 70, 71, 82
- [Höhne 1998] Karl Heinz Höhne (Ed.). *VOXEL-MAN Junior: Interactive 3D Anatomy and Radiology in Virtual Reality Scenes, Part 1: Brain and Skull*. Springer-Verlag Electronic Media, Heidelberg, 1998. (CD-ROM, ISBN 3-540-14676-8). 78
- [Jackèl 1985] Dietmar Jackèl. The graphics parcum system. *Computer Graphics Forum*, 4(1):21–32, 1985. 10
- [Kaufman, Bakalash 1988] Arie Kaufman, R. Bakalash. Memory and processing architecture for 3D voxel-based imagery. *IEEE Comput. Graphics Appl.*, 8(11):10–23, 1988. 10
- [Kaufman *et al.* 1990] Arie Kaufman, Roni Yagel, Daniel Cohen. Intermixing surface and volume rendering. In Karl Heinz Höhne, Henry Fuchs, Stephen M. Pizer (Eds.), *3D-Imaging in Medicine: Algorithms, Systems, Applications*, volume 60 of *NATO ASI Series F*, 217–227. Springer-Verlag, Berlin, 1990. 76
- [Kaufman *et al.* 1993] Arie Kaufman, Daniel Cohen, Roni Yagel. Volume graphics. *IEEE Computer*, 26(7):51–64, 1993. 12
- [Kaufman *et al.* 1994] Arie Kaufman, Karl Heinz Höhne, Wolfgang Krüger, Larry J. Rosenblum, Peter Schröder. Research issues in volume visualization. *IEEE Comput. Graphics Appl.*, 14(2):63–67, 1994. 16
- [Kaufman 1991] Arie Kaufman (Ed.). *Volume Visualization*. IEEE Computer Society Press, Los Alamitos, CA, 1991. 10, 16
- [Kaufman 1996] Arie Kaufman. Volvis - Volume Visualization System. World Wide Web: <http://www.cs.sunysb.edu/~vislab/>, 1996. 16

- [Klimaszewski, Sederberg 1997] Kris Klimaszewski, Thomas W. Sederberg. Faster ray tracing using adaptive grids. *IEEE Comput. Graphics & Appl.*, 17(1):42–51, 1997. 49
- [Krämer 1997] Henning Krämer. Interaktiver 3D-Atlas eines menschlichen Fötens. Dissertation, Fachbereich Medizin, Universität Hamburg, 1997. 70, 72
- [Levoy 1988] Marc Levoy. Display of surfaces from volume data. *IEEE Comput. Graphics Appl.*, 8(3):29–37, 1988. 12
- [Lorensen, Cline 1987] William E. Lorensen, Harvey E. Cline. Marching Cubes: A high resolution 3D surface construction algorithm. *Comput. Graphics*, 21(4):163–169, 1987. 11
- [Meagher 1982] D. J. Meagher. Geometric modeling using octree encoding. *Comput. Graphics Image Process.*, 19(2):129–147, 1982. 49
- [Montgomery 1998] Kevin Montgomery. 3D reconstruction software. World Wide Web: <http://biocomp.stanford.edu/3dreconstruction/index.html>, 1998. 16
- [National Library of Medicine 1996] National Library of Medicine. The Visible Human Project. World Wide Web: http://www.nlm.nih.gov/research/visible/visible_human.html, 1996. 59
- [Ney *et al.* 1991] Derek Ney, Elliot K. Fishman, Donna Magid, D. D. Robinson, A. Kawashima. Three-dimensional volumetric display of CT data: Effect of scan parameters upon image quality. *J. Computer Assisted Tomography*, 15(5):875–885, 1991. 6
- [Perlin 1985] Ken Perlin. An image synthesizer. In *Proceedings SIGGRAPH '85, July 22-26*, volume 19, 287–296, 1985. 36
- [Pfister 1996] Hanspeter Pfister. *Architecture for Real-Time Volume Rendering*. Dissertation, Department of Computer Science, State University New York at Stony Brook, 1996. 10
- [Pflesser *et al.* 1995] Bernhard Pflesser, Ulf Tiede, Karl Heinz Höhne. Towards realistic visualization for surgery rehearsal. In Nicholas Ayache (Ed.), *Computer Vision, Virtual Reality and Robotics in Medicine, Proc. CVRMed '95*, volume 905 of *Lecture Notes in Computer Science*, 487–491. Springer-Verlag, Berlin, 1995. 35
- [Pflesser 1992] Bernhard Pflesser. Simulation und 3D-Darstellung von Bewegungen medizinischer Objekte im Grauwert-Volumen-Modell. Diplomarbeit, Fachbereich Informatik, Universität Hamburg, 1992. 82
- [Pietrzyk 1994] Uwe Pietrzyk. Registrierung von PET- und MR-Volumendaten. Persönliches Gespräch, MPI für neurologische Forschung, Köln, 1994. 68
- [Pommert *et al.* 1990] Andreas Pommert, Ulf Tiede, Gunnar Wiebecke, Karl Heinz Höhne. Surface shading in tomographic volume visualization: A comparative study. In *First Conference on Visualization in Biomedical Computing, Proc. VBC '90*, 19–26. IEEE Computer Society Press, Los Alamitos, CA, 1990. 20, 23

- [Pommert *et al.* 1991] Andreas Pommert, Wolf-Joachim Hölzje, Nikolaus Holzknicht, Ulf Tiede, Karl Heinz Höhne. Accuracy of images and measurements in 3D bone imaging. In Heinz U. Lemke, Michael L. Rhodes, C. Carl Jaffe, Roland Felix (Eds.), *Computer Assisted Radiology, Proc. CAR '91*, 209–215. Springer-Verlag, Berlin, 1991. 6
- [Pommert *et al.* 1992a] Andreas Pommert, Michael Bomans, Karl Heinz Höhne. Volume visualization in magnetic resonance angiography. *IEEE Comput. Graphics Appl.*, 12(5):12–13, 1992. 14
- [Pommert *et al.* 1992b] Andreas Pommert, Michael Bomans, Ulf Tiede, Karl Heinz Höhne. Simulation studies for quality assurance of 3D-images from computed tomograms. In Andrew E. Todd-Pokropek, Max A. Viergever (Eds.), *Medical Images: Formation, Handling and Evaluation*, volume 98 of *NATO ASI Series F*, 325–332. Springer-Verlag, Berlin, 1992. 6, 20
- [Pommert *et al.* 1994] Andreas Pommert, Rainer Schubert, Martin Riemer, Thomas Schiemann, Ulf Tiede, Karl Heinz Höhne. Symbolic modeling of human anatomy for visualization and simulation. In Richard A. Robb (Ed.), *Visualization in Biomedical Computing 1994, Proc. SPIE 2359*, 412–423, Rochester, MN, 1994. 70
- [Pommert 1999] Andreas Pommert. Untersuchung der Bildqualität für die dreidimensionale Darstellung medizinischer Objekte aus tomographischen Volumendaten. Dissertation, Fachbereich Informatik, Universität Hamburg, in Vorbereitung, 1999. 11, 81
- [Rheinwald 1996] Klaus Rheinwald. Ein interaktiver 3D-Editor zur Modellierung von Blutgefäßen. Diplomarbeit, Fachbereich Informatik, Universität Hamburg, 1996. 75
- [Robb 1996] R. A. Robb. Analyze. World Wide Web: http://www.mayo.edu/bir/BIR_home.html, 1996. 16
- [Sakas 1996] Georgios Sakas. Invivo - interactive visualizer of volume data. World Wide Web: <http://www.igd.fhg.de/www/igd-a7/Projects/Invivo/>, 1996. 16
- [Samet, Webber 1988] Hanan Samet, Robert E. Webber. Hierarchical data structures and algorithms for computer graphics. *IEEE Comput. Graphics & Appl.*, 8(3):48–68, 1988. 49
- [Schiemann *et al.* 1992] Thomas Schiemann, Michael Bomans, Ulf Tiede, Karl Heinz Höhne. Interactive 3D segmentation of MRI volume data. In *Society of Magnetic Resonance in Medicine: Book of Abstracts, Vol. 2*, 4203, Berlin, 1992. 8
- [Schiemann *et al.* 1996] Thomas Schiemann, Jochen Nuthmann, Ulf Tiede, Karl Heinz Höhne. Segmentation of the Visible Human for high quality volume based visualization. In Karl Heinz Höhne, Ron Kikinis (Eds.), *Visualization in Biomedical Computing, Proc. VBC '96*, volume 1131 of *Lecture Notes in Computer Science*, 13–22. Springer-Verlag, Berlin, 1996. 1
- [Schiemann *et al.* 1997] Thomas Schiemann, Ulf Tiede, Karl Heinz Höhne. Segmentation of the Visible Human for high quality volume based visualization. *Med. Image Anal.*, 1(4):263–271, 1997. 28, 59

- [Schiemann, Höhne 1997] Thomas Schiemann, Karl Heinz Höhne. Definition of volume transformations for volume interaction. In James Duncan, Gene Gindi (Eds.), *Information Processing in Medical Imaging, Proc. IPMI '97*, volume 1230 of *Lecture Notes in Computer Science*, 245–258. Springer-Verlag, Berlin, 1997. 66
- [Schiemann 1998] Thomas Schiemann. *Interaktive Verfahren für deformierende Eingriffe an volumenbasierten digitalen Körpermodellen*. Dissertation, Technische Universität Hamburg-Harburg, 1998. 82
- [Schroeder *et al.* 1992] William J. Schroeder, Jonathan A. Zarge, William E. Lorensen. Decimation of triangle meshes. In *Proceedings SIGGRAPH '92, July 26-31*, volume 26, 65–70, 1992. 12
- [Schroeder *et al.* 1998] William J. Schroeder, Kenneth M. Martin, William E. Lorensen (Eds.). *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Prentice Hall, New Jersey, 1998. 16, 24
- [Schubert *et al.* 1992] Rainer Schubert, Michael Bomans, Karl Heinz Höhne, Thomas Schiemann, Ulf Tiede, Werner Lierse. A new method for representing the human anatomy. In C. Roux, Gabor T. Herman, R. Collorec (Eds.), *3D Advanced Image Processing in Medicine, Proc. 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 3–8, Rennes, 1992. 70
- [Schubert *et al.* 1995] Rainer Schubert, Andreas Pommert, Karl Heinz Höhne. Methods for model-based knowledge representation in anatomy. In *Annu. Symp. Comput. Appl. Med. Care, Proc. SCAMC*, 971, New Orleans, LA, 1995. (abstract). 70
- [Spitzer *et al.* 1996] Victor Spitzer, Michael J. Ackerman, Ann L. Scherzinger, David Whitlock. The Visible Human male: A technical report. *J. Am. Med. Inf. Ass.*, 3(2):118–130, 1996. 59
- [Stiehl, Jackèl 1987] H. Siegfried Stiehl, Dietmar Jackèl. On a framework for processing and visualizing spatial images. In Heinz U. Lemke, Michael L. Rhodes, C. Carl Jaffe, Roland Felix (Eds.), *Computer Assisted Radiology, Proc. CAR '87*, 665–670, Berlin, 1987. Springer-Verlag. 15, 16
- [Stiehl 1987] H. Siegfried Stiehl. On spatial image sequence understanding. Habilitation, Fachbereich Informatik, Technische Universität Berlin, 1987. 15, 16, 38
- [Stollnitz *et al.* 1995] E. J. Stollnitz, T. D. DeRose, D. H. Salesin. Wavelets for computer graphics: A primer, part 1. *IEEE Comput. Graphics Appl.*, 15(3):76–83, 1995. 82
- [Tiede *et al.* 1985] Ulf Tiede, Fritz R. P. Boecker, Gerd Witte, Karl Heinz Höhne. Eine neue Heuristik für die 3D-Rekonstruktion medizinischer Bildsequenzen mittels Triangulation. In H. Niemann (Ed.), *Mustererkennung 1985, Proc. 7. DAGM-Symposium*, volume 107 of *Informatik-Fachberichte*, 207–212. Springer-Verlag, Berlin, 1985. 10
- [Tiede *et al.* 1990] Ulf Tiede, Karl Heinz Höhne, Michael Bomans, Andreas Pommert, Martin Riemer, Gunnar Wiebecke. Investigation of medical 3D-rendering algorithms. *IEEE Comput. Graphics Appl.*, 10(2):41–53, 1990. 12, 13, 20, 23

- [Tiede *et al.* 1993] Ulf Tiede, Michael Bomans, Karl Heinz Höhne, Andreas Pommert, Martin Riemer, Thomas Schiemann, Rainer Schubert, Werner Lierse. A computerized three-dimensional atlas of the human skull and brain. *Am. J. Neuroradiology*, 14(3):551–559, 1993. 69
- [Tiede *et al.* 1996] Ulf Tiede, Thomas Schiemann, Karl Heinz Höhne. Visualizing the Visible Human. *IEEE Comput. Graphics Appl.*, 16(1):7–9, 1996. 59
- [Tiede *et al.* 1998] Ulf Tiede, Thomas Schiemann, Karl Heinz Höhne. High quality rendering of attributed volume data. In David Ebert, Hans Hagen, Holly Rushmeier (Eds.), *Proc. IEEE Visualization '98*, 255–262, Los Alamitos, CA, 1998. IEEE Computer Society Press. 20
- [Tönnies, Lemke 1994] Klaus D. Tönnies, Heinz U. Lemke. *3D-Computergraphische Darstellungen*. Number 9.2 in Handbuch der Informatik. R. Oldenbourg Verlag, München, 1994. 16
- [Tönnies 1989] Klaus D. Tönnies. Surface triangulation by linear interpolation in intersecting planes. In Max A. Viergever (Ed.), *Proc. SPIE Vol. 1137, Science and Engineering of Medical Imaging*, 98–105, 1989. 10
- [Udupa 1995] Jayaram K. Udupa. 3D ViewNix 1.2. World Wide Web: <http://www.mipg.upenn.edu/~vnews/index.html>, 1995. 16
- [Urban *et al.* 1997] Markus Urban, Christoph Groden, Thomas Schiemann, Rainer Schubert, Karl Heinz Höhne. A 3D model of the cranial vessels for anatomy teaching and rehearsal of interventions. In Heinz U. Lemke, Michael W. Vannier, Kiyonari Inamura (Eds.), *Computer Assisted Radiology and Surgery, Proc. CAR '97*, volume 1134 of *Excerpta Medica ICS*, 1014–1015. Elsevier, Amsterdam, 1997. 54, 76
- [Šrámek 1994] Miloš Šrámek. Fast surface rendering from raster data by voxel traversal using chessboard distance. In *Proceedings Visualization '94, October 17-21*, 188–195. IEEE Computer Society Press, 1994. 49
- [Watt 1992] Alan and Mark Watt. *Advanced Animation and Rendering Techniques*. Addison Wesley Publ. Comp., Wokingham, 1992. 30, 36
- [Wegener *et al.* 1996] Otto H. Wegener, Regine Fassel, Doris Welger (Eds.). *Grundkurs Computertomographie*. Blackwell Wissenschafts-Verlag, 1996. 1
- [Whitman *et al.* 1994] Scott Whitman, Charles D. Hanson, Thomas W. Crockett. Recent developments in parallel rendering. *IEEE Comput. Graphics & Appl.*, 14(4):21–22, 1994. 81
- [Wiebecke *et al.* 1989] Gunnar Wiebecke, Michael Bomans, Ulf Tiede, K. H. Höhne. 3D-Visualisierung von schwer segmentierbaren tomographischen Volumendaten. In H. Burkhardt, Karl Heinz Höhne, B. Neumann (Eds.), *Mustererkennung 1989, Proc. 11. DAGM-Symposium*, volume 219 of *Informatik-Fachberichte*, 433–439. Springer-Verlag, Berlin, 1989. 13
- [Wienhard *et al.* 1989] K. Wienhard, R. Wagner, W.-D. Heiss. *PET: Grundlagen und Anwendungen der Positronen-Emissions-Tomographie*. Springer-Verlag, Berlin, 1989. 1

- [Wilmer 1993] Frank Wilmer. Reduktion der Oberflächenbeschreibung triangulierter Oberflächen durch Anpassung an die Objektform. Diplomarbeit, Fachbereich Informatik, Universität Hamburg, 1993. 12
- [Wulff 1992] Jürgen Wulff. Optimierung der Parameter für die semitransparente Darstellung medizinischer Objekte aus tomographischen Volumendaten. Diplomarbeit, Fachbereich Informatik, Universität Hamburg, 1992. 13
- [Yagel *et al.* 1992a] Roni Yagel, D. Cohen, Arie Kaufman. Discrete ray tracing. *IEEE Comput. Graphics Appl.*, 12(5):19–28, 1992. 8, 15
- [Yagel *et al.* 1992b] Roni Yagel, Daniel Cohen, Arie Kaufman. Normal estimation in 3d discrete space. *The Visual Computer*, 8(5-6):278–291, 1992. 23
- [Yagel 1992] Roni Yagel. Realistic display of volumes. *Proceedings of Medical Imaging VI: Image Capture, Formatting and Display*, SPIE 1653:470–476, 1992. 8
- [Zuiderveld 1995] Karel Jan Zuiderveld. *Visualization of Multimodality Medical Volume Data using Object-Oriented Methods*. Dissertation, Faculteit Geneeskunde, Universiteit Utrecht, 1995. 37, 49

A Bestimmung der Gradientengewichtung

Für die Berechnung der Oberflächennormale aus dem Grauwertgradienten ist ein Filterkern erforderlich, der alle möglichen Richtungen der Normalen gleichermaßen zulässt. Die Faktoren, mit denen die Voxel bei der Gradientenberechnung gewichtet werden, sollten daher eine Kugel möglichst gut approximieren. Im Gegensatz zum Zucker-Hummel Operator, bei dem die Gewichtung durch den Abstand des Zentralvoxels zur Kugeloberfläche bestimmt wird, soll hier der entsprechende Volumenanteil einer Kugel in einer $3 \times 3 \times 3$ -Umgebung benutzt werden. OBdA mögen die Voxel der Umgebung die Kantenlänge 1 haben. Daraus ergibt sich der Radius der eingeschlossene Kugel als $R = \frac{3}{2}$ (Abbildung 62).

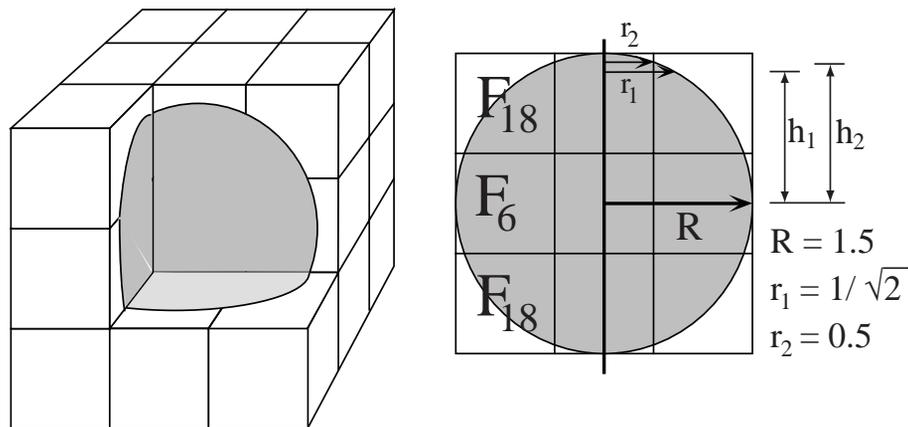


Abbildung 62: Zerlegung einer Kugel in $3 \times 3 \times 3$ isotrope Voxel

Aus Symmetriegründen setzt sich das Volumen V der Kugel aus 4 unterschiedlichen Voxeltypen zusammen:

1. einem Zentralvoxel mit dem Volumen 1,
2. den 6 Nachbarn mit dem Volumen F_6 , die mit dem Zentralvoxel eine gemeinsame Fläche haben,
3. den 12 Nachbarn mit dem Volumen F_{18} , die über eine Kante mit dem Zentralvoxel verbunden sind und
4. den 8 Eckvoxeln mit dem Volumen F_{26} .

Es gilt:

$$\begin{aligned}
 V &= 1 + 6F_6 + 12F_{18} + 8F_{26} \\
 &= V_1 + 2V_2 \quad \text{mit} \quad V_1 = 1 + 4F_6 + 4F_{18} \\
 &= \frac{4}{3}\pi R^3 \quad \text{mit} \quad R = \frac{3}{2} \\
 \Rightarrow V &= \frac{9}{2}\pi
 \end{aligned}$$

Daraus ergibt sich für das Volumen V_2 der beiden äußeren Kugelabschnitte:

$$\begin{aligned}
V_2 &= F_6 + 4F_{18} + 4F_{26} \\
&= \frac{\pi}{3}h^2(3R - h) \quad \text{mit } h = 1 \\
\Rightarrow V_2 &= \frac{7}{6}\pi
\end{aligned}$$

Die Faktoren F_{18} und F_{26} lassen sich daraus in Abhängigkeit von F_6 wie folgt bestimmen:

$$\begin{aligned}
F_{18} &= \frac{1}{4}(V - 2V_2 - 1 - 4F_6) \\
&= \frac{1}{4}\left(\frac{13}{6}\pi - 1\right) - F_6 \\
F_{26} &= \frac{1}{4}(3V_2 - V + 1 + 3F_6) \\
&= \frac{1}{4}(1 - \pi + 3F_6)
\end{aligned}$$

Aus Abbildung 62 ergibt sich:

$$\begin{aligned}
r_1 &= \sqrt{\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2} = \frac{1}{\sqrt{2}} \\
\Rightarrow h_1 &= \sqrt{R^2 - r_1^2} = \frac{\sqrt{7}}{2} \\
r_2 &= \frac{1}{2} \\
\Rightarrow h_2 &= \sqrt{R^2 - r_2^2} = \sqrt{2}
\end{aligned}$$

Zur Lösung muß also das Schnittvolumen der Kugel mit dem Voxel F_6 berechnet werden. Dazu wird F_6 in 3 Komponenten zerlegt:

1. Einer Kugelkappe mit $h = R - h_2$
2. Einem Quader mit Kantenlänge 1 und Höhe $h = h_1 - \frac{1}{2}$
3. Der Schnittmenge des Voxels und der Kugel zwischen h_1 und h_2

Die Berechnung dieser Schnittmenge erfolgt als Integral über die Kreisabschnittsflächen, die die Voxelfläche überdecken (Abbildung 63). Die Fläche eines Kreisabschnitts berechnet sich als:

$$A = \pi r^2 \frac{\alpha}{360} - \frac{s(r-h)}{2}$$

Aus Abbildung 63 ergibt sich:

$$\begin{aligned}
\alpha &= \left(\frac{360}{2\pi}\right) 2 \arccos\left(\frac{1}{2r}\right) \\
s &= \sqrt{4r^2 - 1} \\
h &= r - \frac{1}{2} \\
\Rightarrow A &= r^2 \arccos\left(\frac{1}{2r}\right) - \frac{\sqrt{4r^2 - 1}}{4}
\end{aligned}$$

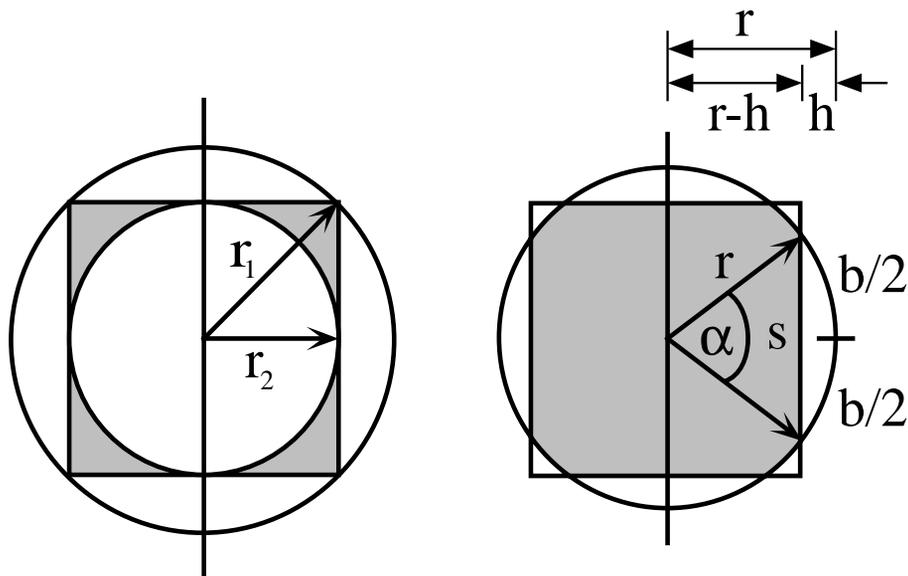


Abbildung 63: Berechnung der Schnittmenge von Kreis und Quadrat

Damit ergibt sich als Schnittmenge eines Kreises mit dem Radius r und einem Quadrat der Kantenlänge 1:

$$\begin{aligned}
 F_r &= \pi r^2 - 4A \\
 &= \pi r^2 + \sqrt{4r^2 - 1} - 4r^2 \arccos\left(\frac{1}{2r}\right)
 \end{aligned}$$

Der Volumenanteil von F_6 berechnet sich dann als:

$$\begin{aligned}
 F_6 &= \underbrace{\frac{\pi}{3} (R - h_2)^2 (2R + h_2)}_{\text{Kappe}} + \underbrace{h_1 - \frac{1}{2}}_{\text{Quader}} + \int_{h=h_1}^{h=h_2} F_r(h) dh \quad \text{mit} \\
 F_r(h) &= \pi x^2 + \sqrt{4x^2 - 1} - 4x^2 \arccos\left(\frac{1}{2x}\right) \quad \text{mit } x = \sqrt{R^2 - h^2}
 \end{aligned}$$

Das Integral wurde numerisch bestimmt. Für die Gewichtungsfaktoren ergeben sich:

$$\begin{aligned}
 F_6 &= 0.9429075152 \\
 F_{18} &= 0.5087885055 \\
 F_{26} &= 0.1717824730
 \end{aligned}$$

B Abbildungstransformation

Die Abbildung eines Volumens der Größe $W \times H \times D$ auf eine Bildebene der Größe $U \times V$ wird durch die Verkettung der nachfolgenden Transformationsschritte mit Hilfe homogener 4×4 Matrizen realisiert:

1. Initialisierung der Transformationsmatrize T auf die Einheitsmatrix:

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Skalierung der X-, Y-, Z-Achse mit dem Kehrwert des Maximums von W , H und D sowie bei nicht isotroper Voxelgröße V mit den jeweiligen Kantenlängen der Voxel, damit mit Einheitswerten, die unabhängig von der Datensatzgröße sind, weiter gerechnet werden kann.

$$S_x = \frac{V_x}{\max(W, H, D)}$$
$$S_y = \frac{V_y}{\max(W, H, D)}$$
$$S_z = \frac{V_z}{\max(W, H, D)}$$
$$T = T \cdot \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3. Verschiebung des Koordinatenursprungs auf den Mittelpunkt des Volumens durch Translation der X-, Y-, Z-Achse um -0.5 .

$$T = T \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0.5 & -0.5 & -0.5 & 1 \end{pmatrix}$$

4. An dieser Stelle erfolgt nun die Drehung der Blickrichtung und die Verschiebung des Betrachterstandortes an die gewünschte Position, wobei die Reihenfolge der Transformationen in der Regel nicht kommutativ und vom Vorstellungsgefühl des Benutzers abhängig ist.

Rotation um die X-Achse mit dem Winkel α

$$T = T \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation um die Y-Achse mit dem Winkel β

$$T = T \cdot \begin{pmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation um die Z-Achse mit dem Winkel γ

$$T = T \cdot \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Translation an die entsprechende Position im Raum

$$T = T \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{pmatrix}$$

5. Für die Erzeugung stereoskopischer Ansichten müssen zwei Abbildungen berechnet werden: Eine für das linke Auge und eine weitere, die nur dem rechten Auge gezeigt wird. Diese Abbildungen unterscheiden sich durch einen Rotationswinkel θ um die Y-Achse, der bei einem Augenabstand von 7 cm und einem Betrachterabstand von 50 cm ca. 8 Grad beträgt. Die Drehachse befindet sich im Horopter Fixationspunkt. Der Horopter ist definiert als der geometrische Ort für alle einfach gesehenen Raumpunkte. In der Regel kann hierfür das Zentrum der darzustellenden Szene benutzt werden, weil die Strahlverfolgungsmethode kein echtes Linsensystem mit Unschärfen außerhalb der eingestellten Brennweite simuliert. Als Verschiebung ergibt sich dann der Abstand des Betrachters vom Mittelpunkt der Szene.

Translation der Z-Achse auf den Fixationspunkt H

Rotation der Y-Achse um $\theta/2$ für die linke Projektion

Rotation der Y-Achse um $-\theta/2$ für die rechte Projektion

Inverse Translation der Z-Achse vom Fixationspunkt H

$$T = T \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & H & 1 \end{pmatrix} \begin{pmatrix} \cos \pm \frac{\theta}{2} & 0 & \sin \pm \frac{\theta}{2} & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \pm \frac{\theta}{2} & 0 & \cos \pm \frac{\theta}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -H & 1 \end{pmatrix}$$

6. Die zentralperspektivische Transformation mit einer fokalen Länge D wird durch Multiplikation der bisherigen Transformation M mit folgender Matrize erreicht [Blinn 1993]:

$$T = T \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1/D \\ 0 & 0 & -D & 0 \end{pmatrix}$$

7. Verschiebung des Koordinatensystems auf den Bildschirmursprung. Dieser liegt bei X- Windows in der linken oberen Ecke des Fensters. Bei anderen Systemen wie OpenGL liegt der Fensterursprung in der linken unteren Ecke, so daß eine zusätzliche Inversion der Y-Koordinate erforderlich ist. Translation der X- und Y-Achse um 0.5 (Die Z-Achse bleibt unverändert).

$$T = T \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.5 & 0.5 & 0 & 0 \end{pmatrix}$$

8. Skalierung auf Fenstergröße der X- und Y-Achse mit dem Maximum von U und V

$$T = T \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \max(U, V) \end{pmatrix}$$

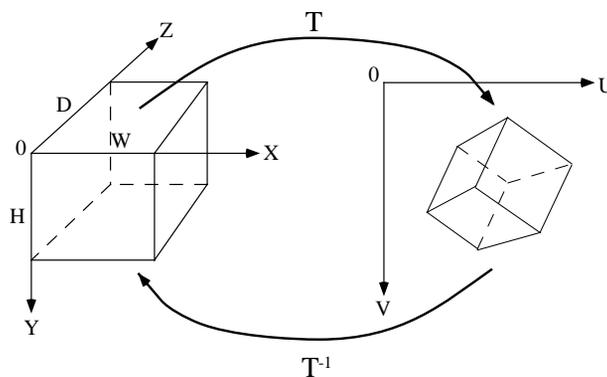


Abbildung 64: Die Transformation T bildet das Volumen im XYZ-Koordinatensystem auf die UV-Bildkoordinaten ab. Für das Ray-Casting wird die inverse Transformation T^{-1} benötigt.

Die resultierende Transformationsmatrize T bildet die Voxel des Datenvolumens in den Bildraum ab (Abbildung 64). Für das Ray-Casting-Verfahren wird jedoch die inverse Transformation T^{-1} benötigt, mit der für jeden Bildpunkt (engl. Pixel) ein Strahl durch das Volumen erzeugt wird. Diese inverse Transformation kann entweder dadurch ermittelt werden, daß die vorgenannten Transformationsschritte in umgekehrter Reihenfolge mit negierten Werten durchgeführt werden, oder durch direkte Berechnung der inversen Matrize. Die inverse Matrize existiert, weil die einzelnen Transformationen invertierbar sind.

C Syntax der Beschreibungssprache

Im folgenden Abschnitt ist die Syntax der Beschreibungssprache für die Visualisierung aus dem VOXEL-MAN-System wiedergegeben. Die einzelnen Komponenten sind in Abschnitt 3.11 beschrieben worden. Der überwiegende Teil der Kommandos wird durch eine graphische Steuerung des Systems (Abbildung 51) abgedeckt. Für Anwendungen, insbesondere für die Erzeugung von Filmen, bei denen es auf präzise numerische Angaben ankommt, wie etwa dem Betrachterstandort, der Blickrichtung oder Transparenzwerten, werden hier die möglichen Parameterwerte aufgeführt. Die Reihenfolge der Sprachelemente ist so angeordnet, wie sie für eine funktionsfähige Anwendung erforderlich sind, d. h., es sind zunächst die Komponenten wie etwa Farbe und Datensätze beschrieben, die später von anderen Komponenten wie Objekte und Regionen benutzt werden.

C.1 General

Resources

Syntax:

```
<resource> - BOUNDINGVOLUMES # see C.15
| CAMERAS # see C.16
| COLORS # see C.3
| COMPOSITIONS # see C.11
| DATASETS # see C.4
| DOMAINS # see C.9
| GRADIENTS # see C.5
| LIGHTSOURCES # see C.18
| OBJECTS # see C.10
| PLANES # see C.13
| RESOLUTIONS # see C.2
| SCENES # see C.12
| SHADINGS # see C.8
| TABLES # see C.6
| TEXTURES # see C.7
```

```
<resource list > - [* | [NO]<resource>] [, <resource list>]
```

SHOW RESOURCES # prints the current resource usage

Initialization

Syntax:

```
INITIALIZE
  [VERSION=<version name>]
  *[MAX<resource>=<number>]*
```

Description:

The initialize command allocates memory for each of the specified resource (C.1) so that up to <number> entries can be defined.

If a version name is specified it is checked against the linked library. A version mismatch indicates changes in the command syntax and/or the behavior. The version is automatically set in the save status (C.1) command.

Initialize is only valid at program startup, or after the resources have been previously destroyed with the release (C.1) command.

Example:

Initialize

```
Version="V3.8.0 - 21-MAR-1996 10:00"
```

```
MaxDomains=20  
MaxBoundingVolumes=10  
MaxCameras=10  
MaxColors=1100  
MaxCompositions=10  
MaxDataSets=100  
MaxGradients=10  
MaxLightSources=8  
MaxPlanes=31  
MaxResolutions=5  
MaxScenes=10  
MaxShadings=150  
MaxTables=200  
MaxTextures=50
```

Release

Syntax:

```
RELEASE [= [(]<resource list>[)]]
```

Description:

Release frees the memory allocated by the specified resources (C.1). This command is used in conjunction with restoring a status (C.1) of a previous session.

CAUTION: Any reference to a released resource is invalid and may crash the program.

Example:

```
release=(*,NoData) # releases all resources except the datasets
```

Exit

Syntax:

```
EXIT
```

Description:

Exit first executes the release (C.1) command and then terminates the program.

Status

Syntax:

```
SAVE STATUS [= <resource list>] FILE=<file name> [APPEND]
```

Restoring a status (C.19):

```
< <file name>
```

Description:

The current definition of the specified resources are stored as an editable text file thus enabling modifications and later recovery of an interrupted session.

Example:

```
save status=(*,NoData) file=ThisSession  
< AnotherSession # executes commands in file: AnotherSession
```

C.2 Resolution

Syntax:

```
DEFINE | MODIFY RESOLUTION <name>
  [<definition>]
  [END [RESOLUTION]]
```

```
SET RESOLUTION <name> CURRENT
```

```
<definition> - [ACCESS <access>]
               [FLAGS=(|BESTID | FASTID|)]
               [NEIGHBOURS
                 [GRAYLEVEL= [6 | 18 | 26]]
                 [Z-GRADIENT= [4 | 8]]]
               [SAMPLING=<0.01 ... >]
<access>      - [FILTER=<method>]
               [GRADIENT=<method>]
               [MAPPING=<method>]
               [SLICE=<method>]
               [THRESHOLD=<method>]
<method>     - B-SPLINE
               | HR-SPLINE [ALPHA=<-0.5 ... 1.0>]
               | LINEAR | NEARESTNEIGHBOUR
               | SHARPEN [WEIGHT=<0.00 ... 0.99>]
```

Example:

```
Define Resolution "VERY LOW"
  Access
    Gradient=NearestNeighbour
    Mapping=NearestNeighbour
    Slice=NearestNeighbour
    Threshold=NearestNeighbour
  Neighbours
    GrayLevel=6
    Z-Gradient=4
  Sampling=4
End Resolution

Define Resolution "VERY HIGH"
  Flags=(BestID)
  Access
    Gradient=Linear
    Mapping=Linear
    Slice=HR-Spline
    Alpha=-1.000000
    Threshold=Linear
  Neighbours
    GrayLevel=26
    Z-Gradient=8
  Sampling=0.1
End Resolution

Set Resolution "STANDARD" current
```

C.3 Color

Syntax:

```
DEFINE | MODIFY COLOR <name>
```

[INTERACTIVE | <definition>]

SET COLOR <option>

<definition> - <model>=(<c1>,<c2>,<c3>)
| NAME=<predefined X-windows color name>
| TABLE=<table name> # (C.6)
<model> - RGB | HLS | HSV | XYZ | LUV | YIQ | CMY | YUV
<option> - COMPRESSMODE=<mode>
| CONVERSION [METHOD=<method>]
| MODEL=<model>
<method> - ORDEREDDITHERED
| FLOYDSTEINBERG [[NO]DITHER] [PRECOMPUTED | [NO]FAST [TEST]]
<mode> - CLAMP # simply truncates values
| CLIP # shift components towards white
| SCALE # linear scale of all components

Examples:

```
set color model=RGB
set color CompressMode=clip
set color conversion method=OrderedDithered
```

```
define color "White" RGB=(1.00, 1.00, 1.00)
define color BACKGROUND name="dark slate blue"
modify color CORNEA RGB=(0.70, 0.70, 1.00)
```

C.4 Dataset

Syntax:

SET DATAFORMAT=[ALPHA | HP | MIPS | SPARC | SGI | VAX] # automatic byte swapping
SET DATAOFFSET=<number of voxels> # number of voxels surrounding the volumes

DEFINE | MODIFY DATASET <name>

[<definition>
[END [DATASET]]]

MODIFY RAY [OFF
| [COMPOSITION=[<composition name> | CURRENT]]
| [OBJECT=[(| <id list> |)]] # see C.14
| [NEW OBJECT [&= | |= | += | -= | =]
| [ERASE | UNERASE | <id>]]
| [DEPTH=<number of voxels>]
| [SIZE=<neighbourhood size>]
| [BACKWARD | FORWARD]
| [IMAGESPACE | OBJECTSPACE]
| [ON]]

UPDATE DATASET <name> HISTOGRAM

<definition> - [TYPE=[VOLUME <vol-spec> |
POLYLINE <line-spec> |
POLYHEDRON <poly-spec> |
IMAGE <image-spec> |
ELLIPSOID <ell-spec>]]

<vol-spec> - [FLAGS=[<flag> | (<flag list>)]]
[FILE=<file name>]

[DATA=[8BIT | 16BIT | 32BIT | FLOAT | RGBA]]
 [COMPONENTS=<number of elements per data item>]
 [DX=<width>] [DY=<height>] [DZ=<depth>]
 [VOXELSIZE=(< X_{scale} >, < Y_{scale} >, < Z_{scale} >)]
 [NOWINDOW | WINDOW=(<data window>)]
 [MINVALUE=<minimum value in data>]
 [MAXVALUE=<maximum value in data>]
 [HISTOGRAM=<table name>]
 [MODALITY=[LABEL | CT[A] | MR[A]I | PET |
 SPECT | US | CRYOTOM | DOSE | VOI |
 FILTER | COMPOSITION]]

<flag list> - <flag> [, <flag list>]
 <flag> - [NO]EXTERNAL | [NO]TRANSFORMED | [NO]MODIFIED |
 [NO]MEMORY | [NO]SHAREDMEMORY | [NO]CRYPTIC |
 [NO]COMPRESSED

<data window> - [XO=< X_{min} >] [,] [YO=< Y_{min} >] [,] [ZO=< Z_{min} >] [,]
 [X1=< X_{max} >] [,] [Y1=< Y_{max} >] [,] [Z0=< Z_{max} >]

<image-spec> - FILE=<file name>
 [FORMAT=[TIFF | IMDM [FRAME=<frame number>]]]

<line-spec> - DATA=(<coord list length>,(<coord list>))
 [FLAGS=([NO]RENDERGL)]

<coord list> - <coordinate 1>, ..., <coordinate n>

<coordinate> - (< X >, < Y >, < Z >)

<poly-spec> - FILE=<file name> # special format
 [FLAGS=([NO]RENDERGL)]

<ell-spec> - [COMPONENTS=<number of elements per data item>]
 [CENTER=(<value list>)]
 [RADIUS=(<value list>)]
 [BASIS=(<value matrix>)]

<value list> - <value 1>, ..., <value n> # n = number of components

<value matrix>- (<value list 1>, ..., <value list n>)

MOVE DATASET <name> [<model view>]

<model view> - [DIRECTION=(<view dir>)]

[POSITION=(<view pos>)]

[<model view>]

[ABSOLUTE | RELATIVE]

<view dir> - [X | Y | Z]=<degree> [, <view dir>]

<view pos> - [X | Y | Z]=<coord> [, <view pos>]

REMOVE DATASET [* | <name>]

SAVE DATASET [* | <name>] [<save option>]

<save option> - [DESCRIPTION]

[OVERRIDE]

FILE=<description file name> [APPEND]

[DATA=<data file name>]

Example:

```
Define DataSet "VOXEL-MAN/Logo"
  Type=Image
```

```

Format=TIFF
File=/demo/atlas/V2/VOXEL-MAN.Logo.tiff
End DataSet

Define DataSet "MR"
Type=Volume
Modality=MR
Flags=(shared)
File=/demo/atlas/brain/volumes/mr.ima
Data=8bit
DX=150 DY=200 DZ=192 DT=1
Window=(X0=0, Y0=0, Z0=0, T0=0,
        X1=149, Y1=199, Z1=191, T1=0)
VoxelSize=(1.08, 1.08, 1.08)
MinValue=0
MaxValue=255
End DataSet

Define DataSet "lens"
Type=Ellipsoid
components=3
Center=( 97.671310, 160.794022, 152.017578 )
Radius=( 8.975043, 17.650211, 101.617500 )
Basis=(( 0.069722, -0.697610, 0.713077 ),
       ( 0.865482, -0.313158, -0.390989 ),
       ( 0.496064, 0.644416, 0.581935 ))
End DataSet

Define DataSet "sulcus lateralis dext"
Type=Polyline
Data=(9,
      (34.0, 71.0, 117.0),
      (35.0, 80.0, 115.0),
      (32.0, 83.0, 119.0),
      (27.0, 87.0, 121.0),
      (30.0, 91.0, 121.0),
      (19.0, 104.0, 121.0),
      (19.0, 112.0, 123.0),
      (19.0, 118.0, 125.0),
      (19.0, 120.0, 126.0))
End DataSet

Save DataSet "MR" File=MR.par Data=MR.volume

```

C.5 Gradient

Syntax:

```

DEFINE | MODIFY GRADIENT <gradient name>
[METHOD=
 [ADPATIVE
 [NEIGHBOURS
 [GRAYLEVEL=[ 6 | 18 | 26 ]]
 [Z-GRADIENT=[ 4 | 8 ]]
 [MAXIMUM=<maximum z-buffer difference>]
 [SCALE=<z-buffer difference scaling factor>]
 [THRESHOLD=<switch between graylevel and z-gradient>]] |

```

```
GRAYLEVELGRADIENT
  [NEIGHBOURS=[ 6 | 18 | 26 ]] |
```

```
Z-GRADIENT
  [NEIGHBOURS=[ 4 | 8 ]]
  [MAXIMUM=<maximum z-buffer difference>]
  [SCALE=<z-buffer difference scaling factor>]]
```

```
[[NO]ADAPTIVE]
[OBJECTSPACE | IMAGESPACE]
[WIDTH=<gradient width>]
[END [GRADIENT]]
```

NOTE: The values for the neighbours will be overwritten by the corresponding values of the resolution (C.2) used.

Example:

```
Define Gradient "GLG"
  Method=GrayLevelGradient
  Adaptive
  ImageSpace
  Width=1.00
End Gradient
```

```
Define Gradient "ADAPTIVE"
  Method=Adaptive
#   Neighbours
#   GrayLevel=26
#   Z-Gradient=8
  Scale=1.0
  Threshold=0.02
  Adaptive
  ImageSpace
  Width=1.00
End Gradient
```

C.6 Table

Syntax:

```
DEFINE [| MODIFY TABLE <name>
  [<definition>]
[END [TABLE]]
```

```
REMOVE | SAVE TABLE <name>
```

```
SHOW TABLE <name>
  [CAMERA=[<camera name> | CURRENT]]
  [X=<X pos on screen>] [Y=<Y pos on screen>]
  [WIDTH=<width in pixels>] [HEIGHT=<height in pixels>]
  [ORIENTATION=[LEFTRIGHT | RIGHTLEFT | TOPBOTTOM | BOTTOMTOP]]
```

```
<defintion> - [TYPE=[BOOLEAN | CARDINAL | COLOR | REAL | WORD]]
  [LENGTH=<number of entries>]
  [FILE=<file name> |
  VALUES=(<value list>)]
```

```
<value list> - <value> [, <value list>]
```

```

<value> - <index 1> [- <index 2>] = (<c1>,<c2>,<c3>) # color tables
| <index 1> [- <index 2>] = ON | OFF # boolean tables
| <index 1> [= <value 1>] [- <index 2> [= <value 2>]]

```

Example:

```

Define Table CORNEA
  Type=boolean
  Length=256
  Values=(0-19=off, 20-59=on, 60-255=off)
End Table

Define Table "MR-Head-Reference-Index"
  Type=word
  Length=184632
  File=/demo/atlas/brain/volumes/MR-Head-Reference.index.volatile
End Table

Define Table PET
  Type=color
  Length=256
  File=pet.lut
End Table

Modify Table OPACITY
  Values=(0=0.10-255=0.10)
End Table

```

C.7 Texture

Syntax:

```

DEFINE | MODIFY <texture name>
  [COLOR <texture>]
  [NORMAL <texture>]
[END [TEXTURE]]

```

```

<texture> - [METHOD=[NONE | <gradient> | <noise>
              <picture>| <volume>]]

```

```

<gradient> - GRADIENTMAPPING
  [DATA=<volume data name>] # see C.4
  [GRADIENT=<gradient name>] # see C.5
  [TABLE=<gradient mapping table>] # see C.6
  [WEIGHT=<texture weight [0 ... 1]>]

```

```

<noise> - NOISEMAPPING
  [DATA=<volume data name>] # see C.4
  [FRACTAL=<fractal>]
  [FREQUENCY=<frequency>]
  [ITERATION=<number of iterations>]
  [LAMBDA=<lambda>]
  [MODE=[AGATE | BOZO | BUMP1 | BUMP2 |
        DENT1 | DENT2 | GRANITE1 | HEXAGON | LEOPARD |
        MANDEL | MARBLE1 | MARBLE2 | NOISE1 | NOISE2]]
  [NOISE=<noise volume name>] # see C.4
  [OMEGA=<omega>]
  [RESOLUTION=<resolution>]
  [TABLE=<color table name>] # see C.6

```

[WEIGHT=<texture weight [0 . . . 1]>]

<picture> - PICTUREMAPPING

[CAMERA=<camera name>] # see C.16
[IMAGE=<image data name>] # see C.4
[CENTER=<center value>]
[INVERSE] [SWAP] [ZERO=[NO | YES]]
[INTERPOLATION=[NONE | BILINEAR | NORMDIST]]
[MODE=[CYLINDERMAP | SPHEREMAP |
[[SLIDERMAP | WALLMAP
[FRONT | BACK | LEFT | RIGHT | BOTTOM | TOP]]
[<model view>]

[WEIGHT=<texture weight [0 . . . 1]>]
[XREPEAT=<number of repetitions>]
[XROTATE=<angle [0 . . . 360]>]
[YREPEAT=<number of repetitions>]
[YROTATE=<angle [0 . . . 360]>]

<volume> - VOLUMEMAPPING

[CONTRAST
[<low> - <high> |
SLOPE=<m> INTERSECTION=<n>]] # $f(x) = mx + n$
[DATA=<volume data name>] # see C.4
[TABLE=<color table name>] # see C.6
[OFFSET=<offset from position along surface normal>]
[WEIGHT=<texture weight [0 . . . 1]>]

<model view> - [DIRECTION=(<view dir>)]

[POSITION=(<view pos>)]
[SCALE=(<scale>)]
[ZOOM=<zoom factor>]
[<model view>]

[ABSOLUTE | RELATIVE]

<view dir> - [X | Y | Z]=<degree> [, <view dir>]

<view pos> - [X | Y | Z]=<coord> [, <view pos>]

<scale> - [X | Y | Z]=<scale factor> [, <scale>]

SAVE TEXTURE [* | <texture name>] FILE=<file name> [APPEND]

Example:

```
Define Texture "VHP"
```

```
Color
```

```
Method=VolumeMapping
```

```
Data="VHP"
```

```
Offset=1.5
```

```
Weight=1.0
```

```
End Texture
```

```
Define Texture "PET-PLANE"
```

```
Color
```

```
Method=VolumeMapping
```

```
Data="PET Patient"
```

```
Table="PET"
```

```
Weight=0.800000
```

```
Normal
```

```
Method=GradientMapping
```

```

Gradient="DEFAULT"
Table="PLANE"
Weight=0.250000
End Texture

Modify Texture "FRONT"
Normal
Method=PictureMapping
CAMERA="Camera 1"
IMAGE="VOXEL-MAN/Logo"
WEIGHT=0.075
ZERO=NO
INTERPOLATION=BILINEAR
XREPEAT=1
YREPEAT=1
MODE=WALLMAP
Front
Rotation=(X=180.0)
Scale=(X=1.0,Y=1.0,Z=1.0)
Zoom=1.5 abs
End Texture

```

C.8 Shading

Syntax:

```

DEFINE | MODIFY SHADING <shading name>
[CONTRAST
  [<low> - <high> |
   SLOPE=<m> INTERSECTION=<n>]] #  $f(x) = mx + n$ 
[COLOR=<color name>]
[TEXTURE=<texture name>]
[METHOD=
  [ DISTANCE
  | LINE
   WIDTH=<line width>
  | INTEGRAL
   FORWARD=<number of voxels in forward dir>
   BACKWARD=<number of voxels in backward dir>
  | MIP <radio>
   [DISTANCE=<distance attenuation>]
  | PHONG <surface>
  | VOLUME <surface>
   [OPACITY=<gray level opacity table name>]
   [GRADIENT=<gradient weight table name>]
  | XRAY <radio>]
[END [SHADING]]

<surface> - [SURFACENORMAL=<gradient name>]
           - [AMBIENT=<ambient reflection>]
           - [DISTANCE=<distance attenuation>]
           - [DIFFUSE FRACTION=<diffuse reflection>]
           - [SPECULAR FRACTION=<specular reflection>]
           - [SHARPNESS=<size of highlight>]
           - [TRANSPARENCY=<transparency factor [0 .. 1]>]
           - [ATTENUATION=<transparency attenuation>]

<radio> - [SCALE=<contrast multiplier>]

```

- [TRANSPARENCY=<transparency factor [0 . . . 1]>]
- [WEIGHT=<gray level weight table>]

SAVE SHADING [* | <shading name>] FILE=<file name> [APPEND]

Example:

```
Define Shading "MIP"
```

```
  Color="MIP"
  Contrast 0-255
  Method=MIP
  Scale=1.00
  Transparency=0.00
  Weight="MIP"
  Distance=0.00
```

```
End Shading
```

```
Define Shading "STANDARD"
```

```
  Color="DEFAULT"
  Contrast 0-200
  Method=Phong
  SurfaceNormal="GLG"
  Ambient=0.05
  Distance=0.30
  Diffuse Fraction=0.70
  Specular Fraction=0.25
  Sharpness=10
  Transparency=0.00
  Attenuation=0.00
```

```
End Shading
```

```
Define Shading "SULCUS"
```

```
  Color="DEFAULT"
  Contrast 0-64
  Method=Distance
```

```
End Shading
```

```
Define Shading "VOLUME"
```

```
  Color="DEFAULT"
  Contrast 0-200
  Method=Phong
  SurfaceNormal="GLG"
  Ambient=0.05
  Distance=0.30
  Diffuse Fraction=0.70
  Specular Fraction=0.25
  Sharpness=10
  Transparency=0.00
  Attenuation=0.00
  Opacity="OPACITY"
  Gradient="GRADIENT"
```

```
End Shading
```

C.9 Domain

Syntax:

```
DEFINE | MODIFY DOMAIN [<domain name> | CURRENT]
  [DATA=<volume data name>]
```

```
[FLAGS=[([NO]VOLATILE[])]
[MaxObjects=<max number of objects in domain>]
[END [DOMAIN]]
```

```
SET DOMAIN <domain name> CURRENT
```

Description:

A domain is a container for a label volume together with corresponding object descriptions (C.10) for each label.

Note: If the domain is used in a multiple domain composition (C.11) the volatile flag must be set if the labels in the volume will be changed during runtime. For single domain compositions this flag is irrelevant.

Example:

```
Define Domain "STRUCTURE"
  Data="STRUCTURE"
  MaxObjects=256
End Domain
```

```
Define Domain "VOLATILE"
  Data="VOLATILE"
  Flags=(volatile)
  MaxObjects=2
End Domain
```

```
set domain "STRUCTURE" current
```

C.10 Object

Syntax:

```
DEFINE OBJECT [<domain>] <name>
  UID=<uid list>
  <object>
[END [OBJECT]]
```

```
MODIFY OBJECT [<domain>] <object list>
  <object>
[END [OBJECT]]
```

```
<object> - [COLOR=<color name>]
  [DATA
    [DEFINITION=<volume data name>]
    [GRADIENT=<volume data name>]
    [MAPPING=<volume data name>]
    [RADIOLOGICAL=<volume data name>]]
  [FLAGS=[([NO]SURFACE | [IN]VISIBLE)]
  [EXTENT=[DEFAULT | <box>]]
  [NEIGHBOURS=[<uid list> | <name list>]]
  [SHADING
    [SURFACE=<surface shading name>]
    [MIP=<mip shading name>]
    [VOLUME=<volume shading name>]
    [XRAY=<xray shading name>]
    [PLANE
      [ANATOMICAL=<shading name>]
      [RADIOLOGICAL=<shading name>]]]
  [TEXTURE=<texture name>]
  [THRESHOLD=[ELLIPSOID=<ellipsoid data name> | # see C.4
```

<boolean table name>]]

see C.6

CHECK THRESHOLD [<domain>]

SAVE OBJECT [<domain>] <object list>
[EXTENTS | DESCRIPTION]
FILE=<file name> [APPEND]

UPDATE OBJECT [<domain>] <object list> EXTENT

<domain> - DOMAIN=[<domain name> | CURRENT]

<object list> - [* | CURRENT |
[UID=[<uid> | (<uid list>)] |
[<object name> | (<name list>)]]]

<uid list> - <uid> [- <uid>] [, <uid list>]

<name list> - <object name> [, <name list>]

```
define object "soft tissue"
  uid=(2)
  color="SKIN"
  data
    definition="MR"
  shading
    surface="STANDARD"
  threshold="SKIN"
  extent=((0,2,4),(148,194,190))
  neighbours=(0-1, 3-5, 7-9, 17-19, 24, 28-29, 40, 43-45, 80-83, 88-89,
            94, 110-120, 138-142, 146-149, 153-154)
end object
```

```
define object "left cornea"
  uid=(113)
  color="CORNEA"
  data
    definition="MR"
  shading
    surface="STANDARD"
  threshold="CORNEA"
  extent=((100,39,96),(113,45,106))
  neighbours=(1-2, 110-111)
end object
```

C.11 Composition

Syntax:

```
DEFINE | MODIFY COMPOSITION [<composition name> | CURRENT]
  [CASE=(|<KB case list>|)] # knowledgebase concept
  [DOMAIN=(|<domain list>|)]
  [[DATA=<16 bit volume name>]
  [INDEX=<id index table name>
  INVERSE=<inverse id index table name>]]
  [WINDOW=[NONE | (<window>)]]
[END [COMPSOSITION]]
```

<KB case list> - <KB case name> [, <KB case list>]

<domain list> - <domain name> [, <domain list>]

<window> - [X0=< X₀ >] [Y0=< Y₀ >] [Z0=< Z₀ >]
[X1=< X₁ >] [Y1=< Y₁ >] [Z1=< Z₁ >]

```
SET COMPOSITION <composition name> CURRENT
UPDATE COMPOSITION [<composition name> | CURRENT]
```

Description:

A composition provides the mapping of object labels from multiple domains to unique identifiers. If only a single domain is used the identifiers are identical with the object labels.

Example:

```
Define Composition "MR-Head-Reference"
  Domain=( "STRUCTURE" , "FUNCTION" , "SUPPLY" , "VOLATILE" )
  Case=( "MR-Head-Reference" )
  Data="MR-Head-Reference"
  Index="MR-Head-Reference-Index"
  Inverse="MR-Head-Reference-Inverse"
End Composition

set composition "MR-Head-Reference" current
```

How to create a multiple domain composition volume for faster execution:

1. Define the composition with all wanted domains:

```
DEFINE COMPOSITION "SpeedUp"
  DOMAINS=( "domain 1" , ... , "domain n" )
  DATA="SpeedUp"
END
```

A dataset definition for “SpeedUp must not exist before. The object id mapping tables and the composition volume “SpeedUp will be generated.

2. Save the composition volume and the mapping tables:

```
SAVE DATASET "SpeedUp" FILE=SpeedUp.par DATA=SpeedUp.volume
SAVE TABLE "SpeedUp-INDEX"
SAVE TABLE "SpeedUp-INVERSE"
```

3. Modify the parameter definitions as follows:

- (a) If the data volumes of the domains specified in the composition definition are not used elsewhere they can be marked as “nomemory, because they will not be accessed. However, the description must still be available.

```
MODIFY DATA "domain 1" FLAGS=NOMEMORY
...
MODIFY DATA "domain n" FLAGS=NOMEMORY
```

- (b) Load the definition of the mapping tables and the composition volume:

```
DEFINE TABLE "SpeedUp-INDEX" ...
DEFINE TABLE "SpeedUp-INVERSE" ...
< SpeedUp.par
```

- (c) Modify the composition definition:

```
DEFINE COMPOSITION "SpeedUp"
  DOMAINS=( "domain 1" , ... , "domain n" )
  DATA="SpeedUp"
  INDEX="SpeedUp-INDEX"
  INVERSE="SpeedUp-INVERSE"
END
```

- (d) Restart the visualizer

C.12 Scene

Syntax:

```
DEFINE | MODIFY SCENE [<scene name> | CURRENT]
  [COMPOSITION=<composition name>] # see C.11
  [OFFCUT
    [BOX=[DEFAULT | <box>]]
    [MODE=[ANATOMICAL | RADIOLOGICAL]]
    [SHADING [ANATOMICAL=<shading name>]
      [RADIOLOGICAL=<shading name>]]
    [OFF | ON]]
  [END [SCENE]]
```

<box> - ((< X_{min} >, < Y_{min} >, < Z_{min} >),(< X_{max} >, < Y_{max} >, < Z_{max} >))

```
MOVE SCENE [<scene name> | CURRENT] <model view>
<model view>- [DIRECTION=(<view dir>)]
  [POSITION=(<view pos>)]
  [<model view>]
  [ABSOLUTE | RELATIVE]
<view dir> - [X | Y | Z]=<degree> [, <view dir>]
<view pos> - [X | Y | Z]=<coord> [, <view pos>]
```

```
SET SCENE <scene name> CURRENT
UPDATE SCENE [<scene name> | CURRENT]
```

Example:

```
Define Scene "MR-Head-Reference"
  Composition="MR-Head-Reference"
  OffCut
    box=( (0,0,6) , (149,199,191) )
    mode=anatomical
    shading
      radiological="OFFCUT"
    off
End Scene
```

```
Move Scene "MR-Head-Reference"
  Direction=(X=0.00,Y=0.00,Z=0.00)
  Position=(X=0.00,Y=0.00,Z=0.00)
  absolute
```

```
Set Scene "MR-Head-Reference" current
```

C.13 Planes

Syntax:

```
SET PLANE <plane spec>
  [BACKGROUND=<color name>] # see C.3
  [MODE=[ANATOMICAL | RADIOLOGICAL]]
  [NAME=<plane name>]
  [SHADING [MARKER=<line shading name>]
    [ANATOMICAL=<shading name>]
    [RADIOLOGICAL=<shading name>]]
  [SLICE=<slice spec> ID=<id spec> [OFF | ON]]
  [[NO]MARK]
  [OFF | ON]
```

[CURRENT]

<plane spec> - [CAMERA=[<camera name> | CURRENT]
 [X=<screen pos> Y=<screen pos> | <plane id>]
 | [SCENE=[<scene name> | CURRENT]] <plane id>]

<plane id> - [* | <plane number> | [<plane name> | CURRENT]]

<slice spec> - [* | (<area code i>, <area code j>)] # $i \neq j$

<id spec> - see C.14

<geometry> - P1=(X_1 >, Y_1 >, Z_1 >)
 P2=(X_2 >, Y_2 >, Z_2 >)
 P3=(X_3 >, Y_3 >, Z_3 >)
 | EQUATION=(A >, B >, C >, D >)
 | SHIFT=<voxels along plane normal>
 | ORTHOGONAL
 | CLEAR

MARK PLANE [CAMERA=[<camera name> | CURRENT] |
 SCENE=[<scene name> | CURRENT]]
 <plane id>

SHOW PLANE [SCENE=[<scene name> | CURRENT]]
 [<plane number> | [<plane name> | CURRENT]]
 [[NO]AUTOMATIC]
 [DATA=<volume data name>]
 [X=<screen pos>] [Y=<screen pos>] [WIDTH=<width>] [HEIGHT=<height>]
 [OFF]

Example:

```
set plane 1
  background="PLANEBACKGROUND"
  mode=radiological
  shading
    marker="MARK"
    anatomical="ANATOM"
    radiological="RADIO"
  P1=(-35.0,103.0,94.151321) P2=(191.0,103.0,94.151321) P3=(78.0,-
10.0,94.151321)
  slice=(1,4)
    id=* on
  slice=(2,3)
    id=* on
    id=(0-17, 31, 83, 85) off
  mark on

set plane 4 name="Stereotaxy" current

show plane 3 data="PET"
```

C.14 Areas

SET AREA <area spec>
 [ID=($\{$ <id list> $\}$)]
 [COLOR=<color name>)]

```

[FLAGS=([IN]ACTIVE | [NO]SURFACE | [IN]VISIBLE |
        NOBACKFACE | [NO]DISPLAYINFRONT)]
[SPECIFICATION=[DEFAULT | <single object uid>]]
[TRANSPARENCY=<transparency [0 ··· 1]]>
[MODE=[MIP | SURFACE | VOLUME | XRAY]]
[NAME=<area name>]
[SHADING
  [MIP=<mip shading name>]
  [SURFACE=<surface shading name>]
  [VOLUME=<volume shading name>]
  [XRAY=<xray shading name>]

```

```

<area spec> - *
             | CAMERA=[<camera name> | CURRENT]
             |   X=<pixel> Y=<pixel> [Z=<pixel>]
             | SCENE=[<scene name> | CURRENT]
             |   X=<voxel> Y=<voxel> Z=<voxel>
             | <area code> | <area name>

```

<pixel> - specification in camera coordinates

<voxel> - specification in volume coordinates

```

<id list> - [([NOT] <id spec> [[AND | OR | XOR] <id list>]]]
           | [INFRONTOF <id list> [CAMERA=[<camera name> | CURRENT]]]

```

```

<id spec> - [NEIGHBOURS(] <domain name>=(<object list>) ]]

```

```

<object list> - [* | CURRENT |
                [UID=[<uid> | (<uid list>)] |
                [<object name> | (<name list>)]]

```

```

<uid list> - <uid> [- <uid>] [, <uid list>]

```

```

<name list> - <object name> [, <name list>]

```

```

CLEAR AREAS [SCENE=[<scene name> | CURRENT]]

```

Example:

```

set area 0
  mode=surface
  shading
    mip="MIP"
    surface="SURFACE"
    volume="VOLUME"
    xray="XRAY"
  id=* color=none flags=active transparency=0.00 mode=surface

```

```

set area 1
  id=("STRUCTURE"=("soft tissue", 17, 88))
  color="PAINT_2"

```

```

set area 3
  id=(4)
  transparency=0.50

```

C.15 Bounding Volume

Syntax:

```

SET BOUNDINGVOLUME [SCENE=[<scene name> | CURRENT]] <name>
[BOX=[DEFAULT | <box>] | CONTOUR=<file name>]
[EXCLUDED OBJECTS=<id list>]

```

[COMMAND=<command>]
[CURRENT]

<id list> - see C.14

<box> - ((< X_{min} >, < Y_{min} >, < Z_{min} >),
(< X_{max} >, < Y_{max} >, < Z_{max} >))

Example:

```
set BoundingBoxVolume "BOX"
  box=( (3,3,3), (152,202,194) )

set BoundingBoxVolume "BRAIN"
  contour=/demo/atlas/brain/volumes/brain.bv
  excluded objects=("STRUCTURE"=("soft tissue", "bone"))
```

C.16 Camera

Definition

Syntax:

```
DEFINE | MODIFY CAMERA [<camera name> | CURRENT]
  [TYPE=[3D-VIEW <3D camera> | 3D-SLICE]]
  [FLAGS=( [[NO]AUTOCURRENT] [[NO]AUTOUPDATE] [[UN]MAPPED) )]
  [X=<screen pos>] [Y=<screen pos>] [WIDTH=<width>] [HEIGHT=<height>]
  [RESOLUTION=[<resolution name> | CURRENT]]
[END [CAMERA]]
```

<3D camera> - [PROJECTION=[PERSPECTIVE | ORTHOGONAL]]
[SCENE=[<scene name> | CURRENT]]
[STEREO=[RED/GREEN | INTERLACED | MIT | HP]
| NOSTEREO]

SET SHADOW [CAMERA=[<camera name> | CURRENT]] [ON | OFF]

Example:

```
define camera "Camera 1"
  type=3D-view
  flags=(AutoCurrent)
  nostereo
  scene="MR-Head-Reference"
  x=450 y=425 width=256 height=256
  resolution="VERY HIGH"
end camera
```

Move / Update

Syntax: MOVE CAMERA [<camera name> | CURRENT] # 3D cameras only
[MODE=[SCENE <model view> |
VIEWER <model view> |
WINDOW <wind view>]]

[FOCUS=<focal length>]
[ZOOM=<zoom factor>]
[ABSOLUTE | RELATIVE (default)]
[ANGLE=<stereo angle>]

```

<model view> - [DIRECTION=(<view dir>)]
               [POSITION=(<view pos>)]
               [<model view>]
               [ABSOLUTE | RELATIVE]
<view dir>   - [X | Y | Z]=<degree> [, <view dir>]
<view pos>   - [X | Y | Z]=<coord> [, <view pos>]

<wind view>  - [POSITION=(<wind pos>)]
<wind pos>   - [X | Y]=<coord> [, <wind pos>]

MOVE CAMERA [<camera name> | CURRENT]    # Orthogonal projection
           [FRONT | BACK | LEFT | RIGHT | TOP | BOTTOM]
           [UPSIDEDOWN]
           [ZOOM=<1 ... n>]

UPDATE CAMERA [<camera name> | CURRENT]
            [MODE=[BACKGROUND | PAINT | SHADING | REMOVE |
                  ADD | RESOLUTION | VIEWPORT | RESIZE]]
            [TOTAL]

```

Example:

```

move camera "Camera 1"
  mode=scene
    direction=(x=-107.229637,y=-0.300011,z=-31.473663)
    position=(x=0.0,y=0.0,z=0.0)
    absolute
  mode=viewer
    direction=(x=83.834061,y=22.007130,z=0.667014)
    position=(x=0.307851,y=-0.234024,z=0.228969)
    focus=0.7
    absolute
  mode>window
    position=(x=0.0,y=0.0)
    zoom=1.0
    absolute

update camera current

```

Save / Show

Syntax:

```

SAVE IMAGE [[ON] CAMERA=[<camera name> | CURRENT]]
          [FORMAT=[TIFF | QUICKTIME | IMDM]]
          FILE=<file name> [APPEND] [NOSTATUS]

```

```

SHOW MOVIE FILE=<QuickTime file name>

```

Example:

```

save image file=PrettyPicture

save image on camera "Stereo Camera"
  format=QuickTime file=MyMovie append nostatus

```

Miscellaneous

Syntax:

```

REMOVE CAMERA [<camera name> | CURRENT]

```

```

SET CAMERA [<camera name> | CURRENT]
  [BUFFER=[LEFT | RIGHT]]
  | [CURRENT [[NO]PERMANENT]]
  | [DUMP [OFF |
    [[TRAJECTORY [COMMAND=<command>]] |
    [ACTION=[* | OBJECT | LIGHT | RAY]]]
    [FILE=<file name> [APPEND]] ON]]
  | [KEYFRAME]
  | [[UN]MAPPED]

SHOW CAMERA [<camera name> | CURRENT]
  [PANORAMA=(<horizontal [0..360]>,<vertical [0..180]>)]
  | [AREACODE | BACKINGSTORE | BACKLIMIT | FRONTLIMIT |
  GRADIENT | OBJECTS | RAYLENGTH | RAYOFFSETS |
  TESTIMAGE | UID-MISMATCH | Z-BUFFER [GL] |
  Z-GRADIENTIMAGE]

```

C.17 Background

Syntax:

```

SET BACKGROUND [CAMERA=[* | <camera name> | CURRENT]]
  [COLOR=<color name>]
  [FIXED <model view> | NOFIXED]
  [BOX <box spec> | NOBOX]
  [WALL <wall spec> | NOWALL]
  [NONE]

<box spec> - [BOX=[DEFAULT | <box>]
  [SHADING=<line shading>]

<wall spec> - [WALL=[DEFAULT | <box>]
  [[BACK | FRONT] SHADING *[[<side>]=<surface shading>]*]
  [SIDES=(*[NO]<side> [,]*)]

<box> - ((< Xmin > , < Ymin > , < Zmin > ), (< Xmax > , < Ymax > , < Zmax > ))

<side> - [LEFT | RIGHT | TOP | BOTTOM | FRONT | BACK]

<model view> - [DIRECTION=(<view dir>)]
  [POSITION=(<view pos>)]
  [<model view>]
  [ZOOM=<zoom factor>]
  [ABSOLUTE | RELATIVE]

<view dir> - [X | Y | Z]=<degree> [, <view dir>]
<view pos> - [X | Y | Z]=<coord> [, <view pos>]

```

Example:

```

set background camera="Camera 1"
  color="BACKGROUND"
  box
    box=( ( 3 , 3 , 3 ) , ( 152 , 202 , 194 ) )
    shading="BOX"
  wall
    wall=( ( 3 , 3 , 3 ) , ( 152 , 202 , 194 ) )
  back
    shading
      left="LEFT"
      right="RIGHT"

```

```

    top="TOP"
    bottom="BOTTOM"
    front="FRONT"
    back="BACK"
    sides=(left,right,top,bottom,front,back)
nobox wall
nofixed

```

C.18 Lightsource

Syntax:

```

SET LIGHTSOURCE [CAMERA=[* | <camera name> | CURRENT]]
    [<ls number> | <NAME=<ls name>]
    [TYPE=[AMBIENT | DIRECTIONAL | POINT]]
    [COLOR=<color name>]
    [INTENSITY=<value [0. . . ]>]
    [ON | OFF]
    [<direc spec> [<point spec>]]

```

```

<direc spec> - [C1=<c1> C2=<c2> C3=<c3>]          # directional light
    [X=<vertical angle> Y=<horizontal angle>]
    [SHADOW=<shadow [0. . . 1]]
    [[NO]FIXED]

```

```

<point spec> - [POSITION=(<x>,<y>,<z>)] # point light
    [ANGLE=<width of spot> [0. . . 180]>]
    [SPOT=<sharpness> [0. . . ]>]

```

Example:

```

Set Light Camera="Camera 1" 1
    Type=ambient
    Intensity=1.00
    Color="AMBIENT"
    on

```

```

Set Light Camera="Camera 1" 3
    Type=point
    Intensity=0.50
    Color="SPOT_1"
    C1=0.00 C2=0.50 C3=0.10
    X=10.000000 Y=10.000001
    position=(-0.250000, 0.150000, -0.950000)
    angle=70.00 spot=2.00
    Shadow=0.50
    on

```

C.19 User Interface Command Language (UICL)

Execute a command language parameter file:

```
< <file name> # Typein "<" followed by a file name
```

Execute a shell script with stdout piped into the command interpreter:

```
<< ["<shell command> [<parameter 1> . . . <parameter n>"]>[">
```

Special commands:

```
UICL ECHO <string>
```

```
UICL [LIST=<file name> [APPEND] [ALL] | NOLIST]
```

```

UICL [NO]PROTECT=<password> [MODE=[DEMO | FRIENDS | PUBLIC | UNPROTECTED]]
UICL SET DATAPATH=(*[<directory> | $DATAPATH]*)
UICL SET MODE= [BATCH | INTERACTIVE]
UICL SET RESOURCE=[CPU=<cpu time limit> | CORE=<core dump file size>]
UICL SHELL <shell command>
UICL SHOW CPU
UICL SHOW DATAPATH
UICL SHOW RESOURCE
UICL SHOW SUBSTITUTION
UICL SUBSTITUTE [(|<string 1> [|,] <string 2> | CLEAR)]
UICL TRACE [FILENAME] [ON | OFF]

```

C.20 Miscellaneous

```

VM:                                # command prefix
BREAK                              # return from VmCommand
[NO]XSYNCRONIZE                    # X-Windows synchronization

SET ACTIVITY [COLOR=<color name>]
SET BORDER [COLOR=<color name>] [WIDTH=<border width>]
SET HIGHLIGHTWEIGHT=<weight [0..1]>
SET INACTIVITYWEIGHT=<weight [0..1]>
SET TRANSPARENCY THRESHOLD=<value [0..1]>

CANCEL                              # cancel current computation
SET CANCEL WIDGET=[<widget address> | 0]

CREATE FUNCTION VOLUME ...          # for external volume generation
MODIFY VOLUME ...                  # B. Pflesser entry

# Currently unused commands:
SET PAINT COLOR=<color name>
SET RAYDEPTH [CAMERA=[<camera name> | CURRENT]] <depth>
SET SURFACE BOUNDARY ...

```

Danksagung

Diese Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Mathematik und Datenverarbeitung in der Medizin (IMDM) des Universitäts-Krankenhauses Eppendorf (UKE) in Hamburg.

Mein besonderer Dank gilt Herrn Prof. Dr. Höhne für seine intensive Unterstützung meiner Forschungsarbeit. Mit seiner Hilfe konnte ich die Ergebnisse in Fachzeitschriften veröffentlichen und auf internationalen Konferenzen präsentieren.

Herrn Prof. Dr.-Ing. Flessner und Herrn Prof. Dr.-Ing. Stiehl danke ich für die Betreuung seitens des Fachbereichs Informatik. Ihre Anmerkungen und Verbesserungsvorschläge haben wesentlich zur jetzigen Form der Arbeit beigetragen.

Bedanken möchte ich mich auch bei allen Kollegen am IMDM, Bernhard Pflesser, Kay Priesmeyer, Martin Riemer und Dr. Rainer Schubert für die gute Zusammenarbeit im Rahmen des VOXEL-MAN-Projekts, insbesondere bei Thomas Schiemann für seine Hilfe bei der Visualisierung der multiparametrischen Visible-Human-Daten und Andreas Pommert für seine sorgfältige Durchsicht der Arbeit und die sich daraus ergebenden Diskussionen.

Nils Hausig, Henning Krämer, Jochen Nuthmann, Claus Recoschewitz, Norbert Scheller, Christian Seebode, Markus Urban und Hans-Christian Wulf haben im Rahmen eigener Studien- und Diplomarbeiten oder medizinischer Dissertationen das entwickelte Visualisierungssystem verwendet und somit maßgeblich zu dessen Verbesserung, Weiterentwicklung und Stabilität beigetragen.

Die registrierten PET- und MR-Daten wurden freundlicherweise von Dr. Uwe Pietrzyk, Max-Planck-Institut für neurologische Forschung in Köln zur Verfügung gestellt.

Frau Renate Müller danke ich für die orthographische Durchsicht der Arbeit.

Schließlich bedanke ich mich bei meiner Frau Friederike und meinen Kindern Felix, Julia und Lara für ihr Verständnis meiner häufigen Abwesenheit während dieser Zeit.