



Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik
Institut für Simulation und Graphik

**Animationen und Explorationstechniken
zur Unterstützung der chirurgischen
Operationsplanung**

Dissertation

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von: Dipl.-Inform. Konrad Mühler
geb. am 7. September 1979 in Magdeburg

Gutachter: Prof. Dr. Bernhard Preim
Prof. Dr. Thomas Ertl
Prof. Dr. em. Karl Heinz Höhne

Magdeburg, den 28.06.2010

Mühler, Konrad (Dipl.-Inform.):
Animationen und Explorationstechniken
zur Unterstützung der chirurgischen
Operationsplanung
Dissertation, Otto-von-Guericke-Universität
Magdeburg, 28.06.2010.

Für Emilia und Sandra

Zusammenfassung

Der Einsatz immer besserer medizinischer Visualisierungen in der chirurgischen Operationsplanung gewinnt zunehmend an Bedeutung. Ziel der Arbeit war daher die Entwicklung neuer Techniken zur Unterstützung des chirurgischen Operationsplanungsprozesses. Der Fokus lag auf Techniken der 3D-Exploration. Es konnte ein neuartiges Verfahren zur automatischen Bestimmung von Sichtpunkten in Szenen anatomischer Strukturen entwickelt und validiert werden. Die dadurch geschaffenen Grundlagen wurden genutzt, um optimale Kamerapfade in Animationen zu berechnen und ähnliche Sichtpunkte zusammenzufassen. Zur Darstellung zusätzlicher Informationen wurden zwei- und dreidimensionale Szenen mit textuellen Annotationen versehen. Die dabei entwickelten Techniken erlauben die automatische Annotation beliebiger Strukturen mit einer großen Bandbreite an verschiedenen Stilen. Weiterhin wurde ein Framework zur automatischen Generierung von Animationen entwickelt und mit den *Keystates* ein Konzept vorgestellt, das vor allem den komplexen Prozess der Definition von Animationsinhalten vereinfacht und einmal getroffene Visualisierungsentscheidungen wiederverwendbar macht.

Anwendung fanden die entwickelten Techniken in einem neuartigen chirurgischen Trainingssystem, dem LIVERSURGERYTRAINER. Durch dessen Entwicklung konnten viele der Techniken in einem praktischen Umfeld validiert werden. Die im Rahmen der Entwicklung gewonnen Erkenntnisse bezüglich der Gestaltung von Benutzeroberflächen für chirurgische Anwendungen konnte in einen ersten Style-Guide überführt werden.

Es wurde ein spezielles Toolkit für die verbesserte Erstellung chirurgischer Applikationen entwickelt, das Medical Exploration Toolkit. Zusammen mit der Entwicklungsplattform MEVisLAB ist es mit dem METK möglich, nutzerfreundliche Applikationen für den klinischen Einsatz schnell und effektiv zu erstellen. Dabei kann der Funktionsumfang einer Applikation aus einem breiten Angebot aus verfügbaren Visualisierungs- und Explorationstechniken individuell zusammengestellt werden.

Abstract

The main scope of this thesis is the development of new techniques to support the surgical intervention planning. The focus concentrates on techniques for the exploration of anatomical 3d scenes. A new technique to select good viewpoints in such scenes automatically was invented and, e.g., used to generate good camera paths in animations and to cluster similar viewpoints. To provide additional textual information visualizations, techniques for automatic annotation of 2d and 3d scenes with a broad range of styles were developed.

Furthermore, a framework for the automatic generation of animations was established and the concept of keystates was introduced to facilitate the definition process of animations and the reuse of visualizations.

The developed techniques were integrated and evaluated in a new surgical training environment: the LIVERSURGERYTRAINER. The gained knowledge regarding the design of user interfaces of surgical applications was used to create a first style guide for such applications.

For the fast prototyping of medical applications, the Medical Exploration Toolkit was developed. Integrated in the development environment MEVISLAB this toolkit supports an efficient prototyping of applications for clinical use with a lot of state of the art visualization and interaction techniques.

Danksagung

An erster Stelle möchte ich meinem Betreuer und Doktorvater Prof. Bernhard Preim danken. Er hat durch die Schaffung eines unvergleichlichen Arbeitsklimas und durch eine sehr praktische Orientierung in Wissenschaftsfragen nicht nur die vorliegende Arbeit sondern meinen gesamte Art zu forschen maßgeblich beeinflusst.

Ich danke allen aktuellen und ehemaligen Kollegen der Arbeitsgruppe Visualisierung, insbesondere Dr. Christian Tietjen und Jeanette Mönch. Christian Tietjen hat mit mir das METK auf die Beine gestellt und war stets ein kreativer Gesprächspartner. Jeanette Mönch hat mit mir den LIVERSURGERYTRAINER entwickelt und viele der in dieser Arbeit genutzten Datensätze segmentiert.

Danken möchte ich meinen ehemaligen Diplomanden Mathias Neugebauer, Markus Wolff und Stefan Hiller, die wesentlich zu dieser Arbeit beigetragen haben. Dem Bundesministerium für Bildung und Forschung möchte ich für meine Finanzierung im Rahmen des FUSION-Projektes danken. Einen Großteil der softwaretechnischen Arbeit habe ich mit MEVISLAB bewältigt. Für die Berteistellung und den umfassenden Support möchte ich dem Fraunhofer Institut MEVIS und MeVis Medical Solutions in Bremen danken; im Speziellen Christian Hansen, Dr. Felix Ritter, Olaf Konrad und Milo Hindenach. Bei der Vermittlung chirurgischer Grundlagen waren mir Prof. Karl Oldhafer, Christoph Logge und Dr. Gregor Stavrou geduldige und erfahrene Lehrer. Vieles, was ich über Leberoperationen weiß, konnte ich durch ihre Beschreibungen und die Beobachtung ihrer Operationen lernen.

Spezieller Dank gebührt meiner Schwester Friederike Mühler. Mit ihrer Anwendersicht als MTRA war sie mir bei vielen Fragen zur Nutzung radiologischer und chirurgischer Applikationen eine Inspiration. Ich danke meinen Eltern Ute und Roland Mühler. Sie haben mich auf meiner bisherigen Laufbahn bestmöglich unterstützt und viel von ihrer vorbildlichen Art des (wissenschaftlichen) Arbeitens an mich weitergegeben. Abschließend danke ich meiner Freundin Sandra Hartmann und meiner Tochter Emilia für ihre schier unglaubliche Fähigkeit, mich immer wieder durch ihre bloße Anwesenheit zu motivieren und aufzubauen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufgaben und Fragestellungen der Arbeit	3
1.2	Ergebnisse und Beitrag der Arbeit	5
1.3	Aufbau der Arbeit	6
2	Grundlagen der Operationsplanung und Ausbildung	9
2.1	Planung von Leberoperationen	9
2.1.1	Leberanatomie und chirurgische Eingriffe	10
2.1.2	Fallbeschreibung	14
2.1.3	Computerunterstützung bei der Planung	17
2.2	Planung von chirurgischen HNO-Eingriffen	18
2.3	Polygonale Patientenmodelle	18
2.4	Planungssysteme	20
2.4.1	Applikationen zur Leberoperationsplanung	23
2.4.2	Eine Applikation zur HNO-Operationsplanung	24
2.5	Gemeinsame Aspekte des chirurgischen Planungsworkflows	27
2.5.1	Diagnose, Bildakquisition und Segmentierung	27
2.5.2	Individuelle Therapieplanung	29
2.5.3	Interdisziplinäre Diskussion	29
2.5.4	Patientenaufklärung	29
2.5.5	Chirurgischer Eingriff	30
2.5.6	Post-Operative Untersuchungen	30
2.5.7	Zusammenfassende Workflowanalyse	30
2.6	Computerunterstützte chirurgische Ausbildung	31

INHALTSVERZEICHNIS

3	Automatische Generierung von Animationen	35
3.1	Einordnung und Anforderungen	36
3.2	Verwandte Arbeiten zur automatischen Animationsgenerierung	39
3.3	Ein skriptbasiertes Framework zur Animationsgenerierung	41
3.3.1	Paradigmen	41
3.3.2	Konzept der Skriptsprache	42
3.3.3	Beispiele für Animationen	45
3.4	Zusammenfassung und Analyse des Frameworks	47
4	Automatische Bestimmung von Sichtpunkten	49
4.1	Anforderungsanalyse und Einordnung	50
4.2	Grundlagen der Kamerasteuerung	54
4.3	Verwandte Arbeiten	55
4.3.1	Sichtpunktbestimmung in polygonalen Darstellungen	56
4.3.2	Sichtpunktbestimmung in direkten Volumendarstellungen	60
4.3.3	Bestimmung der Sichtbarkeit von Strukturen in polygonalen Darstellungen	61
4.3.4	Anwendungen von Verfahren zur Sichtpunktbestimmung	63
4.3.5	Diskussion und Zusammenfassung	64
4.4	Konzept der parametrisierbaren Kameraposition	65
4.4.1	Vorberechnungen	67
4.4.2	Genutzte Parameter	75
4.4.3	Berechnung guter Sichtpunkte zur Laufzeit	83
4.4.4	Beispiel	86
4.5	Evaluierung des Verfahrens	86
4.5.1	Durchführung	87
4.5.2	Ergebnisse und Diskussion der Evaluierung	87
4.6	Zusammenfassung und Ausblick	90
4.6.1	Anwendung des Verfahrens auf direkte Volumendarstellungen	90
4.6.2	Automatische Parametrisierung von Sichtpunkten	92

5	Anwendungen für automatische Sichtpunktbestimmungen	93
5.1	Gute Sichten auf mehrere Strukturen (Clustering)	94
5.2	Gute Sichten auf minimale Distanzen zwischen Strukturen . .	95
5.3	Automatische Bestimmung von Sichtpunkten im intra-operativen Szenario	99
5.4	Interaktives Blenden	101
5.4.1	Ermittlung verdeckender Strukturen	101
5.4.2	Visualisierung in der Interaktion	102
5.5	Zusammenfassung	102
6	Kamerafahrten für Animationen	105
6.1	Anforderungen an Kamerapfade	106
6.2	Verwandte Arbeiten	108
6.3	Planung von Kamerapfaden zwischen zwei Sichten	110
6.3.1	Informationsmaximierung	110
6.3.2	Kameraausrichtung	111
6.3.3	Zoomverhalten	112
6.3.4	Geschwindigkeit der Kamerabewegung	112
6.4	Zusammenfassung	115
7	Automatische Annotation medizinischer Darstellungen	117
7.1	Motivation	118
7.2	Anwendungsfelder und Anforderungsanalyse	119
7.2.1	Anwendungsfelder in medizinischen Visualisierungen	119
7.2.2	Grundlegende Anforderungen an Annotationen und stilistische Aspekte	121
7.2.3	Annotationstexte	122
7.3	Verwandte Arbeiten	123
7.3.1	2D Annotationstechniken	125
7.3.2	3D Annotationstechniken	126
7.3.3	Annotationen in medizinischen Systemen	131
7.3.4	Zusammenfassung der verwandten Arbeiten	131
7.4	Verfahren zur Beschriftung von 2D-Darstellungen	132
7.4.1	Grundalgorithmus zur Annotationsplatzierung in 2D-Darstellungen	133
7.4.2	Besondere Aspekte der Visualisierung von Annotationen in 2D-Schichtbildern	137

INHALTSVERZEICHNIS

7.5	Verfahren zur Beschriftung von 3D-Szenen	141
7.5.1	Berechnung der Multi-IDBuffer	142
7.5.2	Bestimmung der Ankerpunkte	142
7.5.3	Annotation verdeckter Strukturen	144
7.5.4	Verhalten bei starker Vergrößerung	144
7.5.5	Gruppierung von 3D-Annotationen	145
7.5.6	Symbole in Annotationen	148
7.6	Evaluierung	148
7.7	Zusammenfassung und Ausblick	150
8	Wiederverwendbare Visualisierungen und Animationen	153
8.1	Motivation	154
8.2	Verwandte Arbeiten	156
8.2.1	Effiziente Erstellung von Visualisierungen	156
8.2.2	Wiederverwendung von Visualisierungen	157
8.2.3	Animations-Authoring	158
8.3	Konzept der Keystates	159
8.3.1	Informationen in einem Keystate	159
8.3.2	Erfassung der Informationen	160
8.3.3	Wiederverwendung von Keystates	163
8.3.4	Vergleichbare Konzepte	165
8.4	Anwendung von Keystates in der Interaktion	166
8.4.1	Automatische Generierung von Keystates	166
8.4.2	Anwendungsgebiete für Keystates	168
8.5	Animations-Authoring mit Keystates	168
8.5.1	Interaktionstechniken zum Animations-Authoring	169
8.5.2	Animations-Übergänge	170
8.6	Beispielanwendungen für Keystates	172
8.7	Diskussion und Zusammenfassung	175

9 Das Medical Exploration Toolkit	179
9.1 Motivation und konzeptuelle Vorüberlegungen	180
9.1.1 Motivation des METK	180
9.1.2 Einordnung des METK	181
9.1.3 Konzeptuelle Vorüberlegungen	182
9.1.4 Anforderungen an chirurgische Applikationen	183
9.2 Verwandte Toolkits und Frameworks	184
9.2.1 Visualisierungs-Toolkits für die medizinische Anwendung	185
9.2.2 Die Entwicklungsumgebung MeVisLab	188
9.2.3 Diskussion und Einordnung existierender Lösungen	190
9.3 Architektur des METK	191
9.3.1 Datenmanagement- und Kommunikations-Ebene	193
9.3.2 Visualisierungsebene	196
9.3.3 Explorationsebene	198
9.4 Applikationsentwicklung mit dem METK	202
9.5 Evaluierung des METK	205
9.6 Zusammenfassung	207
10 Der LiverSurgeryTrainer	211
10.1 Projekthintergrund, Analyse und Zielsetzung	212
10.2 Konzept des LiverSurgeryTrainers	213
10.2.1 Anforderungsanalyse	214
10.2.2 Didaktisches Konzept	215
10.2.3 Aufbau und Funktionen	216
10.3 Evaluierung	225
10.3.1 Durchführung der Evaluierung	225
10.3.2 Ergebnisse der Evaluierung	227
10.3.3 Konsequenzen aus der Evaluierung	229
10.4 Spezielle Aspekte von Benutzerschnittstellen chirurgischer Applikationen	229
10.4.1 Style Guides in der Literatur	230
10.4.2 Besonderheiten chirurgischer Benutzerschnittstellen	231
10.4.3 Umsetzung verschiedener exemplarischer Lösungen für chirurgische Benutzerschnittstellen	236
10.5 Zusammenfassung	243

INHALTSVERZEICHNIS

11 Zusammenfassung und Ausblick	245
11.1 Kritische Anmerkungen und Ausblick	246
11.1.1 Internetbasierte Dienste	246
11.1.2 Evaluierung von Animationen	247
A Evaluierung der Annotationsdarstellungen	249

1

Einleitung

Die Planung chirurgischer Eingriffe auf der Basis einer Computerunterstützung erfährt eine immer breitere Nutzung. Längst kommen bei der Planung komplexer Eingriffe weitreichendere Techniken als die reine Darstellung von 2D-Schichtbildern des CT oder MRT zum Einsatz. Zweidimensionale Grauwertbilder können nur bedingt die räumlichen Verhältnisse und die relative Lage der anatomischen Strukturen zueinander vermitteln. Daher ist die dreidimensionale Visualisierung der gewonnenen Volumendaten inzwischen ein Standard, der zum Alltag nicht nur der Radiologie, sondern auch der Chirurgie gehört. Das dabei häufig genutzte direkte Volumerendering stößt jedoch an seine Grenzen, wenn es um die Darstellung schwer voneinander abzugrenzender Strukturen, Inhomogenitäten in der Bildgebung oder quantitative Maße wie Abstände, Volumina oder Versorgungsgebiete geht. Daher kommt es in solchen Fällen oft zu einer Segmentierung wichtiger Strukturen (Abb. 1.1). Segmentierungen liefern semantische Informationen, mit deren Hilfe sich Transferfunktionen für Volumerendering-Darstellungen gezielter erzeugen lassen und dreidimensionale Visualisierungen auch auf Basis polygonaler Netze möglich sind.

Solche Visualisierungen müssen dem Nutzer (z.B. einem Chirurgen) angemessen präsentiert werden. Dabei spielt neben dem Umfang der dargebotenen Informationen – der Nutzer sollte nur die Informationen präsentiert bekommen, die er wirklich braucht oder angefordert hat – auch die Art der Darstellung eine wesentliche Rolle. In der jüngsten Vergangenheit ist daher eine Reihe von Arbeiten entstanden, die sich mit der Hervorhebung wichtiger Strukturen und der angemessenen Visualisierung des Kontextes beschäftigt haben (TIETJEN [2009], BRUCKNER [2008]). Die Exploration solcher Visualisierungen stellt ein wichtiges Element im computerunterstützten Prozess der Planung komplexer chirurgischer Eingriffe dar. Eine Unterstützung des Chirurgen in dieser Phase einer Planung ist dabei

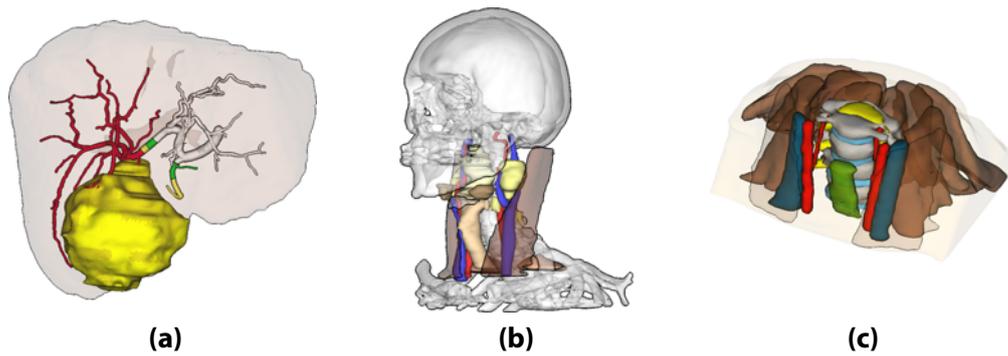


Abbildung 1.1: Visualisierung von Rekonstruktionen auf Basis von Segmentierungen verschiedener CT- und MRT-Datensätze wie sie typischerweise zur Planung von Operationen eingesetzt werden. Zu sehen sind polygonale Netze (a) einer Risikoanalyse für eine Tumorsektion der Leber, (b) mehrere Strukturen des Halses zur Planung einer Halslymphknotenausräumung und (c) eine Halswirbelsäule mit umliegenden Muskeln und Gefäßen zur Planung einer orthopädischen Intervention.

ein großer Beitrag zur vermehrten Akzeptanz solcher Visualisierungen. Möglich werden solche Visualisierungen jedoch erst mit dem semantischen Wissen, um welches die Daten während der Segmentierung angereichert werden. Dieses Wissen ist eine wesentliche Voraussetzung, um später Zusammenhänge zwischen einzelnen Strukturen darstellen und gezielt auf Anfragen des Nutzers reagieren zu können.

Das Wissen um zusätzliche Zusammenhänge und Zugehörigkeiten von Strukturen in Kombination mit dem Kontext einer Applikation ermöglichen die Generierung von Visualisierungen in einem weitestgehend automatischen Prozess. Ohne solches Wissen müsste der Nutzer bei jedem Datensatz aufs Neue verschiedenste Parameter manuell justieren, um zu einer gewünschten Visualisierung zu gelangen, die dann erst die Grundlage seiner chirurgischen Planung sein kann. Mithilfe solchen Wissens ist es jedoch möglich, im Voraus einzelne Visualisierung oder gar Animationen automatisch zu generieren.

Bewegtbilder spielen im Prozess der dreidimensionalen Exploration eine ganz besondere Rolle. Sie sind besser geeignet, räumliche Zusammenhänge zu erfassen, als dies mit rein statischen Darstellungen (ebenfalls dreidimensionaler Strukturen) der Fall wäre. Die kontinuierliche Veränderung von Strukturen (z.B. deren kontinuierliches Einblenden) sowie des Sichtpunktes des Nutzers sind dabei besonders hervorzuheben. Solche kontinuierlichen Veränderungen können den Nutzer dabei in zwei Darbietungsformen erreichen:

- a) als Animation, die der Nutzer betrachten kann, und
- b) als interaktive Szene, in der der Nutzer sämtliche Änderungen selbst steuert.

Der Einsatz von Bewegtbildern muss dabei individuell und abhängig vom jeweiligen Nutzungskontext erfolgen. So bieten sich automatisch ablaufende 3D-Animationen, wie man sie beispielsweise bei medizinischen Lernsystemen vorfindet, für unerfahrene Nutzer mit einem geringen Vorwissen an. Für erfahrene Nutzer dagegen haben sich interaktiv zu steuernde Animationen als vorteilhafter herausgestellt (BOUCHEIX & SCHNEIDER [2009]). Auch wenn hier abhängig vom Applikationskontext bestimmte Voreinstellungen sinnvoll erscheinen, sollte in jedem Fall der Nutzer selbst die letzte Kontrolle über Art und Umfang der möglichen Interaktionen haben.

1.1 Aufgaben und Fragestellungen der Arbeit

Ziel dieser Arbeit ist die Entwicklung und Validierung von Methoden und Techniken zur Unterstützung des Planungsprozesses chirurgischer Eingriffe, im Speziellen der Exploration dreidimensionaler Visualisierungen anatomischer und pathologischer Strukturen. Dabei steht zu jeder Zeit der Chirurg als Nutzer von verschiedenen Software-Werkzeugen im Vordergrund – seine Fragestellungen und Bedürfnisse während der Planung eines Eingriffes sind wesentliche Leitbilder der Entwicklungsarbeit. Jede einzelne Technik wird in einem übergeordneten Kontext einer Gesamtapplikation und des chirurgischen Workflows gesehen. Dies bietet zum einen die Möglichkeit, jede Technik an mehreren Punkten eines längeren Workflows unterstützend zum Einsatz zu bringen. So können Techniken der textuellen Annotation oder der automatischen Sichtpunktbestimmung sowohl während der Planung eines Eingriffes als auch während des Eingriffes selbst eingesetzt werden. Zum anderen erzwingt eine ganzheitliche Betrachtung immer wieder eine Fokussierung und Auseinandersetzung mit den Chirurgen als Nutzern.

Neben der Unterstützung des klinischen Planungsprozesses werden die in dieser Arbeit entwickelten Techniken auch im Kontext der chirurgischen Ausbildung betrachtet. In vielen Bereichen gibt es Überlappungen zum Planungsprozess, wenn es darum geht, angehenden Chirurgen die dreidimensionale Exploration zur Therapieplanung näher zu bringen. Jedoch zeigen sich auch Unterschiede, beispielsweise was den Grad der Unterstützung angeht, die es bei der Entwicklung neuer Techniken zu berücksichtigen gilt.

Ein zweiter Fokus der Arbeit liegt auf der Bereitstellung und Erforschung geeigneter Techniken und integrierter Konzepte zur Unterstützung von Entwicklern im Bereich chirurgischer Software. Wenn man aus einem forschungsorientierten Umfeld heraus Applikationen für Nutzer entwickelt, spielt derzeit immer noch die einzelne (neue) Technik eine zu große Rolle. Neue Techniken müssen vielmehr öfter im Kontext umfangreicher Applikationen betrachtet und validiert werden. Daher müssen auch altbewährte Techniken und Verfahren in eine Applikation integriert sein, die dem Nutzer einen optimalen Umgang mit dieser ermöglicht. Nur dann kann er sich auch auf neue Techniken und deren Validierung konzentrieren. Es ist daher ein wichtiges Ziel dieser Arbeit, den Entwicklungsprozess

von Applikationen mit dem Ziel umfänglicher und gut nutzbarer Endprodukte zu fördern.

Die entwickelten Techniken selbst sind teilweise so grundlegend und allgemein angelegt, dass sie ebenso außerhalb des chirurgischen Anwendungsfokus in anderen medizinischen Bereichen, sogar in gänzlich anderen Feldern der dreidimensionalen Exploration, zur Anwendung kommen könnten.

Folgende Fragestellungen werden im Rahmen dieser Arbeit auf verschiedene Art und Weise untersucht:

- Wie viel Unterstützung des Chirurgen bei der individuellen Exploration von dreidimensionalen Darstellungen ist notwendig und wie kann diese realisiert werden?
- Wie lassen sich einmal getroffene Visualisierungsentscheidungen für spätere Prozesse wiederverwenden?
- Wie können standardisierte Visualisierungen automatisch generiert werden?
- Wie kann die Generierung von Animationen dreidimensionaler Visualisierungen zur Präsentation effektiv gestaltet werden und wie kann deren Wiederverwendbarkeit sichergestellt werden?
- Wie können Entscheidungen des Chirurgen im Verlauf einer Operationsplanung angemessen und möglichst automatisch dokumentiert werden?
- Welchen Einfluss haben neue Planungsverfahren auf die chirurgische Ausbildung und wie lässt sich die Zielgruppe der in der Ausbildung befindlichen Chirurgen gezielt bei der Erlernung der neuen Verfahren unterstützen?
- Welche Aspekte spielen bei der Akzeptanz und Nutzungsintensität von chirurgischen Applikationen eine Rolle und wie können diese effektiv in der Softwareentwicklung erreicht werden?
- Wie können Entwickler von chirurgischen Applikationen effektiv unterstützt werden?

1.2 Ergebnisse und Beitrag der Arbeit

Ausgehend von den Zielstellungen liegt den Ergebnissen der Arbeit ein umfassender Einblick in den computergestützten chirurgischen Planungsprozess zugrunde. Es wird gezeigt, dass es eine Vielzahl an Techniken zur Unterstützung dieses Prozesses gibt. Diese müssen jedoch mit Bedacht und gezielt eingesetzt werden, um ihre vollständige Wirkung entfalten zu können. Das betrifft sowohl den Bereich der Explorationsunterstützung wie auch den Entwurf und das Design von komplexen Applikationen. Im Einzelnen wurden folgende Ergebnisse erzielt:

- Entwicklung eines Frameworks zur automatischen Generierung adaptiver Animationen
- Entwicklung und Validierung eines Konzeptes zur textuellen Annotation von dreidimensionalen Visualisierungen semi-transparenter Strukturen und 2D-Schichtbilddaten
- Ermittlung von Kriterien zur Bestimmung der Güte von Sichtpunkten in dreidimensionalen polygonalen medizinischen Visualisierungen
- Entwicklung und Validierung eines Verfahrens zur automatischen Bestimmung von guten Sichtpunkten mannigfaltiger Natur in Darstellungen multipler Strukturen
- Ermittlung von Kriterien zur Bewertung von Kamerapfaden in dreidimensionalen Darstellungen
- Entwicklung eines Verfahrens zur automatischen Generierung angemessener Kamerapfade in interaktiven Darstellungen und Animationen
- Entwicklung eines Verfahrens zum Authoring von Visualisierungen und Animationen auf der Basis wiederverwendbarer Visualisierungsparameter
- Bereitstellung eines quelloffenen Toolkits zur effektiven und modularen Entwicklung komplexer und anwendungsfreundlicher chirurgischer Applikationen
- Entwicklung eines Softwaresystems zum Training von Therapieentscheidungen in der Leberchirurgie

Insgesamt konnte gezeigt werden, dass erst ein abgestimmtes Zusammenspiel von einzelnen Techniken der Explorationsunterstützung und einem effektiven Entwicklungsprozess mit dem Ziel einer anwenderfreundlichen Applikation einen wirklichen Gewinn für den planenden Chirurgen bedeutet.

1.3 Aufbau der Arbeit

Die Gliederung der Arbeit folgt einem modularen Prinzip. In einem ersten Teil werden Grundlagen geschaffen für die einzelnen, im zweiten Teil vorgestellten, Techniken. In einem dritten Teil werden softwaretechnische Aspekte und ausgesuchte Applikationen vorgestellt, bevor im abschließenden Teil die Arbeit mit einem Ausblick auf zukünftige Entwicklungen zusammengefasst wird.

Kapitel 2 legt die medizinischen Grundlagen der Arbeit dar und analysiert den chirurgischen Planungsprozess hinsichtlich auftretender Gemeinsamkeiten. Es wird ein detaillierter Einblick in Prozesse und Therapieentscheidungen der Leberchirurgie und HNO-Chirurgie gegeben. Weiterhin wird anhand ausgewählter Beispiele der aktuelle Stand der computergestützten Ausbildung im Bereich der Chirurgie analysiert.

Kapitel 3 analysiert die Möglichkeiten des Einsatzes von Animationen in der chirurgischen Operationsplanung. Ein entwickeltes Framework zur automatischen Generierung von adaptiven Animationen wird im Detail vorgestellt.

Kapitel 4 widmet sich der Ermittlung guter Sichtpunkte in dreidimensionalen medizinischen Visualisierungen. Es wird dabei diskutiert, welche Parameter die Güte eines Sichtpunktes ausmachen und wie sich solche Sichtpunkte mithilfe eines neu entwickelten Frameworks effektiv bestimmen lassen. Das Kapitel schließt mit einer Evaluierung der neuen Technik.

Kapitel 5 stellt ausgewählte Anwendungen des zuvor vorgestellten Verfahrens zur automatischen Sichtpunktbestimmung vor. Dabei werden die zur Bestimmung generierten Daten in vielfältiger Art und Weise für weitere Explorationsunterstützungen eingesetzt.

Kapitel 6 analysiert die Bedeutung von Kamerafahrten zwischen verschiedenen Sichtpunkten und präsentiert ein System, welches solche Pfade automatisch und innerhalb vorgegebener Grenzen berechnet.

Kapitel 7 stellt ein umfangreiches Framework zur textuellen Annotation medizinischer Visualisierungen vor. Dabei werden sowohl Strukturen in zweidimensionalen Schichtbilddaten durch Annotationen hervorgehoben, wie auch dreidimensionale Szenen aus semi-transparenten Strukturen erstmals automatisch und in Echtzeit annotiert. Zusätzlich wird eine Evaluierung der neuen Technik unter dem Gesichtspunkt der visuellen Ausprägung verschiedener Annotationsstile vorgestellt.

Kapitel 8 führt ein Konzept zur Erfassung und Wiederverwendung mannigfaltiger Visualisierungsparameter ein. Mithilfe der entwickelten Keystates ist es für einen Chirurgen möglich, einmal getroffene Visualisierungsentscheidungen auf viele ähnliche Darstellungen zu übertragen. Außerdem wird eine Möglichkeit vorgestellt, auf der Basis der Keystates automatisch wiederverwendbare Animationen für den klinischen Alltag zu generieren.

Kapitel 9 präsentiert das MEDICALEXPLORATIONTOOLKIT als ein quelloffenes Toolkit zur effektiven Entwicklung von Applikationen im chirurgischen Umfeld. Es werden die praktisch bei der Entwicklung getroffenen Entscheidungen vorgestellt, wobei auf softwaretechnisch bedeutsame Aspekte gesondert eingegangen wird.

Kapitel 10 stellt den LIVERSURGERYTRAINER, ein Lernprogramm für angehende Leberchirurgen, vor. Dabei wird neben dem Konzept und einer Evaluierung besonderes Augenmerk auf die Aspekte von Benutzerschnittstellen chirurgischer Applikationen gelegt. Zur Anregung weiterer Arbeiten werden ausgewählte Beispiele präsentiert und diskutiert.

Kapitel 11 fasst die Arbeit zusammen und gibt in Form einzelner, dem Autor bedeutsam erscheinender Aspekte, einen Ausblick auf mögliche zukünftige Arbeiten.

2

Grundlagen chirurgischer Operationsplanung und computergestützter Ausbildung

In diesem Kapitel werden als Grundlage späterer Techniken verschiedene Szenarien und der Workflow zur Planung von Operationen vorgestellt. Der Fokus liegt dabei auf den in dieser Arbeit häufig verwendeten Eingriffen an der Leber (Abschnitt 2.1) und im Halsbereich (Abschnitt 2.2). Es wird ein konkreter Fall einer Tumorresektion der Leber vorgestellt (Abschnitt 2.1.2). Ebenso wird die Nutzung von polygonalen Patientenmodellen motiviert (Abschnitt 2.3). Im zweiten Teil des Kapitels werden aktuelle chirurgische Planungssysteme beispielhaft präsentiert (Abschnitt 2.4), gemeinsame Aspekte des chirurgischen Planungsworkflows diskutiert (Abschnitt 2.5) und ein kurzer Überblick über Anwendungen zur computergestützten Ausbildung von Medizinstudenten und Ärzten gegeben (Abschnitt 2.6). Die Verwendung und Entwicklung solcher Systeme bildet eine wesentliche Motivation der in dieser Arbeit entwickelten Techniken.

2.1 Planung von Leberoperationen

Die chirurgische Behandlung von Erkrankungen der Leber stellen durch die Komplexität des Organs und der Eingriffe hohe Anforderungen an den Chirurgen¹

¹Als Chirurgen werden im Folgenden alle medizinischen Spezialisierungen im operativen Bereich angesehen, also auch operierende HNO-Ärzte und operativ tätige Orthopäden.

und bieten damit schon seit längerem ein Feld intensiver Unterstützung durch eine computergestützte Planung. Im Folgenden werden kurz die Leberanatomie, die verschiedenen (chirurgischen) Therapieoptionen sowie ein exemplarischer Fall vorgestellt.

2.1.1 Leberanatomie und chirurgische Eingriffe

Die Leber ist als zentrales Organ des menschlichen Stoffwechsels unter anderem für die Nahrungsverwertung und Galleproduktion zuständig. Sie ist von vier großen Gefäßbäumen mit komplexen Verästelungen durchzogen: Der Lebervene, der Pfortader, der Leberarterie und einem System von Gallengängen zum Abfluss der Gallenflüssigkeit. Da die Leber äußerlich kaum über prägnante Landmarken verfügt, bedient man sich seit 1954 einer Segmenteinteilung nach Couinaud zur groben Strukturierung (COUINAUD [1954], Abb. 2.1). Diese Einteilung orientiert sich an den Verzweigungen der Pfortader. Auch wenn die groben anatomischen Eigenschaften der drei Gefäßbäume (z.B. Anzahl und Ort der ersten Verzweigungen) bei vielen Patienten gleich sind, so lässt sich dennoch eine hohe individuelle Formvarianz des gesamten Organs feststellen (DIRISAMER u. a. [2005]). Hinzu kommen krankhafte Veränderungen, zumeist Metastasen und Tumore (die im Folgenden aufgrund ihres ähnlichen Einflusses auf die Planung synonym verwendet werden), die zu Verschiebungen innerhalb der Leber führen können. Wurde ein Patient schon einmal an der Leber operiert, so stellt sich bei erneuten Operationen die Leber in ihrer (Gefäß-)Anatomie völlig anders dar (LANG u. a. [2005b]). All diese Aspekte machen eine Operation an der Leber kompliziert und erfordern zumeist eine genaue Planung.

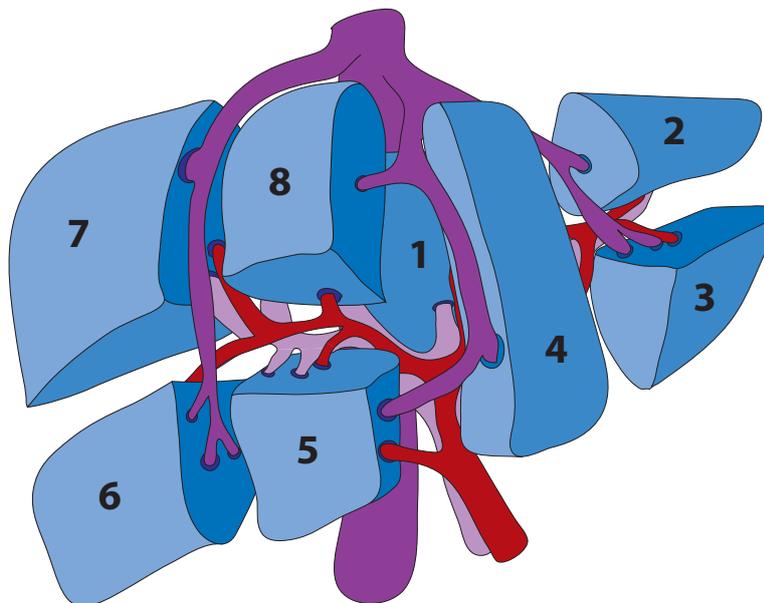


Abbildung 2.1: Segmenteinteilung der Leber nach Couinaud [1954].

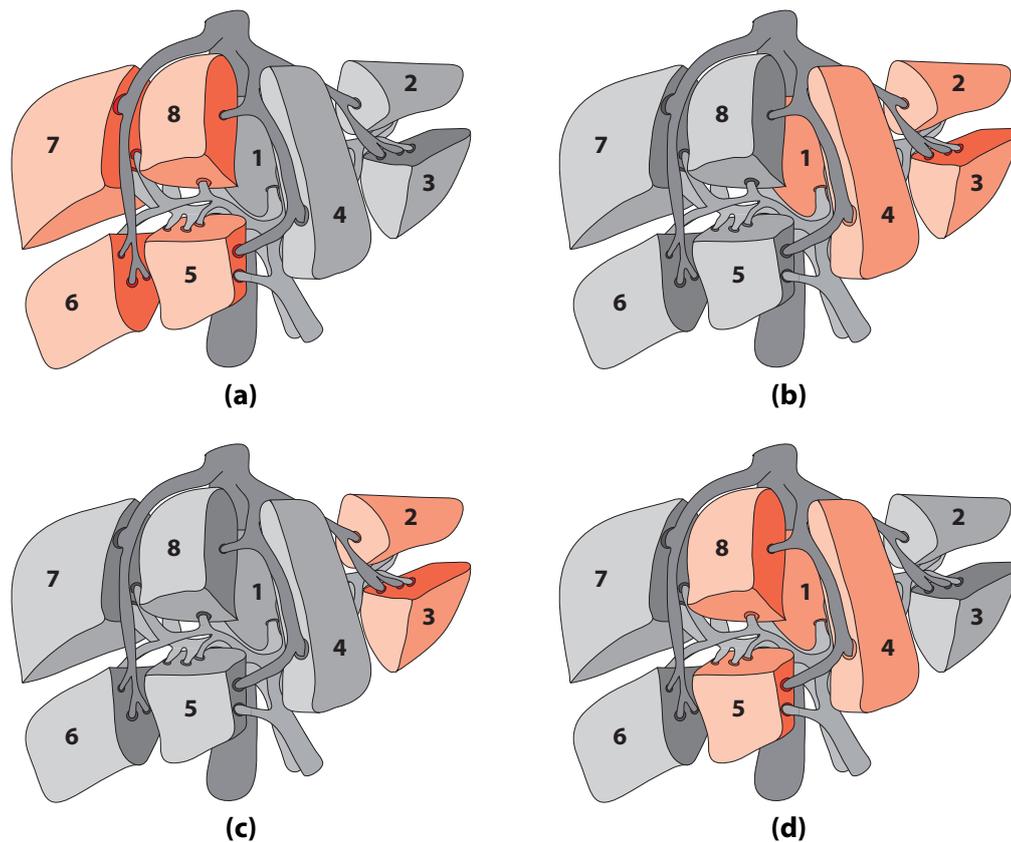
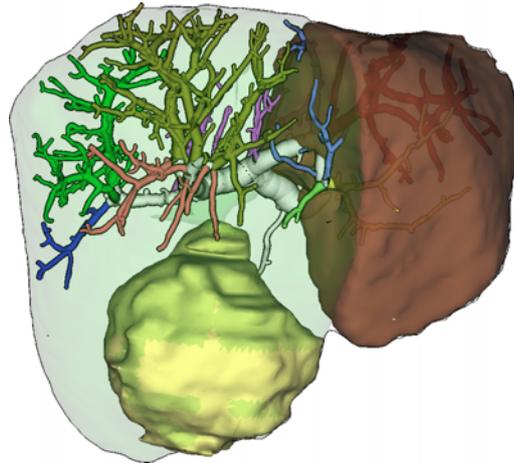


Abbildung 2.2: Verschiedene Leberresektionen. Je nach Befall der Leber mit Metastasen können verschiedene Segmente reseziert werden (hier farblich hervorgehoben). **(a):** Hemihepatektomie rechts. **(b):** Hemihepatektomie links. **(c):** Links laterale Resektion. **(d):** Zentrale Resektion.

Tumorresektion

Eine der häufigsten Operationen an der Leber ist die Resektion eines oder mehrerer Tumore. Bei bösartigen Tumoren zeigt sie die mit Abstand besten Langzeitüberlebensraten ([OLDHAFFER & STAVROU \[2006\]](#)). Dabei werden die Tumore chirurgisch aus der Leber entfernt (*reseziert*). Der Eingriff wird durch die Eigenschaft der Leber begünstigt, sich selbst in gewissen Maßen zu regenerieren. Bei einem Erwachsenen können bis zu 70% des Lebergewebes (Parenchyms) entfernt werden, wenn keine Erkrankung des verbleibenden Gewebes vorliegt ([BROELSCH u. a. \[1990\]](#), [LANG u. a. \[2001\]](#)). Die Leber regeneriert sich innerhalb weniger Wochen wieder auf ihr ursprüngliches Volumen. Wichtig ist dabei die ausreichende Versorgung des verbleibenden Leberparenchyms (Remnant) über die Gefäßsysteme ebenso wie die Garantie eines ausreichenden Abflusses bspw. der Gallenflüssigkeit und des venösen Blutes. Bei der Resektion maligner (bösartiger) Tumore muss auf einen ausreichenden Sicherheitsrand um den Tumor geachtet werden, um eine erneutes Ausstreuen von Tumorzellen zu verhindern. Etabliert hat sich dabei ein Sicherheitsrand von 10 mm.

Abbildung 2.3: Dreidimensionale Darstellung einer geplanten Leberresektion. Das im Patienten verbleibende Lebergewebe ist braun dargestellt, das resezierte Volumen grün. Zusätzlich ist der portalvenöse Gefäßbaum zu erkennen. Zur Vermeidung von größeren Unterversorgungen und Stauungen im verbleibenden Parenchym sowie zur Einhaltung von Sicherheitsrändern wird mehr Gewebe im Umfeld des Tumors reseziert.



Die Einhaltung der Randbedingungen (ausreichende Ver- und Entsorgung des Remnants sowie der Sicherheitsrand) erfordern eine gründliche Planung der möglichen Resektionslinie. Bei der Planung einer solchen Schnittlinie unterscheidet man zwischen *anatomischen* und *atypischen Resektionen*. Bei anatomischen Resektionen wird sich bei der Schnittführung an den (virtuellen) Segmentgrenzen orientiert (Abb. 2.2). Sind solche Resektionen auf nur einen Leberlappen beschränkt, kommt meist eine Standardresektionstechnik wie die linksseitige oder rechtsseitige Hemihepatektomie zum Einsatz (RAAB u. a. [1998]). Atypische Resektionen sind meist erforderlich, wenn durch eine anatomische Resektion nicht ausreichend Leberparenchym erhalten werden könnte. Diese Art der Resektionen können Segmentgrenzen überschreiten und versuchen zumeist große Gefäße zu schonen.

Ein Beispiel für eine Resektion anhand einer 3D-Darstellung ist in Abbildung 2.3 zu sehen. Es ist zu erkennen, dass neben dem Tumor noch viel zusätzliches Leberparenchym entfernt wird. Dies liegt zum einen in der Minimierung der Resektionsfläche begründet. Zum anderen werden bei einer Resektion all jene größeren Bereiche der Leber mit entfernt, für die keine ausreichende Ver- oder Entsorgung garantiert werden kann.

Eine sehr anschauliche und ausführliche Einführung in die anatomischen Grundlagen und die Durchführung von Leberresektionen bieten OLDHAFFER u. a. [2006] und OLDHAFFER & STAVROU [2006].

Tumorablation

Ist aufgrund des Leberzustandes (z.B. durch eine Leberzirrhose), des Zustandes des Patienten oder der ungünstigen Lage eines Tumors eine Resektion nicht möglich,² wird versucht, diesen mittels einer Ablation zu zerstören (BIRTH u. a. [2003], HELMBERGER u. a. [2005]). Dabei wird eine Applikatornadel durch die Bauchdecke in die Leber und den Tumor eingeführt. Durch eine Hitzerzeugung an der Spitze des Applikators (z.B. durch Hochfrequenzstrom bei der Radiofrequenz-Ablation (RFA)) wird der Tumor koaguliert (zerstört).

Für die Planung eines solchen Eingriffes ist entscheidend, ob ein geeigneter Zugangsweg zum Tumor gefunden werden kann. Dieser kann durch Knochen oder große Gefäße blockiert sein. Auch ist im Vorfeld abzuklären, ob der Tumor ausreichend klein für eine rein thermische Zerstörung ist – je nach Art des Applikators können nur Bereiche einer bestimmten Größe koaguliert werden. Ebenso haben größere Gefäße in unmittelbarer Nähe des Tumors einen kühlenden Effekt und können die Wirksamkeit einer Ablation beeinflussen (WELP u. a. [2006]). Die Simulation betroffener Regionen in Abhängigkeit der Gewebeart, umliegender Gefäße und des Applikatortyps ist zwar schon einige Zeit Gegenstand der Forschung (z.B. von KRÖGER u. a. [2009]) – jedoch konnte bisher noch kein Verfahren entwickelt werden, welches eine zuverlässige Simulation in einer praktikablen Zeit durchführt.

In einigen Fällen wird auch auf eine kombinierte Therapie aus Ablation und Resektion gesetzt (KLAR u. a. [2003]). Dies kann zum Beispiel sinnvoll sein, wenn mehrere Tumore innerhalb einer Leber entfernt oder zerstört werden müssen. Dann können einige, chirurgisch schwer zugängliche Tumore mittels Ablation zerstört werden, während andere reseziert werden (Abb. 2.4). Auch können kühlende Gefäße kurzzeitig abgeklemmt werden, um eine Ablation zu ermöglichen.

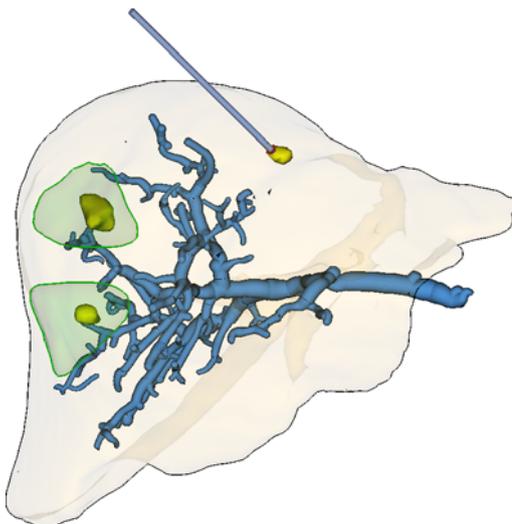


Abbildung 2.4: Dreidimensionale Darstellung einer geplanten Leberresektion zweier Metastasen bei zusätzlicher Ablation einer dritten Metastase. Das zu resezierende Parenchym ist grün dargestellt. Zusätzlich ist der portalvenöse Gefäßbaum visualisiert.

²Oft ist dies auch der Wunsch des Patienten.

Leberlebenspende

Ist eine Leber irreparabel geschädigt oder kann sie aus anderen Gründen nicht reseziert oder abladiert werden, so muss sie komplett entfernt und durch ein Spenderorgan ersetzt werden.³ Leider gibt es in Europa viel weniger Spender als benötigt werden, so dass Anfang der 90er Jahre die Technik der Leberlebenspende entwickelt wurde (RAIA u. a. [1989], STRONG u. a. [1990]). Dabei wird von einem gesunden Erwachsenen ein Teil der Leber entnommen und dem Empfänger verpflanzt. Dies wurde zunächst nur bei Kindern durchgeführt, da hier lediglich ein kleinerer Teil der Leber beim Spender entnommen werden muss. Bei Kindern ist dabei die Letalität auf der Warteliste für Spendenempfänger faktisch eliminiert worden. Kinder haben durch die Technik der Leberlebenspende beste Kurz- und Langzeitüberlebensraten (WALTER u. a. [2008]). Seit Mitte der 1990er Jahre wird diese Operationstechnik aber auch zur Gewinnung von Spenderlebern für Erwachsene genutzt (LO u. a. [1997]). Die Schwierigkeit bei der Leberlebenspende besteht, wie bei der normalen Tumorsektion, darin, möglichst viel Leberparenchym im Spender zu erhalten, welches sich selbstständig regenerieren kann.⁴ Hinzu kommt, dass die entnommene Leberhälfte in ihrer Größe für den Empfänger ausreichend Volumen gewährleisten muss. Bei der Planung einer solchen Operation spielt die genaue Kenntnis der Gefäßanatomie eine noch größere Rolle, da hier nicht nur die ver- und entsorgenden Gefäße des Remnants beachtet werden müssen, sondern auch die Leberhälfte für den Empfänger über ein funktionierendes und vollversorgendes Gefäßsystem verfügen muss. Einen guten Überblick über den aktuellen Stand bei der Planung und Durchführung von Leberlebenspenden bieten SCHROEDER & RUEHM [2005].

2.1.2 Fallbeschreibung

Im Folgenden wird ein konkreter Fall der onkologischen Leberresektion Schritt für Schritt von den ersten Beschwerden des Patienten bis hin zur Operation betrachtet.

Beispielfall

- *Der Patient ist ein 66 Jahre alter Mann, 1,75m groß und 65kg schwer.*
- *Der Patient stellte sich mit starken Bauchschmerzen seinem Arzt vor. Zur Diagnose wurde ein kontrastmittelverstärktes Abdominal-CT angefertigt,*

³Bei der Leberlebenspende zum Austausch einer Leber mit Metastasen ist eine Voraussetzung der Ausschluss extra-hepatischer Metastasen.

⁴Es sollten nicht mehr als 60% der Spenderleber entfernt werden (LANG u. a. [2001]).

auf dessen Basis eindeutig ein großer Tumor in der Leber identifiziert werden konnte.

- Der Tumor wurde punktiert und als maligne eingestuft. Eine Umfelduntersuchung ergab keine weiteren Metastasen. Daraufhin wurde der Patient zur weiteren Behandlung in ein spezialisiertes Krankenhaus überwiesen.
- Zur Entscheidung, ob der Tumor entfernt werden kann und wie eine mögliche Resektion aussehen könnte, wurden die CT-Daten zur Aufbereitung zu MeVis Medical Solutions in Bremen geschickt ([MEVIS MEDICAL SOLUTIONS - DISTANT SERVICES \[2010\]](#)). Eine Ablation konnte aufgrund der Größe des Tumors von vornherein ausgeschlossen werden.
- Dort wurden alle wesentlichen Strukturen der Leber segmentiert: Gefäßbäume, Tumor, Leberparenchym, Gallengänge. Auf der Basis der unterschiedlichen Gefäßbäume wurden die verschiedenen Segmente der Leber automatisch ermittelt. Zusätzlich wurden erste Resektionsvorschläge erstellt (Abb. 2.5). Es wurde eine ausführliche Risikoanalyse erstellt, die für verschiedene Sicherheitsabstände das Volumen des Remnants visuell und quantitativ darstellte.
- Die Daten wurden visuell aufbereitet und sowohl als 2D-Bilder als auch als interaktive 3D-Darstellung zusammen mit den quantitativen Daten zurück zum planenden Chirurgen geschickt (Abb. 2.6).
- Dem Chirurgen standen drei mögliche Resektionsschnitte zur Verfügung (Abb. 2.5):
 - a) eine erweiterte Hemihepatektomie links
 - b) eine zentrale Resektion
 - c) eine Trisektorektomie

Er entschied sich auf Basis der Daten in Bezug auf die Schnittführung an den Gefäßstäben sowie der Volumetrie für eine Trisektorektomie, was einer anatomischen Resektion aller Segmente außer Segment VI und VII entspricht.

- Vier Wochen nach der Vorstellung des Patienten bei seinem Arzt wurde die Operation durchgeführt. In einer fünfständigen OP wurden zunächst die das Resektat versorgenden Gefäße einzeln frei präpariert und verschlossen. Anschließend wurde die Resektionslinie auf der Leberoberfläche eingezeichnet und mittels Wasserstrahldissektor ([RAU u. a. \[2003\]](#)) das Resektat vom Remnant getrennt. Dabei wurde immer wieder ein Ultraschallgerät zu Orientierung im Parenchym und zum Auffinden von großen Gefäßen eingesetzt.
- Der Patient behielt ca. 32% seines Lebervolumens, was ca. 60% des funktionellen Volumens darstellte. Die großen Unterschiede ergeben sich, da zur Berechnung des reinen Lebervolumens ohne Berücksichtigung der funktionalen Versorgung der Tumor hinzugezählt wurde.

- Bei der histologischen Untersuchung des Resektats wurde ein tumorfreier Rand von 2mm identifiziert. Während der Operation entnommene Lymphknoten im direkten Gewebenumfeld wurden ebenfalls als tumorfrei klassifiziert.
- Der Patient konnte 10 Tage nach der Operation mit einer positiven Prognose aus der Klinik entlassen werden.

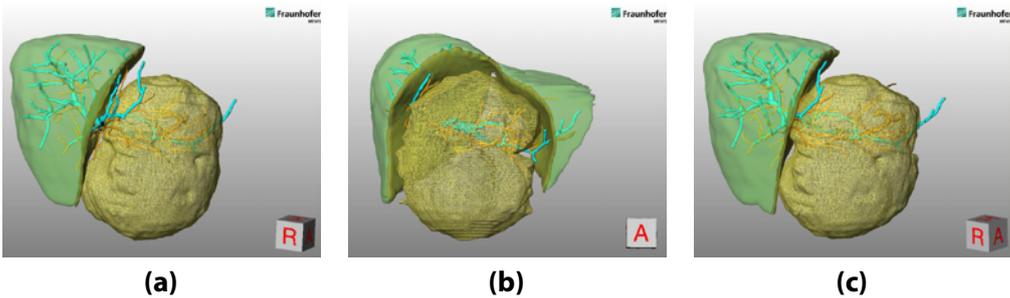


Abbildung 2.5: Drei Vorschläge zur Resektion eines Tumors, wie sie von MeVis Medical Solutions in Bremen zur Planung der Operation erstellt wurden: **(a)** eine erweiterte Hemihepatektomie links; **(b)** eine zentrale Resektion; **(c)** eine Trisektomie;

2. Anatomie

2.1 PV-Anatomie

Territorien	Volumen	Relativ (%)
C2, P3	1100 ml	32,2
P5	280 ml	8,4
P6	240 ml	7,2
P7	360 ml	10,8
P8	910 ml	27,0
Tumor	297 ml	8,9

Minimale Ungenauigkeiten können durch Rundungfehler bedingt sein.

1. Zusammenfassung

1.1 Klinische Informationen

Lebertumor

1.2 Wichtige Hinweise

Bitte überprüfen Sie, ob Folgendes vorliegt:

- Die Farbe P3 und P4 können nicht dargestellt werden.
- Zwei inferiore rechte Lebervenen sind sichtbar.
- Segment P3 wird zunehmend teilweise über den LV3 Ast in die linke Lebervene drainiert.
- Aufgrund der Art des Ausfalls der linken Leberarterie.
- Tumor Feeder geht aus der rechten Leberarterie ab.

Die Risikoanalyse wurde für folgende Sicherheitsabstände vorgenommen: 2 mm (rot), 5 mm (gelb) und 10 mm (grün).

Bei der erweiterten Hemihepatektomie links und der Trisektomie links ist der rechte anteriore Pfortaderast gefährdet. Dgl. sollte der Schwanz dicht am Tumor gefaltet werden, um im rechten anterioren Sektor Gewebe zu erhalten.

Rechnerisch würde bei beiden vorgeschlagenen Resektionen erhalten. Sollten es gelingen, den Tumor von der linken Lebervene abzugrenzen, wäre die Mesoresektomie eine gangbare Alternative.

1.3 Volumina

Struktur	Volumen
Leber	2977 ml
Tumor	1376 ml
Leber (funktionell)	3521 ml

Minimale Ungenauigkeiten können durch Rundungfehler bedingt sein.

Abbildung 2.6: Auszüge aus dem Bericht zur Aufarbeitung eines Falles. Der Bericht enthält quantitative Angaben zu den verschiedenen ermittelten Volumina der Leber und der Segmente sowie drei Resektionsvorschläge. Die Visualisierungen sind Screenshots von 3D-Darstellungen, die manuell erzeugt wurden.

2.1.3 Computerunterstützung bei der Planung

Die Planung einer Leberoperation ohne Computerunterstützung⁵ basierte und basiert noch immer zu großen Teilen auf der bloßen Analyse von 2D-Schichtbilddaten des CT. Hierbei bedarf es einer großen Erfahrung des Chirurgen in der Interpretation der Daten. Auch lassen solche Darstellungen eine visuelle Unterstützung des dreidimensionalen Eindrucks vermissen. Neben diesen subjektiven Einschränkungen der Planung gibt es aber auch Elemente einer Planung, die ohne Computerunterstützung kaum zu bewerkstelligen wären.

Dazu gehört zunächst die **Berechnung des Volumens von Resektat und Remnant**. Für diesen Prozess ist eine Segmentierung des Parenchyms zur Bestimmung des gesamten Lebervolumens unerlässlich. Auch kann die Abschätzung der Volumina ohne Computerunterstützung nur grob erfolgen, z.B. durch eine Planimetrie (OKAMOTO u. a. [1984]), was gerade bei kritischen Grenzfällen dazu führen kann, dass eine Operation nicht durchgeführt wird, weil man glaubt, nicht genügend Lebervolumen erhalten zu können.

Eine weitere wichtige Aufgabe, die einer Computerunterstützung bedarf, ist die **Bestimmung der Lebersegmente**.⁶ Einzig auf der Basis der CT-Schichten lässt sich nicht einschätzen, zu welchem Segment ein bestimmter Teil der Leber gehört – das heißt, von welchem Gefäß es ver- und entsorgt wird. Mithilfe einer Segmentierung der Gefäßbäume und weiteren ausgefeilten Algorithmen (SELLE u. a. [2002], FISCHER u. a. [2005], MEINZER u. a. [2002]) ist es möglich, jeden Voxel des Lebervolumens seinem wahrscheinlichsten Segment zuzuordnen. Auf Basis solcher Daten ist überhaupt erst eine genau anatomische Planung von Resektionen möglich.

Weiterhin werden von den Chirurgen die bessere Veranschaulichung der Lage des Tumors, das Testen und Vergleichen verschiedener Resektionsstrategien sowie die Dokumentation der Planungsergebnisse als weitere große Vorteile einer computergestützten Operationsplanung angesehen (OLDHAFFER u. a. [2001]).

Der Nutzen der computergestützten Planung sowohl für Tumorresektionen (LANG u. a. [2005a]) wie auch Leberlebendspenden (RADTKE u. a. [2006]) ist unbestritten. So zeigen erste Studien eine große Akzeptanz einer dreidimensionalen Aufarbeitung der CT-Daten (GÜNTHER u. a. [2008], LAMADÉ u. a. [2000]). Ebenso konnte eine größere Genauigkeit der Planung bei der Resektion Hepatocellulärer Karzinome (HCC) nachgewiesen werden (YAMANAKA u. a. [2007]). Dabei wurde bei 113 Patienten das Volumen des Resektats einmal mit dem bei der computergestützten Planung ermittelten Volumen und einmal mit dem durch eine klassische Planimetrie (OKAMOTO u. a. [1984]) ermittelten Volumen verglichen.

⁵Die Betrachtung von CT-Schichtbilddaten an einer radiologischen Workstation wird in diesem Zusammenhang nicht als Computerunterstützung angesehen.

⁶Ein Lebersegment ist ein durch bestimmte, definierte Gefäße versorgter Teil der Leber. Es steht in keiner Relation zur sprachlich ähnlichen Segmentierung.

2.2 Planung von chirurgischen HNO-Eingriffen

Die Planung von halschirurgischen Eingriffen auf der Basis segmentierter CT-Schichtbilddaten wird seit etwa 5 Jahren aktiv betrieben. Kritisch sind hier vor allem Fälle, bei denen maligne Tumore im Halsbereich Metastasen in umliegende Lymphknoten ausstreuen. Diese sind dann zumeist vergrößert und müssen bei einer sogenannten Neck-Dissection (Halslymphknotenausräumung) zusammen mit dem Tumor entfernt werden. Entscheidend ist hierbei unter anderem die Frage, welchen Abstand die Lymphknoten zu kritischen Strukturen wie Arterien, Venen oder Muskeln haben (Abb. 2.7). Sind diese Abstände zu gering oder infiltriert ein Lymphknoten eine kritische Struktur, so können diese Lymphknoten nicht herausgenommen werden oder es muss eine radikale Neck-Dissection durchgeführt werden, bei der neben den Lymphknoten z.B. auch Muskeln mit entfernt werden. Solche Entscheidungen, die im Vorfeld getroffen werden, können dem Patienten eine unnötige Operation mit dem damit verbundenen Stress ersparen oder zu einer Anpassung der Operationsstrategie führen.

Die Analyse der Lymphknoten stellt sich als ein aufwendiger Prozess dar, da mitunter bis zu 60 Lymphknoten hinsichtlich ihrer Größe und der verschiedenen Abstände untersucht werden müssen. Die Untersuchungen finden klassisch auf der Basis von CT-Schichtbildern statt, wobei die Ergebnisse von Hand in vorgefertigte Bögen eingetragen werden. Die Ergebnisse einer solchen Planung werden meist in einem Tumorboard vorgestellt und diskutiert, bei dem neben dem planenden HNO-Chirurgen auch Radiologen, Mund-Kiefer-Gesichts-Chirurgen, Internisten, Pathologen und Strahlentherapeuten anwesend sein können. Im Rahmen solcher Zusammenkünfte müssen Fälle schnell und prägnant präsentiert werden, da meist nicht mehr als 5–10 Minuten pro Fall zur Verfügung stehen.

2.3 Polygonale Patientenmodelle

Die in dieser Arbeit vorgestellten Techniken basieren mehrheitlich auf polygonalen Patientenmodellen. Diese Modelle werden durch Segmentierungen aus CT- oder MRT-Schichtbilddaten gewonnen. Auf den aufwendigen Prozess der Segmentierung soll hier nicht weiter eingegangen werden. Für weitergehende Recherchen in dieser Richtung bieten sich folgende Arbeiten an:

- Zur Segmentierung von Strukturen im Kopf- und Halsbereich zur Planung chirurgischer HNO-Eingriffe kamen Verfahren wie LIVEWIRE, die Wasserscheidentransformation oder manuelle Segmentierungen zum Einsatz (CORDES u. a. [2006b]). Neuere Ansätze versuchen, einzelne Strukturen wie Muskeln oder Lymphknoten auch automatisch mithilfe von Modellwissen zu segmentieren (DORNHEIM u. a. [2008]).

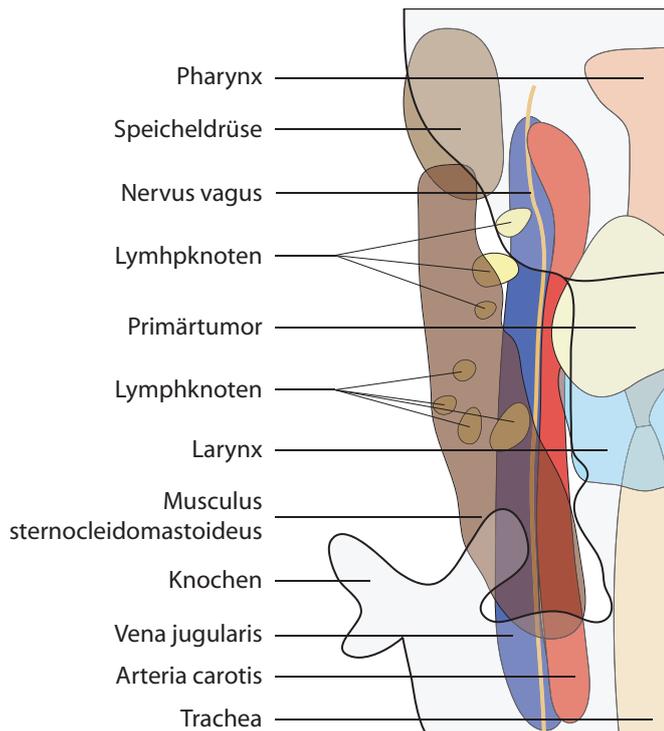


Abbildung 2.7: Schematische Darstellung der im Halsbereich für die Planung von Eingriffen wichtigen Strukturen.

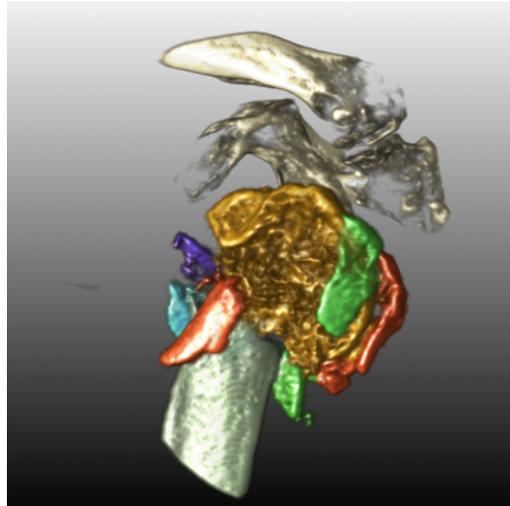
- Bei der Segmentierung von Strukturen der Leber gibt es neue Verfahren, die eine weitgehend automatische Segmentierung der Leber ermöglichen (KAINMUELLER u. a. [2007]). Ein besonderes Augenmerk muss weiterhin auf die Segmentierung der Gefäßbäume gelegt werden. Eine Software, die dies besonders unterstützt, und die auch im Rahmen dieser Arbeit intensiv zum Einsatz kam, ist HEPAVISION (BOURQUAIN u. a. [2002]).

Die automatisch oder manuell erzeugten Segmentierungsmasken wurden mithilfe verschiedener Verfahren in dreidimensionale polygonale Netze überführt. Bei vielen Strukturen kamen dabei Varianten des Marching Cubes Verfahrens mit angepassten Oberflächenglättungen zum Einsatz. Hierzu sei auf die Arbeiten von Ragnar Bade verwiesen (BADE [2008]). Für die Darstellung von Gefäßbäumen der Leber wurden Convolution Surfaces eingesetzt (OELTZE & PREIM [2004]).

Zur Darstellung segmentierter Strukturen bietet sich neben der polygonalen Darstellung auch ein direktes Volumerendering wie das „Tagged Volume Rendering“ an (LINK u. a. [2006]) (Abb. 2.8). Diese Techniken standen jedoch zu Beginn der Arbeit 2005 nicht in ausreichender Qualität und Stabilität zur Verfügung und kamen daher kaum zum Einsatz.

Eine gemeinsame Eigenschaft vieler verwendeter Patientenmodelle ist ihre Kompaktheit und damit einhergehend die Tatsache, dass diese von außen betrachtet werden. Modelle und medizinische Fragestellungen, die auf der inneren Exploration von Szenen, wie z.B. bei der virtuellen Endoskopie, basieren, werden im Rahmen dieser Arbeit nicht behandelt.

Abbildung 2.8: Darstellung segmentierter Strukturen des Ohres in einer direkten Volumendarstellung.



In den folgenden Kapiteln kann der falsche Eindruck entstehen, dass die vorgestellten Techniken nur auf der Basis sehr weniger, ausgesuchter Datensätze erprobt und entwickelt wurden. Das dem nicht so ist, zeigt die kleine Auswahl an im Rahmen dieser Arbeit genutzten Datensätzen aus dem Bereich der HNO-Chirurgie, der Leberchirurgie und der Orthopädie in den Abbildungen 2.9, 2.10 und 2.11. Für die Visualisierung der verschiedenen Techniken wurden zumeist die gleichen Datensätze verwendet, weil sich diese durch eine gute Darstellung der Anatomie anbieten.⁷ Dies soll auch dem fachfremden Leser eine anschauliche Möglichkeit bieten, die zugrunde liegenden medizinischen, anatomischen und chirurgischen Zusammenhänge einfacher zu verstehen.

2.4 Planungssysteme

Im Zuge einer immer stärkeren Etablierung der Computerunterstützung bei der Planung von Operationen sind verschiedene Applikationen entstanden, die diesen Prozess unterstützen sollen. Die folgenden Ausführungen konzentrieren sich auf die Vorstellung der im Rahmen dieser Arbeit relevanten Applikationen zur Planung von Leberoperationen bzw. Eingriffen im Halsbereich konzentriert. Daneben gibt es beispielsweise Applikationen zur Unterstützung der Planung von Schultergelenksoperationen (KREKEL u. a. [2009]), zur Planung von neurochirurgischen Eingriffen – BrainLabs BRAINSUITE (FRASER u. a. [2009]) – oder zur Planung von Eingriffen zur Wiederherstellung von Kieferknochen (WESTERMARK u. a. [2005]).

⁷So werden bspw. in der HNO-Chirurgie für Operationsplanungen im Halsbereich selten CT-Aufnahmen des gesamten Schädels gemacht und nur selten mehr als die wirklich notwendigen Strukturen segmentiert. Beides sind aber Aspekte, die eine Wahrnehmung der Anatomie und eine Wiedererkennung von Strukturen auch durch fachfremde Leser erheblich befördern.

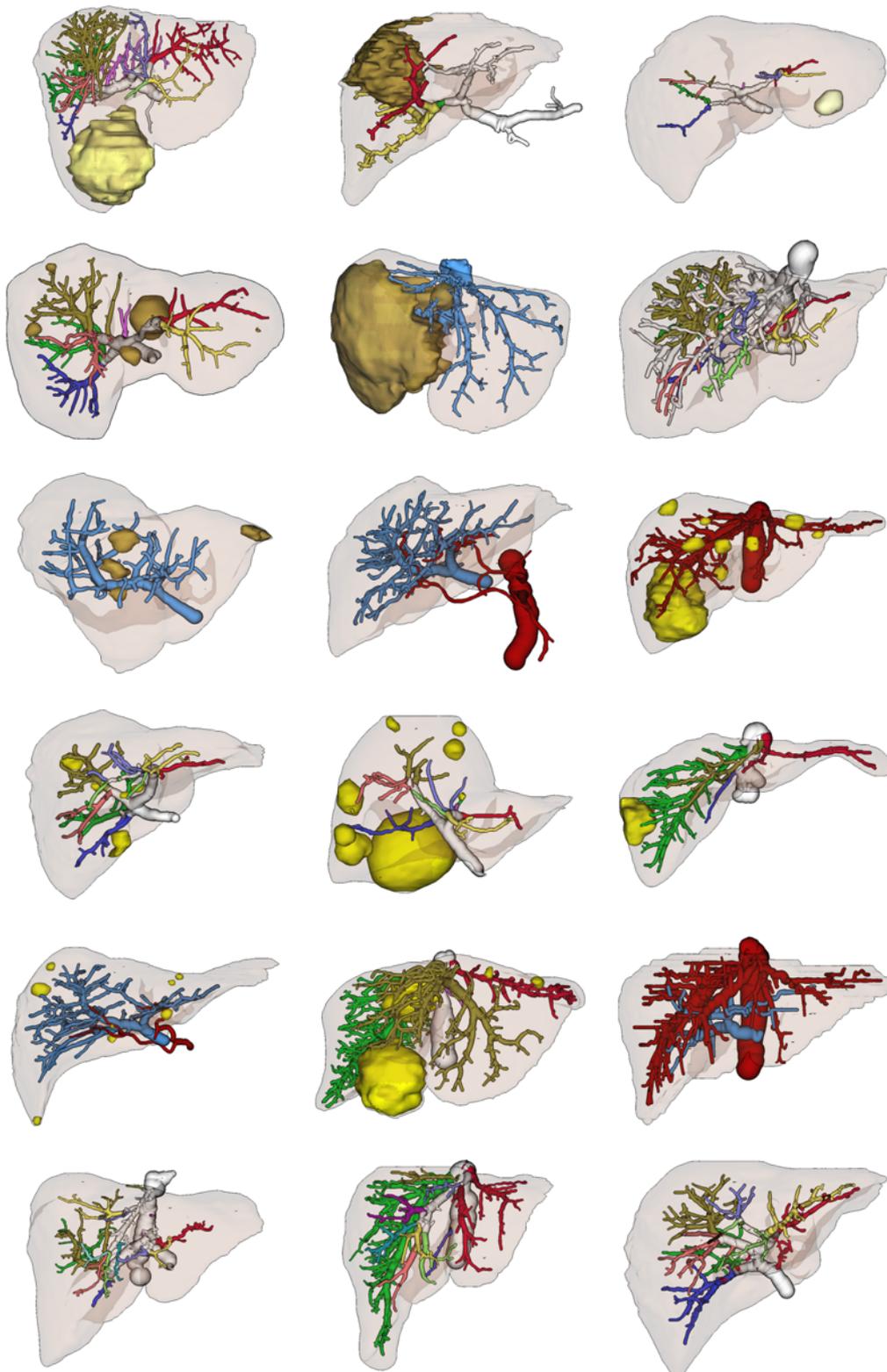


Abbildung 2.9: Eine Auswahl aus ca. 65 Fällen der onkologischen Leberchirurgie und Leberlebenspende, die im Rahmen dieser Arbeit genutzt wurden.

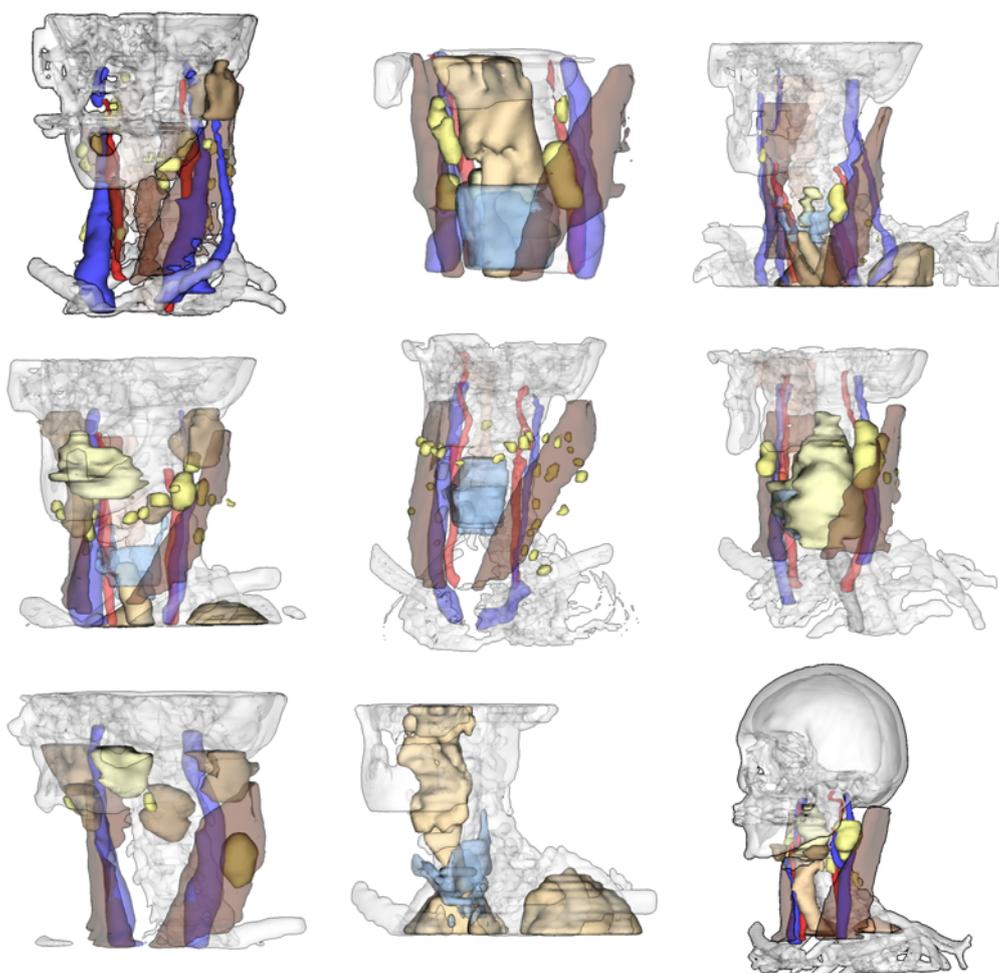


Abbildung 2.10: Eine Auswahl aus ca. 35 chirurgischen HNO-Fällen, die im Rahmen dieser Arbeit genutzt wurden.

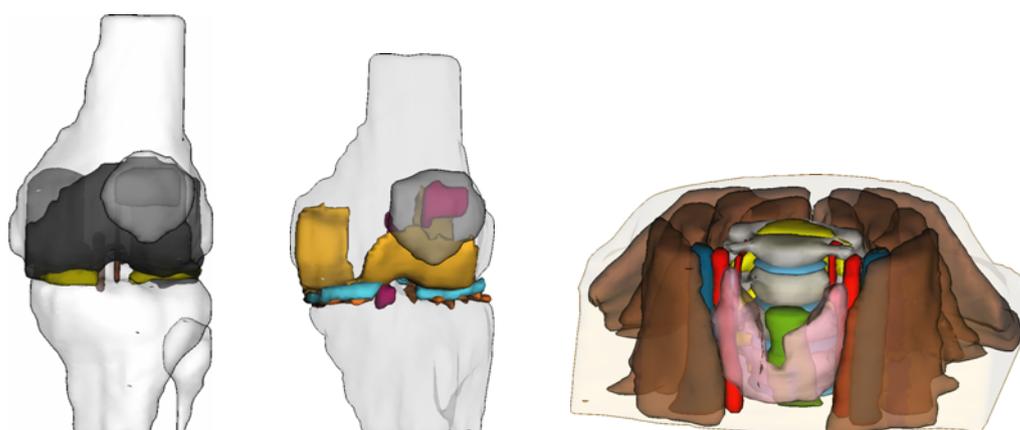


Abbildung 2.11: Eine Auswahl aus ca. 10 orthopädischen Fällen, die im Rahmen dieser Arbeit genutzt wurden.

2.4.1 Applikationen zur Leberoperationsplanung

In Deutschland wurden schon 1992 am Bremer Institut CeVis (heute Fraunhofer MeVis, Bremen) erste Leberoperationen geplant ([MEVIS MEDICAL SOLUTIONS \[2005\]](#)). Bis heute wurden dabei eine ganze Reihe von Applikationen entwickelt, die zum einen die Segmentierung aller wichtigen Strukturen der Leber und die Resektionsplanung unterstützen sollen und zum anderen eine Visualisierung und Aufbereitung der geplanten Daten für den Chirurgen ermöglichen. Mit HEPAVISION ([BOURQUAIN u. a. \[2002\]](#), Abb. 2.12) existiert dabei eine Applikation, mit welcher den speziellen Anforderungen an die Segmentierung der verschiedenen Lebergefäßbäume Rechnung getragen wird. Mit dem MEVISLIVERANALYZER existiert eine Applikation, mit der die aufbereiteten Fälle interaktiv exploriert und visualisiert werden können ([SCHENK u. a. \[2008\]](#), Abb. 2.13).

Kernstück aller Systeme zur Leberoperationsplanung von MeVis ist ein internet-basierter Service, der es angeschlossenen Kliniken aus aller Welt erlaubt, ihre Daten zur Segmentierung und Aufbereitung zu MeVis zu schicken, wo speziell dafür geschultes Personal den immer noch zeitaufwendigen Prozess der Segmentierung durchführt. Bisher konnten so über 1400 onkologische Operationen, 2200 Leberlebenspenden und über 400 Operation an Nieren, dem Thorax und der Pankreas geplant werden. Die Planung einer Resektionsebene ist in HEPAVISION durch die Beschreibung der Ebene in mehreren Schichtbildern möglich. Mithilfe eines Verfahrens von [KONRAD-VERSE u. a. \[2004\]](#) wird aus der so generierten Punktmenge eine dreidimensionale Ebene erzeugt. Diese kann im Nachgang durch den Nutzer in 2D und 3D weiter angepasst und verfeinert werden.

Ein vergleichbares Konzept verfolgt eine Arbeitsgruppe am Deutschen Krebsforschungszentrum (DKFZ). Dort wurde mit LIVERLINE ein System entwickelt, welches dem DKFZ die Aufbereitung von Daten zur Planung von Leberoperationen (onkologischen sowie Leberlebenspenden) an der Universitätsklinik Heidelberg erlaubt ([THORN u. a. \[2004a\]](#)). Die Daten werden von der Klinik für Radiologie an das DKFZ übermittelt, von wo sie aufbereitet an den planenden Chirurgen weitergeleitet werden. Auch hier stellt die Segmentierung und quantitative Analyse der Schichtbilddaten eine wesentliche Herausforderung dar ([THORN u. a. \[1999\]](#)). Neben LIVERLINE existiert mit ORGANICER auch eine spezielle Visualisierungs-Applikation, die auch intra-operativ zum Einsatz kommt und so die Planungsdaten im OP-Saal verfügbar macht ([THORN u. a. \[2004b\]](#)).

An der TU Graz wurde ein System zur Planung von Leberoperationen entwickelt, welches sich auf den Einsatz virtueller Realitäten und Augmented Reality konzentriert ([REITINGER u. a. \[2006\]](#)). Der Nutzer trägt dabei ein Head-Mounted-Display und kann mithilfe eines Stiftes als virtuellem Werkzeug verschiedene Interaktionen vornehmen. So wird das Resektat hier nicht durch die Definition einer Resektionsebene in der Leber beschrieben, sondern durch das Eindringen der Leberoberfläche von außen ähnlich einer Knetmasse erstellt.

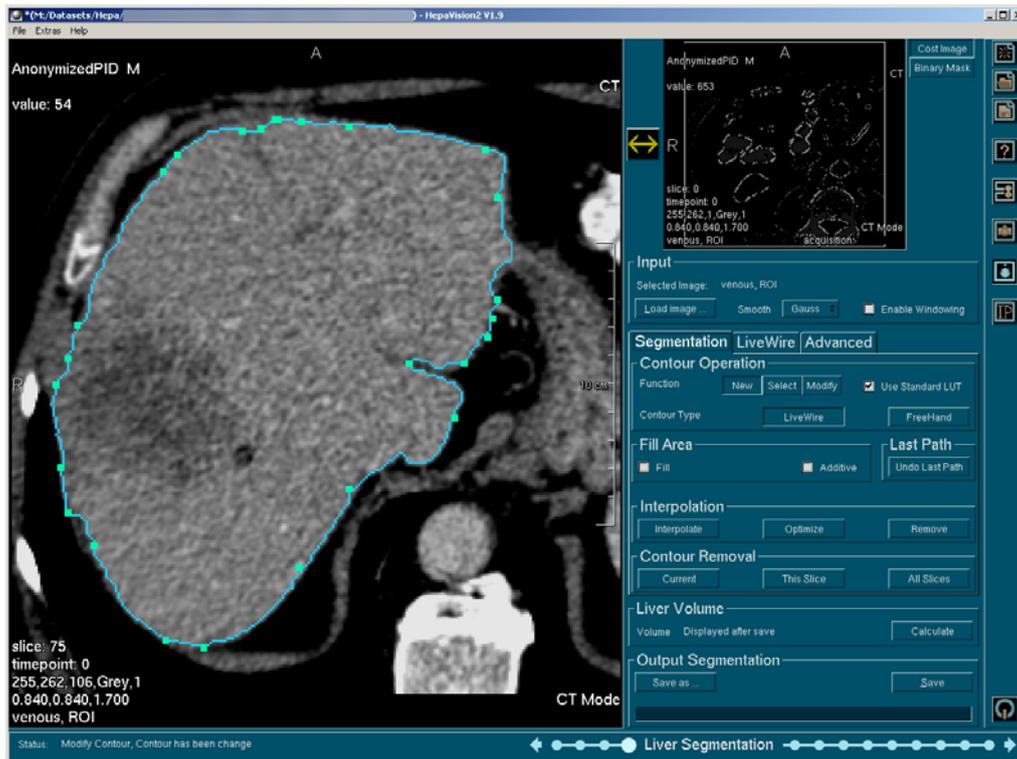


Abbildung 2.12: Screenshot der Applikation HepaVision zur Segmentierung und Analyse von CT-Datensätzen der Leberchirurgie.

2.4.2 Eine Applikation zur HNO-Operationsplanung

In ausgewählten Fällen wird derzeit eine Neck-Dissection auf der Basis von segmentierten Schichtbilddaten geplant. Der Service der Segmentierung und die Aufbereitung der Daten für die Planung wird derzeit vom Institut für Simulation und Graphik der Universität Magdeburg als internetbasierter Service angeboten. Dabei kommt zur Segmentierung die Software NECKVISION zum Einsatz (CORDES u. a. [2006b]). Sie ermöglicht mit individuell auf die verschiedenen Strukturarten angepassten Segmentierungsalgorithmen (z.B. Wasserscheidentransformation für kontrastierte Gefäße oder LIVEWIRE für die Muskeln) die halbautomatische Segmentierung aller relevanten Strukturen.

Die so segmentierten Strukturen können mithilfe des NECKSURGERYPLANNERS (TIETJEN u. a. [2006a], TIETJEN [2009], Abb. 2.14) individuell exploriert werden. Dabei steht die Tumorklassifikation im Vordergrund, bei der der Tumor nach verschiedenen Kriterien (z.B. Lage, Infiltration) klassifiziert wird. Weiterhin werden spezielle Distanzvisualisierungen angeboten, die eine erste grobe Einschätzung von kritischen Abständen erlaubt. Diese können dann in den 2D-Schichtbildern genauer analysiert werden. Die 2D-Schichtbilder werden dabei synchronisiert mit der 3D-Darstellung visualisiert. Die verschiedenen Strukturen können individuell ausgewählt und vermessen werden.

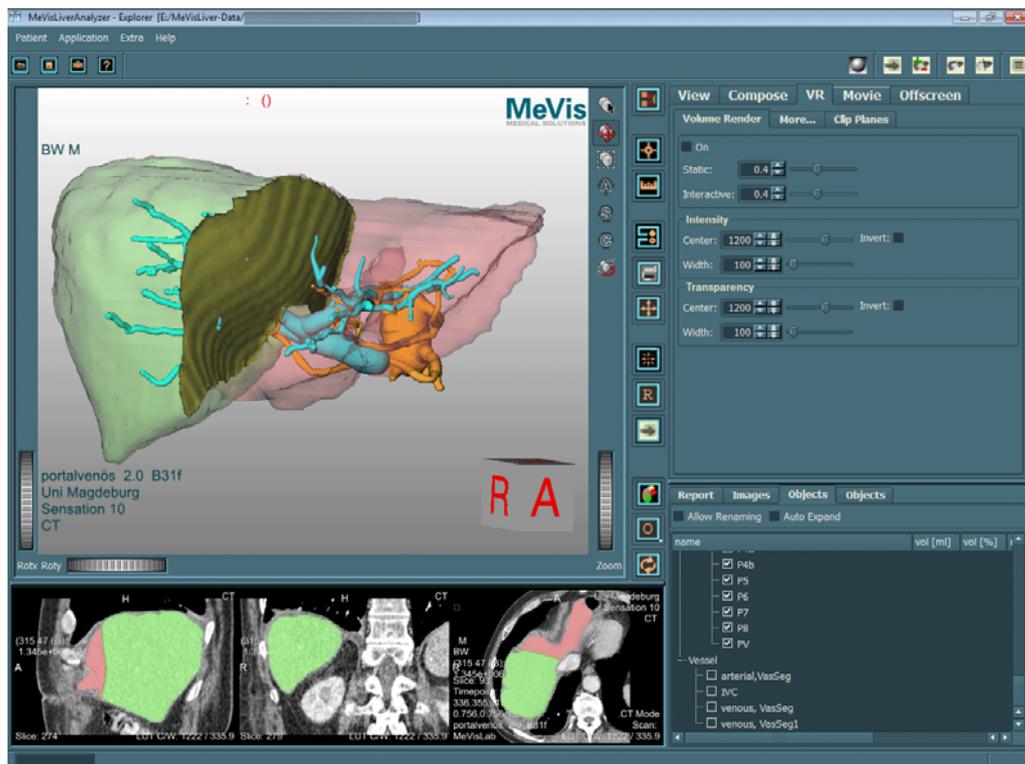


Abbildung 2.13: Screenshot der Applikation MeVisLiverAnalyzer zur Visualisierung und Exploration von aufbereiteten Fällen zur leberchirurgischen Planung. Auffällig ist die große Anzahl an Steuerelementen, Reiteransichten und Teilbereichen, die in ihrer Gesamtheit eine effiziente Benutzung nahezu unmöglich machen.

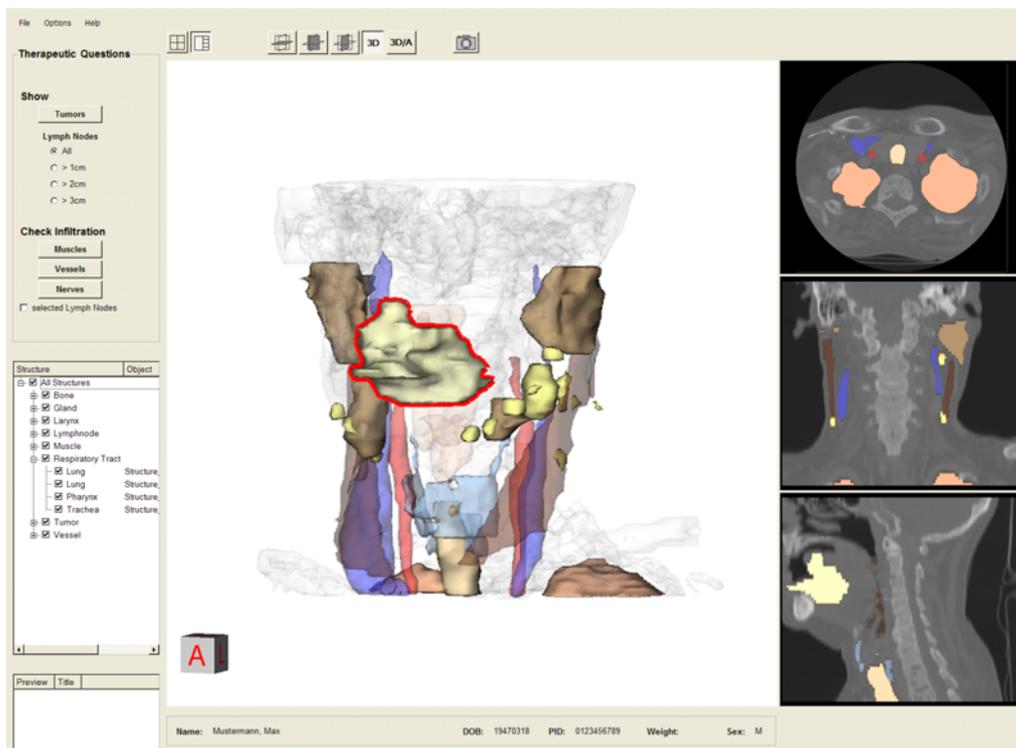


Abbildung 2.14: Screenshot des NeckSurgeryPlanners (Tietjen [2009]), einer Applikation zur Planung von Operationen im Halsbereich. Der Fokus liegt auf der Tumorklassifizierung und der explorativen Beurteilung von Sicherheitsabständen und möglichen Infiltrationen von Lymphknoten und umliegenden vitalen Strukturen.

2.5 Gemeinsame Aspekte des chirurgischen Planungsworkflows

Unabhängig vom chirurgischen Fach und der Art der Operation lassen sich wesentliche Gemeinsamkeiten bei der Betrachtung des Workflows einer Operationsplanung feststellen. Dieser Workflow basiert auf einschlägigen Erfahrungen sowie auf verschiedenen Entscheidungshilfen für Chirurgen zur Therapieplanung, z.B. NAUTA [1998] für die Leberchirurgie oder ALPER u. a. [2001] für die HNO-Chirurgie. Der Ablauf lässt sich in sechs Schritte einteilen (siehe auch Abb. 2.15):

1. Diagnose, Bildakquisition und Segmentierung
2. Individuelle Therapieplanung
3. Interdisziplinäre Diskussion
4. Patientenaufklärung
5. Operation bzw. Intervention
6. Post-operative Untersuchung und evtl. Follow-Up-Studien

Die Schritte werden unter dem Aspekt der Computerunterstützung im Folgenden näher betrachtet.

2.5.1 Diagnose, Bildakquisition und Segmentierung

Der Indikationsstellung für eine Operation folgt in vielen Fällen eine nochmalige medizinische Bildgebung. Dabei werden meist CT- oder MRT-Daten erzeugt (bei der Leberoperationsplanung wird meist ein CT angefertigt (GRUENBERGER [2004])). Viele Eingriffe können und werden derzeit allein auf Basis der 2D-Schichtbilddaten in Verbindung mit einem einfachen Volumerending geplant. In komplexeren Fällen (so auch bei den oben vorgestellten onkologischen Leberoperationen, Leberlebendspenden und HNO-Eingriffen) sind quantitative Analysen und eine 3D-Visualisierung zur verbesserten räumlichen Einordnung notwendig. In solchen Fällen werden die Daten segmentiert. Auch wenn es schon länger Ansätze zu einer vollständig automatischen Segmentierung bspw. der Leber gibt (z.B. KAINMUELLER u. a. [2007]), so ist die Segmentierung derzeit immer noch sehr aufwendig (v.a. die Segmentierung von Gefäßbäumen der Leber (LAMADÉ u. a. [2005])). Unerlässlich ist die Segmentierung für viele quantitative Fragestellungen, die insbesondere die exakte Bestimmung von Volumina, Abständen und Versorgungsgebieten erfordern. Hinzu kommt der oft geringe Weichteilkontrast in den CT-Aufnahmen – hier bedarf es oft großer Erfahrung in der Abgrenzung einzelner Strukturen. Das ist eine Leistung, die derzeit von keinem automatischen System in vollem Umfang erbracht werden kann. Die in dieser Arbeit vorgestellten Techniken basieren zumeist auf segmentierten Datensätzen

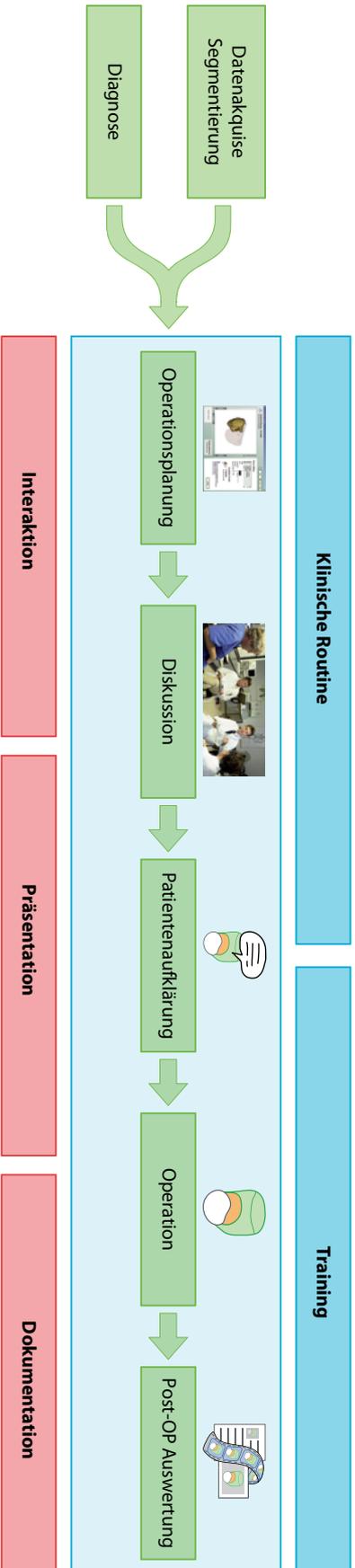


Abbildung 2.15: Allgemeiner Workflow, wie er bei der Planung und Durchführung vieler chirurgischer Eingriffe durchlaufen wird. Nach einer Diagnose und der Bildakquise plant der Chirurg die Operation individuell, um die Ergebnisse anschließend interdisziplinär (z.B. in einem Tumorboard) vorzustellen. Ebenso können die aufbereiteten Daten zur Patientenaufklärung genutzt und intra- und post-operativ eingesetzt werden. Während des gesamten Workflows muss der Chirurg bei der Interaktion mit den Daten, der Präsentation der Daten und der Dokumentation der Ergebnisse unterstützt werden. Weiterhin sind diese Aspekte nicht nur für die klinische Routine, sondern auch für die Ausbildung und das Training von Chirurgen von großer Bedeutung.

und den damit einhergehenden semantischen Informationen zu anatomischen Zugehörigkeiten.

2.5.2 Individuelle Therapieplanung

Aufbauend auf den Segmentierungsergebnissen in Verbindung mit den originalen Schichtbilddaten entwickelt der Chirurg eine individuelle Planung der Operation. Dabei sind Fragen der Operabilität und der Operationsstrategie zu klären. Steht dem Chirurgen entsprechende Software zur Verfügung, z.B. der MEVISLIVERANALYZER (SCHENK u. a. [2008]) oder der NECKSURGERYPLANNER (TIETJEN [2009]), so kann er die Daten dreidimensional explorieren. Dabei können zumeist Strukturen ein- und ausgeblendet und die Kamera frei in der Szene navigiert werden. Hinzu kommen oft spezielle Werkzeuge, die auf den jeweiligen Anwendungsfall zugeschnitten sind – wie die Planung einer Resektionsebene für die Leberoperationsplanung oder Vermessungswerkzeuge (RÖSSLING u. a. [2009]). Zur Dokumentation seiner Therapieentscheidung kann der Chirurg oft Screenshots ausdrucken oder textuelle Berichte verfassen.

2.5.3 Interdisziplinäre Diskussion

In vielen Kliniken (vor allem den Tumorzentren) gibt es regelmäßig interdisziplinäre Besprechungen, bei denen der Fall eines jeden Patienten, der zur Behandlung vorgesehen ist, durchgesprochen wird (z.B. für die Leberchirurgie JAECK u. a. [2003]). Dabei sind neben den Chirurgen des Fachs oft auch Radiologen, Internisten, Pathologen und Fachärzte weiterer Fachrichtungen, die zur Therapie beitragen können, anwesend. Die Fälle müssen im Rahmen solcher Diskussion zügig vorgetragen und präsentiert werden. Oft werden hier auch Therapieentscheidungen des planenden Chirurgen revidiert oder angepasst (GATCLIFFE & COLEMAN [2008]). Als unerlässlich bei Tumorboard-Diskussionen werden inzwischen ausgefeilte Software- und Hardware-Unterstützungen angesehen (BUMM u. a. [2002]) – auch wenn die technische Realisierung derzeit noch in den Anfängen befindlich ist.

2.5.4 Patientenaufklärung

Die Aufklärung von Patienten vor einer Operation gewinnt einen immer größeren Stellenwert und ist oft auch aus haftungsrechtlichen Gründen geboten. Auf der Basis der im Vorfeld aufbereiteten Daten und der Planungsergebnisse kann ein Chirurg heutzutage neben allgemeinen Darstellungen auch auf patientenindividuelle Visualisierungen zurückgreifen. Ebenso besteht hier ein großes Potential für Animationen, die direkt auf den Patientendaten basieren und so diesem

die Operation, ihren Verlauf und mögliche Risiken noch verständlicher machen kann.

2.5.5 Chirurgischer Eingriff

Die eigentliche Operation bildet das Kernstück des chirurgischen Workflows. Die Herausforderungen aus Sicht einer Computerunterstützung liegen hier im Transfer der generierten Planungsdaten in den Operationssaal, um die geplante Strategie auch dort verfügbar zu machen und einen Abgleich mit dem real vorgefundenen Situs zu ermöglichen. Wird die Operation navigiert durchgeführt (z.B. im neurochirurgischen Bereich), so ist eine Registrierung der prä-operativen Planungsdaten auf den Patientensitus zwingend. Auch ohne Navigation sollte es dennoch möglich sein, die Planungsdaten intra-operativ anzupassen, bspw. wenn bei einer Leberoperation neue Metastasen in der Leber gefunden werden, die so prä-operativ nicht erkennbar waren (HANSEN u. a. [2008]). Dabei spielt neben der entsprechenden Hardware und der Frage der Interaktion auch die Gestaltung der Software eine entscheidende Rolle mit derzeit noch vielen Entwicklungsmöglichkeiten (RITTER u. a. [2009]).

2.5.6 Post-Operative Untersuchungen

Zur Kontrolle des Erfolges einer Operation werden meist in von der jeweiligen Operation abhängigen Zeiträumen danach nochmals CT- oder MRT-Aufnahmen angefertigt. So kann beispielsweise die Regeneration der Leber genauso überprüft werden wie die vollständige Resektion eines Tumors oder das Wachstum evtl. nicht resezierter Metastasen. Hierbei sind spezielle Techniken, die eine Registrierung der verschiedenen Patientendatensätze und einen quantitativen und optischen Vergleich erlauben, gefragt.

2.5.7 Zusammenfassende Workflowanalyse

Zusammenfassend lassen sich verschiedene Handlungen aus den Schritten des chirurgischen Planungsworkflows extrahieren, bei denen ein Chirurg besonders unterstützt werden muss (Abb. 2.15):

- Die aufbereiteten Daten müssen **exploriert** werden.
- Die Daten müssen zügig und ansprechend **präsentiert** werden.
- Es gilt, die getroffenen Entscheidungen während des gesamten Workflows ausreichend zu **dokumentieren**.

Unter diesem Dreiklang stehen sämtliche im Rahmen dieser Arbeit entwickelten Techniken. Hinzu kommen noch spezielle Aspekte, die bei der Ausbildung von Chirurgen eine Rolle spielen.

2.6 Computerunterstützte chirurgische Ausbildung

Die Ausbildung von Fachärzten im Allgemeinen und Chirurgen im Besonderen ist sehr auf die jeweilige Klinik, die dort Lehrenden und das vor Ort verfügbare Fallspektrum angewiesen. Einen Großteil des chirurgischen Wissens lernen die angehenden Fachärzte durch die Teilnahme an Operationen und durch die Weitergabe von Wissen durch erfahrene Chirurgen. Werden bestimmte Fälle nur selten oder gar nicht an der örtlichen Klinik behandelt, so kann der Lernende auch nur schwer das für die Durchführung dieser Operationen nötige Wissen erlernen und die dazugehörigen Fähigkeiten erlangen. Für das Erlernen von fallspezifischem Wissen und das Training von Therapieentscheidungen bilden elektronische und computerbasierte Lernmittel schon länger eine Alternative. Auch chirurgische Fähigkeiten sind teilweise ohne konkret verfügbare Fälle zu erlangen. Auf die dafür notwendigen Simulatoren (z.B. [SIMBIONIX USA CORP.](#) oder [DELINGETTE & AYACHE \[2005\]](#)) und Tierpräparate soll hier jedoch nicht weiter eingegangen werden.

Im Folgenden werden einige Systeme zur computergestützten Ausbildung von Ärzten vorgestellt, wie sie aktuell entwickelt werden oder schon erfolgreich zum Einsatz kommen. Es soll dadurch das Potential möglicher weiterer Entwicklung in diesem Bereich, gerade auch in der 3D-Operationsplanung, aufgezeigt werden.

Zwei ältere Systeme, die inzwischen in das ärztliche Curriculum eingegliedert wurden, sind CASUS⁸ ([FISCHER u. a. \[1996\]](#) und [ADLER u. a. \[2006\]](#)) und CAMPUS ([HAAG \[1998\]](#), www.medicase.de). Beides sind fallorientierte Systeme. Die Fälle können über Autorensysteme eingepflegt werden und führen den Lernenden schrittweise durch die Vorstellung des Patienten und seiner Symptome. Sie sollen das Stellen einer Diagnose sowie die Auswahl einer (medikamentösen) Therapie trainieren. Stellenweise kommen multimediale Elemente wie kleine Videos zum Einsatz – im Wesentlichen basieren aber beide Systeme auf der Darstellung von Texten und Bildern.

Rein präsentativ ist das System WEBSURG⁹ ([MUTTER u. a. \[2005\]](#)). WEBSURG vermittelt anatomische Grundlagen und Operationstechniken durch Vorträge, Videos und Beschreibungen klinischer Fälle.

Auf die Vermittlung von anatomischem Wissen auf der Basis von 3D-Modellen sind die Produkte der VOXELMAN-Reihe ausgerichtet ([HÖHNE u. a. \[2003\]](#), [HÖHNE u. a. \[2008\]](#)). Auf Basis der segmentierten Daten des Visible Human ([SPITZER](#)

⁸<http://www.casus.net>, besucht am 2.2.2010

⁹<http://www.websurg.com>, besucht am 2.2.2010

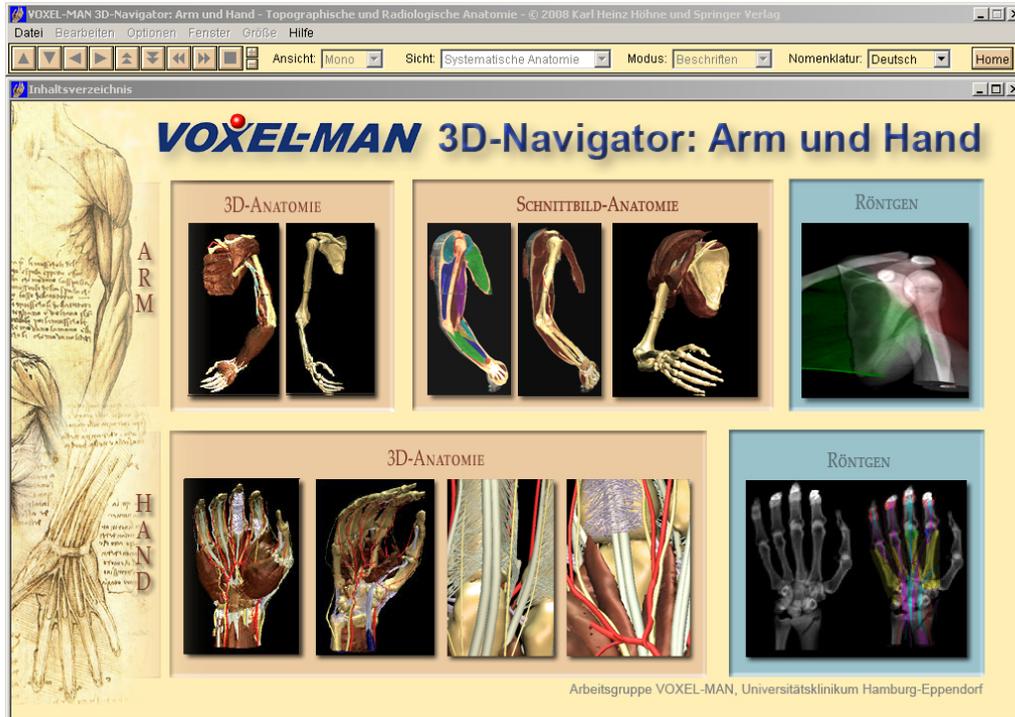


Abbildung 2.16: Screenshot des VoxelMan 2009 (Höhne u. a. [2008]). Die Applikation präsentiert die Anatomie eines gesunden Mannes und greift dabei auf die Daten des Visible Human (Spitzer u. a. [1996]) zurück. Dafür ist eine umfassende Wissensdatenbank hinterlegt, die anatomische und funktionale Zusammenhänge zwischen den einzelnen Strukturen abbilden kann.

u. a. [1996]) werden die anatomischen Strukturen eines gesunden Mannes gezeigt (Abb. 2.16). Sie können individuell exploriert werden, wobei diese Interaktion auf zwei Freiheitsgrade beschränkt ist: zumeist die Rotation um eine Achse und das Ein- bzw. Ausblenden verschiedener Schichtstrukturen von außen (Haut) nach innen (Knochen). Die 3D-Darstellungen sind mit einer Wissensbasis unterlegt, die eine Darstellung der anatomischen Zugehörigkeiten und Versorgungsbeziehungen erlaubt. So können alle Strukturen, die von einem bestimmten Gefäß versorgt werden, eingeblendet werden. Zusätzlich sind die Strukturen in den 3D-Darstellungen annotiert.

Die Firma TELTRA hat ein System zur Vermittlung von Wissen im Bereich der Traumatologie vorgestellt (SEIFFERT u. a. [2000]). Dabei werden dem Lernenden in vielen kleinen Schritten durch Videos und Texte die Abläufe verschiedener Operationstechniken vermittelt. Vereinzelt kommen dabei auch 3D-Modelle der genutzten Implantate zum Einsatz, die der Lernende individuell explorieren (rotieren) kann.

Zur Vermittlung von Wissen für orthopädische Eingriffe haben SOURIN u. a. [2000] ein System vorgestellt, bei dem der Lernende unter anderem virtuelle Implantate zur Fixierung von Knochen an diesen anbringen kann. Die Visualisierung beschränkt sich dabei auf den Knochen und die Implantate und unterlässt

die Darstellung umgebender Strukturen. Die so erzeugte künstliche Darstellung erlaubt zwar die Fokussierung auf das Wesentliche, die Platzierung des Implantates, erschwert aber auch die spätere Übertragung dieses Wissens auf den realen Situs.

Der große Aufwand, der mit der Erstellung von Darstellungen und Animationen zur Vermittlung medizinischen Wissens auf der Basis von 3D-Daten einhergeht, wird sehr gut an der Arbeit von [QUALTER u. a. \[2004\]](#) deutlich. Sie haben auf Basis der Visible-Human-Daten ([SPITZER u. a. \[1996\]](#)) mit dem „3D Animator“ eine Umgebung geschaffen, in der der Lernende interaktiv verschiedene chirurgische Techniken erlernen kann. Dabei wurde besonders auf die Deformierbarkeit der jeweiligen Strukturen geachtet. Die Modelle wurden aufwendig per Hand generiert und erklärende Animationen von spezialisierten Autoren erstellt. Ähnliche Arbeiten sind auch von [JHA u. a. \[2002\]](#) für die Gynäkologie und [LU u. a. \[2005\]](#) für die Anatomieausbildung bekannt.

Das webbasierte System „INTERACTIVE 3D LIVERANATOMY“ zur Vermittlung der Leberanatomie und Variationen bei der Resektion wurde von [CROSSINGHAM u. a. \[2009\]](#) vorgestellt.¹⁰ Dem Nutzer werden dabei auf intuitive Art und Weise die verschiedenen intrahepatischen Strukturen in einer 3D-Ansicht präsentiert. Strukturen können ein- und ausgeblendet werden, es können zusätzliche Informationen zu den Strukturen eingeblendet und die Schichtbilder in Relation zur 3D-Ansicht betrachtet werden. Das System zielt auf Studenten in einem frühen Stadium der chirurgischen Ausbildung ab und basiert auf der Idealdarstellung einer gesunden Leber. Hervorzuheben sind die einfachen und intuitiven Interaktionsmöglichkeiten (siehe auch Abschnitt 10.4.3) – so können die Transparenzen der Strukturen nur zwischen 3 verschiedenen Stufen umgeschaltet werden (opak, semi-transparent, unsichtbar) und die Rotation der Darstellung ist auf jeweils eine Achse beschränkt, was ein ungewolltes „Verreißen“ der Kamera verhindert (Abb. 2.17).

Ein interessanter Ansatz zur Inhaltserstellung für Lernsysteme ist die Generierung durch Studenten in einem dem Reverse Engineering angelehnten Prozess. Bei [BLAKE u. a. \[2008\]](#) wurden Studenten angehalten, Inhalte für ein Lernsystem der Pädiatrie zu erstellen. Bei [PHILIP u. a. \[2008\]](#) wurden den Studenten die Diagnosen für verschiedene Fälle bereitgestellt und sie erstellten dazu entsprechende Fälle. Generell ist zu erkennen, dass Lernsysteme im medizinischen Umfeld immer dann eine größere Akzeptanz und Nutzung erfahren, wenn sie aktiv in das jeweilige Curriculum eingegliedert sind ([FISCHER u. a. \[2008\]](#)).

Zusammenfassend lässt sich feststellen, dass sich die Mehrheit der aktuell verfügbaren Lernsysteme zur fachärztlichen Ausbildung auf der reinen Präsentation von Grundlagen und Techniken, zumeist durch Texte, Bilder und Videos beschränkt. In diesem Sinne sind sie nur die konsequente Erweiterung der klassischen Ausbildung per Buch und Fachliteratur auf den Computer – jedoch ohne die größeren Möglichkeiten einer computerbasierten Ausbildung zu nutzen.

¹⁰http://pie.med.utoronto.ca/VLiver/VLiver_content/VLiver_interactiveLiver.html (besucht am 10.12.2009)

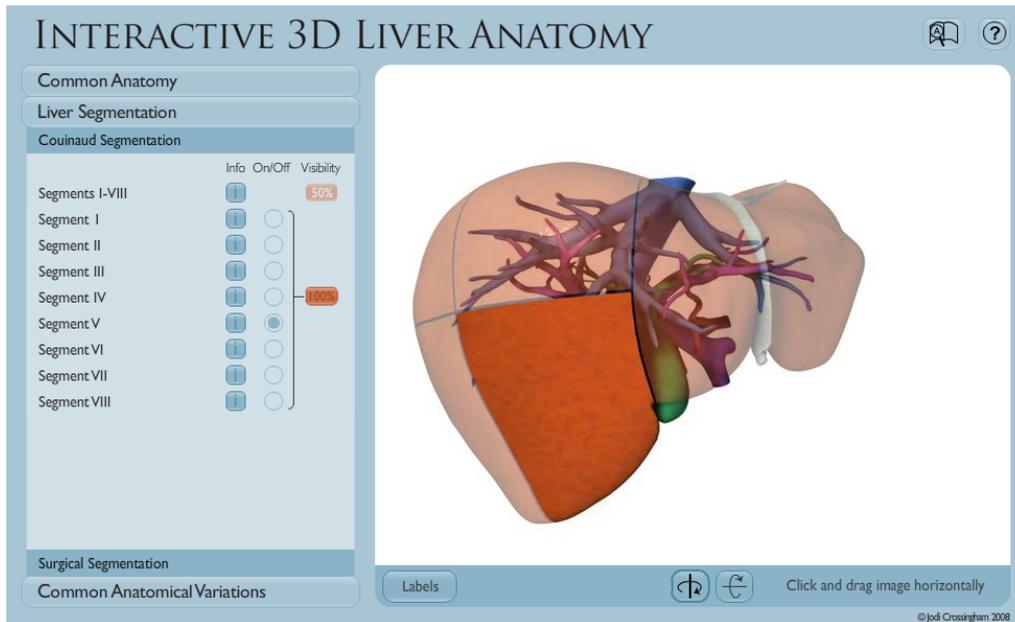


Abbildung 2.17: Das webbasierte System „Interactive 3D LiverAnatomy“ von [Crossingham u. a. \[2009\]](#) zur interaktiven Exploration einer Lebervisualisierung. Das System fällt durch eine intuitive und übersichtliche, weil beschränkte Bedienung auf.

Die Systeme der VOXELMAN-Reihe ([HÖHNE u. a. \[2008\]](#)) bilden in diesem Zusammenhang eine Ausnahme. Die Möglichkeiten der erweiterten Interaktion mit den präsentierten Daten und die individuelle Exploration beispielsweise von dreidimensionalen Modellen der Patientenanatomie bieten ein weites Feld für neue Techniken, die eine verbesserte Ausbildung versprechen: Durch die aktivere Wissensvermittlung (im Gegensatz zur rein rezeptiven) kann die Generierung von tragem Wissen, welches nur schwer angewendet werden kann, vermieden werden ([ZUMBACH u. a. \[2000\]](#)).

3

Automatische Generierung von Animationen zur chirurgischen Planung und Ausbildung

Animationen können die Exploration des Chirurgen bei der Planung von Eingriffen unterstützen, indem sie Visualisierungsübergänge darstellen oder einen animierten Szenenüberblick zeigen. So können sie zum Beispiel bei der Patientenaufklärung eine Bereicherung über die üblichen statischen Bilder hinaus sein. In diesem Kapitel wird analysiert, wo die Vor- und Nachteile solcher Ansätze liegen (Abschnitt 3.1). Weiterhin wird nach einer kurzen Darlegung verwandter Arbeiten (Abschnitt 3.2) ein Animationsframework vorgestellt (Abschnitt 3.3), welches als Vorarbeit dieser Dissertation einen wesentlichen Grundstein für viele der in dieser Arbeit beschriebenen neuen Methoden darstellt.

Eigene Publikationen

Dieses Kapitel basiert u.a. auf folgenden eigenen Publikationen:

MÜHLER u. a. [2006a] Mühler, K.; Bade, R. & Preim, B. *Adaptive script based animations for intervention planning*. In: **Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)**, 2006

MÜHLER u. a. [2006b] Mühler, K.; Bade, R. & Preim, B.: *Skriptbasierte Animationen für die Operationsplanung und Ausbildung*. In: **Proc. of Bildverarbeitung für die Medizin (BVM)**, 2006

3.1 Animationen zur Operationsplanung – Einordnung und Anforderungen

Bei der Nutzung dreidimensionaler Darstellungen von segmentierten Strukturen zur Operationsplanung werden zur Zeit besonders statische Bilder (Screenshots) und interaktive Szenen genutzt. Die **Vorteile** der beiden Formen liegen auf der Hand:

- **Statische Bilder** liefern einen schnellen Zugriff auf die Informationen, die der Erzeuger solcher Darstellungen vermitteln möchte. Sie sind einfach zu handhaben, ausdrückbar, lassen sich per Email verschicken und mit einfachen Bildbearbeitungsprogrammen um Annotationen erweitern.
- In **interaktiven Darstellungen** hat der Nutzer die vollständige Kontrolle über die Szene. Er kann die Szene aus einer ihm wünschenswerten Richtung betrachten, kann Strukturen ein- und ausblenden und zumeist auch deren Darstellungsstil (z.B. Farbe und Transparenz) anpassen. Durch die Bewegung der Szene erhält der Nutzer einen fast perfekten dreidimensionalen Eindruck der Darstellung und kann so Abstände und Relationen verschiedener Strukturen sehr gut beurteilen.

Allerdings weisen beide Darstellungsformen schwerwiegende **Nachteile** auf:

- **Statische Bilder** vermitteln nur einen sehr begrenzten dreidimensionalen Eindruck. Für den Betrachter ist es schwer, Abstände zwischen Strukturen und deren Lage im Raum realistisch abzuschätzen. Auch können verschiedene Strukturen durch andere verdeckt sein – ein Umstand, der sich z.B. in einer interaktiven Darstellung durch eine individuelle Verschiebung der Kamera oder das Ausblenden verdeckender Strukturen beheben lässt. Ebenso ist keine Anpassung der Darstellung an individuelle Bedürfnisse möglich. Wo eine fehlende Modifizierung von Strukturfarben noch tolerabel wäre, ist die fehlende Möglichkeit zusätzliche Strukturen einzublenden oder störende Strukturen auszublenden ein großer Nachteil statischer Bilder.
- Die Manipulation einer **interaktiven Darstellung** ist gerade für wenig erfahrene Nutzer ein sehr zeitintensiver Prozess. Um zu einer gewünschten Darstellung zu gelangen, sind mitunter eine Vielzahl von Strukturen ein- und auszublenden und in ihrem Darstellungsstil anzupassen. Auch muss die Kamera an einen Punkt navigiert werden, von dem aus die gewünschten Informationen optimal zu erfassen sind. Eine Reproduzierbarkeit ist bei einer klassischen interaktiven Darstellung nicht gegeben. Bisher kann eine einmal erzeugte Darstellung nur als statisches Bild gespeichert werden, wobei alle interaktiven Elemente und Möglichkeiten verloren gehen.

Diese schwer wiegenden Nachteile liefern die Motivation, als Alternative zu statischen Bildern und interaktiven Darstellungen eine dritte Möglichkeit der Datenvisualisierung näher zu betrachten – die **Animation**. Animationen sind eine

Abfolge vieler Einzelbilder, die in der Schnelligkeit der Abfolge eine Bewegung oder kontinuierliche Veränderung vortäuschen. Gemeinhin sind Animationen nicht interaktiv. Sie werden abgespielt und können betrachtet werden. Eingebettet in ein interaktives Umfeld können kurze Animationen jedoch „interaktiv wirken“. Animationen in Form kurzer Kamerafahrten zu einem neuen Sichtpunkt oder zur Einblendung zusätzlicher Strukturen können parallel zur eigentlichen Exploration des Nutzers in einer 3D-Darstellung ablaufen. Für die weitere Arbeit werden daher die folgenden zwei grundsätzlichen Arten von Animationen unterschieden:

1. **Videoanimationen:**

Ein Video wird einmal erzeugt und kann danach beliebig oft abgespielt werden. Mit heute verfügbaren Playern kann es auch an beliebigen Stellen unterbrochen und fortgesetzt werden. Die in einer Videoanimation präsentierten Visualisierungen können jedoch nicht mehr angepasst werden.

2. **Interaktive Animationen:**

Interaktive Animationen wirken immer im Zusammenhang mit interaktiven (hier meist dreidimensionalen) Szenen. Einzelne, kleinere Veränderungen an der Szene werden animiert dargestellt, um dem Nutzer die Arbeit zu erleichtern, z.B. durch einen automatischen Flug zu einem guten Sichtpunkt, oder um einen flüssigen Übergang von einer Visualisierung zu einer anderen zu gewährleisten. Solche Übergänge sind wichtig für den Erhalt der Orientierung des Nutzers, der einen kontinuierlichen Übergang zumeist angenehmer als einen abrupten Wechsel der Darstellung finden wird. In der chirurgischen Exploration sind solche Übergänge bei einem Fokuswechsel, z.B. von einer Metastase auf eine andere, sinnvoll.

Im Kontext des chirurgischen Workflows sowie der elektronisch basierten Ausbildung finden beide Animationsarten in folgenden Bereichen Anwendung:

1. **Exploration:** Bei der individuellen Exploration zur chirurgischen Planung bietet sich der Einsatz interaktiver Animationen aus den oben angesprochenen Gründen der Unterstützung des Nutzers an. Vorgefertigte Videos sind hier nur sinnvoll, wenn zur Planung keine spezielle Software zur Verfügung steht und der Chirurg auf einfache Screenshots und Videos angewiesen ist.
2. **Dokumentation:** Geht es um die Dokumentation der Entscheidungen einer Operationsplanung, so bieten sich neben einzelnen statischen Bildern auch automatisch aus den Visualisierungen generierte Videos an. Im optimalen Fall bilden sie alle durchgeführten Explorationen des Chirurgen ab (Kapitel 8), oft ist aber auch eine reine Dokumentation der zugrunde liegenden 3D-Daten ausreichend.
3. **Präsentation:** Für interdisziplinäre Diskussionen, wie das in Abschnitt 2.5.3 angesprochene Tumorboard, bieten sich sowohl vorgefertigte Videos als

auch interaktive Animationen an. Videos haben den Vorteil der schnellen Verfügbarkeit und Nutzung, unabhängig von einer speziellen Software. Jedoch kommt in neueren Publikationen (TIETJEN [2009]) auch das Unbehagen über zu langwierige Videos in den zeitlich straff organisierten Besprechungen zum Ausdruck. Ist ihr Einsatz aus bestimmten Gründen unverzichtbar, so sollten vorgefertigte Videos gut durchdacht und auf das Wesentliche beschränkt sein.

Interaktive Animationen bieten sich zur Präsentation von Fällen insofern an, als dass sie die Explorationen extrem beschleunigen können. Eine durch interaktive Animationen vorgenommene Exploration zur Präsentation haben den Vorteil, individuell auf einzelne Aspekte des Falles je nach Bedarf genauer eingehen zu können. Wird die Präsentation durch den planenden Arzt selbst durchgeführt, so entfällt die Zeit, sich mit der Darstellung vertraut zu machen.

Bei ortsfernen Präsentationen, z.B. zur Unterstützung einer Fallaufbereitung, wie sie von MeVis (MEVIS MEDICAL SOLUTIONS - DISTANT SERVICES [2010]) vorgenommen wird, sind vorgefertigte Videos eine gute Ergänzung zu den statischen Bildern, Tabellen und Erklärungen in der Fallpräsentation. Sie geben einen zusätzlichen dreidimensionalen Eindruck der Planungsdaten (siehe auch Abbildung 2.6 in Kapitel 2).

4. **Ausbildung:** Für die Ausbildung angehender Chirurgen im Bereich der 3D-Planung bieten sich vorgefertigte Videos an. Der Lernende kann sie im Rahmen einer Applikation wie dem LIVERSURGERYTRAINER (Kapitel 10) betrachten und so wichtige Aspekte eines Falles vermittelt bekommen. Vorgefertigte Lernvideos haben zudem den Vorteil des Einsatzes auch außerhalb der Lernapplikation, z.B. auf einer Lernplattform im Internet (z.B. WEBOP GMBH [2009]). Jedoch sind für die in Lernapplikationen enthaltenden Explorationsabschnitte interaktive Animationen eine große Unterstützung des Lernenden, der sich so auf die wichtigen Aspekte konzentrieren kann und nicht über die Maßen durch die möglicherweise neue und ungewohnte Technik der 3D-Navigation abgelenkt wird.
5. **Patientenaufklärung:** Zur Aufklärung eines Patienten vor einer Operation bieten sich Animation generell als ein zusätzliches Mittel der visuellen Unterstützung der Erklärungen des Chirurgen an (HERRMANN [2002], GLITTENBERG & BINDER [2008]). Animationen, die auf der Basis der realen Daten des Patienten erstellt wurden, können diesen Effekt verstärken. Inwieweit neben vorgefertigten Videos auch interaktive Animationen zum Einsatz kommen könnten, wird derzeit im Wesentlichen noch durch den Grad der Verfügbarkeit entsprechender Technik in Form geeigneter Applikationen vor Ort begrenzt.

Sowohl vorgefertigte Videos als auch interaktive Animationen stellen ganz spezielle Anforderungen an den technischen Prozess ihrer Generierung. So muss ihre Generierung zu allererst **automatisch** erfolgen. Dieser Aspekt ist bei interaktiven Animationen offensichtlich. Bei der Erstellung von Videos darf dem Chirur-

gen kein nennenswerter zusätzlicher Aufwand entstehen. Besonders bei interaktiven Animationen muss die Generierung weiterhin in **Echtzeit** unbemerkt vom Nutzer erfolgen. Auf eine Nutzerinteraktion hin (z.B. die Selektion einer Struktur) muss die Animation sofort ausgeführt werden. Ebenso muss sich die Animation an die jeweiligen Patientendaten **dynamisch anpassen (adaptieren)** können. Dies gilt sowohl im interaktiven Fall, bei dem durch ein- oder ausgeblendete Strukturen Animationen komplett unterschiedlich ausfallen können, als auch für Videos, die, so sie automatisch erzeugt werden sollen, sich an den immer wieder neuen Patientendaten orientieren müssen, um eine optimale Präsentation der Informationen zu erreichen.

Die **Informationsdichte** ist ein weiterer wichtiger Aspekt der hier betrachteten Animationen. Diese müssen über ihre gesamte Laufzeit hinweg ein Maximum an Informationen liefern und längere Bereiche ohne zusätzliche Informationen, z.B. Kameraflüge über uninteressante Regionen, vermeiden.

Aufbauend auf der vorgestellten Klassifizierung und den gestellten Anforderungen an Animationen zur Operationsplanung werden im Folgenden verwandte Arbeiten unter diesen Aspekten analysiert, um anschließend ein skriptbasiertes Framework zur automatischen Animationsgenerierung genauer vorzustellen.

3.2 Verwandte Arbeiten zur automatischen Animationsgenerierung

Einen ausführlichen Überblick über Arbeiten zur Animationsgenerierung mit einem Fokus auf skriptbasierten Methoden bietet [MÜHLER \[2005\]](#). Als wesentliche Arbeiten seien hier die Forschungen von Iserhardt-Bauer zu nennen ([ISERHARDT-BAUER u. a. \[2001\]](#), [ISERHARDT-BAUER u. a. \[2002\]](#)). [ISERHARDT-BAUER u. a. \[2002\]](#) haben ein Framework zur Erstellung automatischer Videos zur Analyse intrakranieller Aneurysmen auf der Basis von CT-Angiographien entwickelt. Dabei werden zunächst mittlere und große Aneurysmen automatisch im Datensatz detektiert, um anschließend automatisch eine vordefinierte Transferfunktion an den Datensatz anzupassen. Dabei kam ein Verfahren von [REZK-SALAMA u. a. \[2000\]](#) zum Einsatz, bei welchem durch eine Histogrammanalyse und eine Adaption an die Gradienten im Datensatz eine Template-Transferfunktion geeignet angepasst wurde. Zusätzlich wurden Clipping-Ebenen platziert und durch einen automatischen Kamerapfad vorbestimmte Punkte angeflogen. Das Video wurde offline gerendert und über einen Webservice an die Nutzer ausgeliefert ([ISERHARDT-BAUER u. a. \[2001\]](#)). In einer Evaluierung wurde festgestellt, dass durch die automatische Erstellung der Videos die verschiedenen Datensätze objektiv vergleichbar wurden und eine große Erleichterung für die nutzenden Ärzte darstellten ([ISERHARDT-BAUER u. a. \[2001\]](#)). Wieder aufgegriffen wurde der Ansatz der Web Services von [RÖSSLER u. a. \[2007\]](#), wobei nicht mehr nur Videos über eine spezielle Java-Applikation angeboten wurden, sondern die Techniken

zur automatischen Anpassung von Transferfunktionen und Kamerasteuerung genutzt wurden, um interaktive Darstellungen ähnlich den Quicktime-Object-Movies von Apple (APPLE [2009b]) zu generieren.

WOHLFART & HAUSER [2007] stellten ein System zur Generierung von Animationen medizinischer Visualisierungen zu Lehrzwecken vor. Ihr System erlaubte es einem Autor, eine Animation als „story“ als eine Abfolge verschiedener Visualisierungen zu gestalten. Die Autoren vertraten die These, dass der Betrachter einer Animation einen tieferen Einblick in die präsentierten Daten bekommt, wenn er die Möglichkeit zur Interaktion mit der Animation hat. Daher kann die Animation für eine individuelle Exploration unterbrochen werden. Der Autor einer Animation erstellt die verschiedenen Zustände einer Animation und kann sie manuell in ihrer Abfolge anordnen und editieren. Dennoch ist es immer noch ein aufwendiger Prozess, auf diesem Wege eine Animation zu erstellen. Für einen Einsatz in der klinischen Routine oder zur Operationsplanung ist der von WOHLFART & HAUSER [2007] vorgestellte Authoring-Prozess nicht geeignet, da er zu viel Handarbeit zur Erstellung nur einer einzigen Animation voraussetzt.

Mit dem ZOOMILLUSTRATOR stellten PREIM u. a. [1996] ein Framework vor, mit welchem einfache Text- und Bildinhalte präsentiert werden sollten. Dabei wurden Lehrbuchtexte passend zu aktuell dargestellten anatomischen 3D-Modellen angezeigt. Um verschiedene Aspekte einer Darstellung erläuternd zu präsentieren, konnte ein Autor Skripte verfassen, in denen er definierte, *was* dargestellt und *wie* es hervorgehoben werden sollte. Die Erstellung von erklärenden Animationen wurde jedoch nur an wenigen ausgewählten 3D-Modellen gezeigt. Eine Übertragung einer einmal generierten Animation auf neue 3D-Modelle war nicht möglich. Auch konnten Skripte aufgrund ihrer Komplexität ähnlich einer Programmiersprache nur von erfahrenen Programmierern erstellt werden.

Nicht weiter betrachtet werden in dieser Arbeit 4D-Animationen, die neben der dreidimensionalen Szene auch eine zeitliche Veränderung des Datensatzes beinhalten (wie z.B. RYAN u. a. [2004]) oder die Veränderungen an den Objekten der Szene selbst, wie Deformationen (z.B. VERNON & PECKHAM [2002]) oder Explosionsdarstellungen (z.B. LI u. a. [2008]). Deformationen sind wichtig für OP-Simulatoren, die jedoch seltener zur OP-Planung als vielmehr zum Training von Eingriffen zum Einsatz kommen. Explosionsdarstellungen schließen sich als Anwendung bei der chirurgischen Operationsplanung aus, da sie eine Veränderung der Lagebeziehungen der Strukturen untereinander zur Folge haben.

3.3 Ein skriptbasiertes Framework zur Animationsgenerierung

Als Vorarbeit der vorliegenden Dissertation wurde eine Skriptsprache zur abstrakten Beschreibung von Animationsvorgängen entwickelt. Diese Skriptsprache wurde im Rahmen dieser Arbeit in ein vollumfängliches Framework zur Animationsgenerierung überführt, welches den Sockel vieler weiterführender Bestandteile dieser Arbeit bildet, z.B. zur Sichtpunktbestimmung (Kapitel 4) oder zu Kamerafahrten (Kapitel 6). Ursprüngliches Ziel der Skriptsprache war es, für den Nutzer eines Planungssystems, zumeist einen Chirurgen, eine Möglichkeit bereitzustellen, eigene Animationen zu entwerfen und bei Bedarf wiederzuverwenden. Im Rückblick hat sich diese Motivation als irreführend herausgestellt. Chirurgen bevorzugen vielmehr einfache, visuelle Eingabeformen und lehnen das Schreiben von Skripten ab. Vielmehr konnte sich im bisherigen Nutzungszeitraum die Skriptsprache als wertvolle Hilfe für die Entwicklung verschiedenster Applikationen und darin enthaltener Animationsabläufe erweisen. Im Folgenden wird das Framework mit der speziell entwickelten Skriptsprache genauer vorgestellt.

3.3.1 Paradigmen

Wesentliches Ziel der Skriptsprache ist die Erzeugung **wiederverwendbarer** und **reproduzierbarer** Animationen für Fälle mit einer ähnlichen therapeutischen Fragestellung des Chirurgen. Daher wurde die Entwicklung der Skriptsprache von vier Paradigmen geleitet:

- **Adaptivität:** Die Lage vieler anatomischer Strukturen gehorcht gut bekannten Prinzipien, dennoch unterscheiden sie sich individuell in ihrer spezifischen Form und Größe. Solche Unterschiede müssen von einem Framework zur automatischen Animationsgenerierung berücksichtigt und die Animationen jeweils angepasst werden. Ebenfalls müssen die Unterschiede in den Pathologien, die in Auftreten, Größe und Lage vorher unbekannt sind, adaptiert werden.
- **Einfachheit und Spezialisierung:** Ausgehend von der Breite der medizinischen Anwendungsfelder muss die Skriptsprache zur Animationsgenerierung einen Ausgleich zwischen Einfachheit, Effizienz der Skripterstellung und der Spezialisierung der Animationen auf den jeweiligen Anwendungsfall abbilden.
- **Spezielle Animationstechniken für die Operationsplanung:** In Erweiterung klassischer Animationstechniken, wie Rotation, Zoom oder Kamerafahrten, muss es möglich sein, spezielle medizinische Visualisierungstechniken zu animieren. Beispiele dafür sind Clip-Ebenen, Volumerendering und schichtbildbasierte Visualisierungen.

- **Standardisierung:** Animationen für die Interventionsplanung sollten standardisiert sein, um einen Vergleich der Animationen zu ermöglichen und die Erkennung von gleichen Strukturen in unterschiedlichen Datensätzen zu erleichtern.

Diese vier Paradigmen fanden auch bei der Entwicklung weiterführenden Techniken, wie den Keystates, Anwendung und wurden dort teilweise verfeinert (Kapitel 8).

3.3.2 Konzept der Skriptsprache

Ein *Skript* dient der Beschreibung des Verhaltens aller Objekte einer Szene sowie der Kamera. Es ist nicht auf einen speziellen Datensatz zugeschnitten, um eine Wiederverwendung für viele ähnliche Datensätze zu erleichtern. Ein Skript besteht aus mehreren einzelnen *Skriptanweisungen*. Eine einzelne Skriptanweisung besteht dabei aus vier Teilen: *Zeit*, *Objektname*, *Befehl* und *Parametern*. Ein Skript wird auf einer Visualisierung ausgeführt, das heißt, alle *Befehle* werden über die *Zeit*¹¹ auf die assoziierten *Objekte* angewendet. Dies können Strukturen, wie Organe oder Gefäße, aber auch die Kamera oder die gesamte Szene sein. Die Reihenfolge der Anweisungen im Skript muss dabei nicht der Reihenfolge ihrer Ausführung entsprechen. Damit werden parallele und sich überlappende Anweisungen möglich. Die Beispielanweisung in Listing 3.1 bewegt eine Clip-Ebene von rechts durch die Szene und clippt dabei alle Strukturen außer den Gefäßen (Abb. 3.1).

```
[0,8] 'All' clipPlane on left 0.8 except 'Vessels'
```

Listing 3.1: Beispiel einer Skriptanweisung

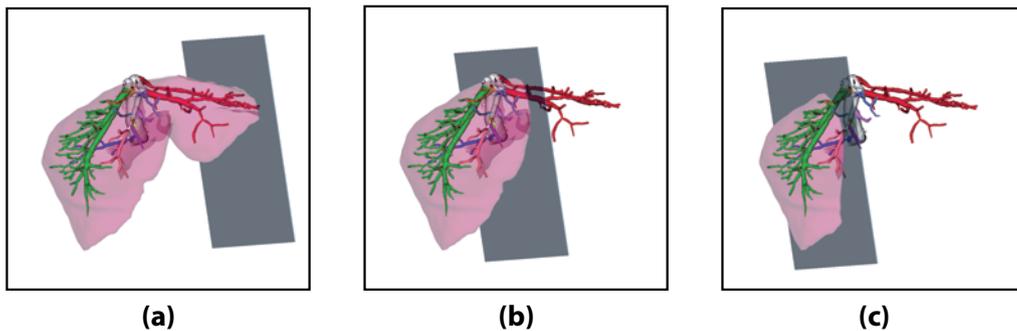


Abbildung 3.1: Beispielframes für die Skriptanweisung in Listing 3.1. Eine Clip-Ebene wird von links durch einen Leberdatensatz bewegt.

¹¹Als Zeit kann dabei ein Zeitraum wie [0, 10] für die ersten 10 Zeiteinheiten oder ein einzelner Zeitpunkt wie [5] fungieren.

Abstraktion von Skriptanweisungen

Zur Vereinfachung der Skriptsprache wurden verschiedene Ebenen der Abstraktion von Skriptanweisungen eingeführt. High-Level-Anweisungen werden dabei unter Anwendung spezifischer Dekompositionsregeln auf Low-Level-Anweisungen abgebildet. Low-Level-Anweisungen sind elementare, direkt ausführbare Anweisungen, vergleichbar dem Maschinencode bei der Programmierung eines Rechners. Solche Anweisungen sind zum Beispiel `move`, `rotate` oder `setColor`. High-Level-Anweisungen sind beispielsweise `setAsFocusObject`, `emphasize` oder `groupOverview`. Durch die Abstraktion von Anweisungen wird eine Vereinfachung der zu schreibenden Skripte erreicht. Komplette Animationen lassen sich so durch wenige Anweisungen beschreiben. Zur feingranularen Steuerung von Animationen können High-Level-Anweisungen beliebig mit Low-Level-Anweisungen kombiniert werden.

Durch die Dekompositionsregeln wird bestimmt, in welche anderen Anweisungen eine High-Level-Anweisung überführt wird. Nach oben hin sind die Anweisungen beliebig abstrahierbar, das heißt, eine High-Level-Anweisung kann wiederum auf andere High-Level-Anweisungen abgebildet werden. Die Auswahl der jeweiligen Dekompositionsregeln basiert auf einer Objektklassifikation (z.B. Muskeln, Gefäße oder Knochen), die den Segmentierungsergebnissen entnommen wird. So kann die High-Level-Anweisung `emphasize` auf die Low-Level-Anweisung `setColor` sowie weitere High-Level-Anweisungen zur Kamerapositionierung abgebildet werden.

Die Umwandlung der Anweisungen entsprechend der Dekompositionsregeln erfolgt in einem Parsing-Schritt direkt vor der Ausführung eines Skriptes (Abb. 3.2). Die Abstraktion erstreckt sich neben den Anweisungen auch auf die Objekte und Parameter. So können Strukturen zu Gruppen zusammengefasst werden und eine Anweisung direkt auf mehrere Strukturen gleichzeitig angewendet werden (siehe Listing 3.2). Parameter können durch das Konzept in eine einfacher lesbare und handhabbarere Form übertragen werden.

Adaptivität

Durch die Definition abstrakter Strukturgruppen kann das Skript im Parsing-Schritt vor der Ausführung einer Animation direkt an die im Datensatz real vorhandenen Strukturen angepasst werden. Es wird also dynamisch entschieden, ob Anweisungen für 'Lymphnode' auf 10 oder 13 Lymphknoten angewendet werden müssen.

Daneben ist die Adaptivität der Kamera entscheidend für eine automatische Animationsgenerierung. Die Position einer Kamera kann im Vorfeld bei der Skriptzeugung nie absolut positioniert werden, da diese Position bei jedem Datensatz an einer völlig anderen Stelle liegen kann. Die Kamera wird daher in Relation zu einzelnen Strukturen oder Strukturgruppen positioniert (Listing 3.3). Dabei

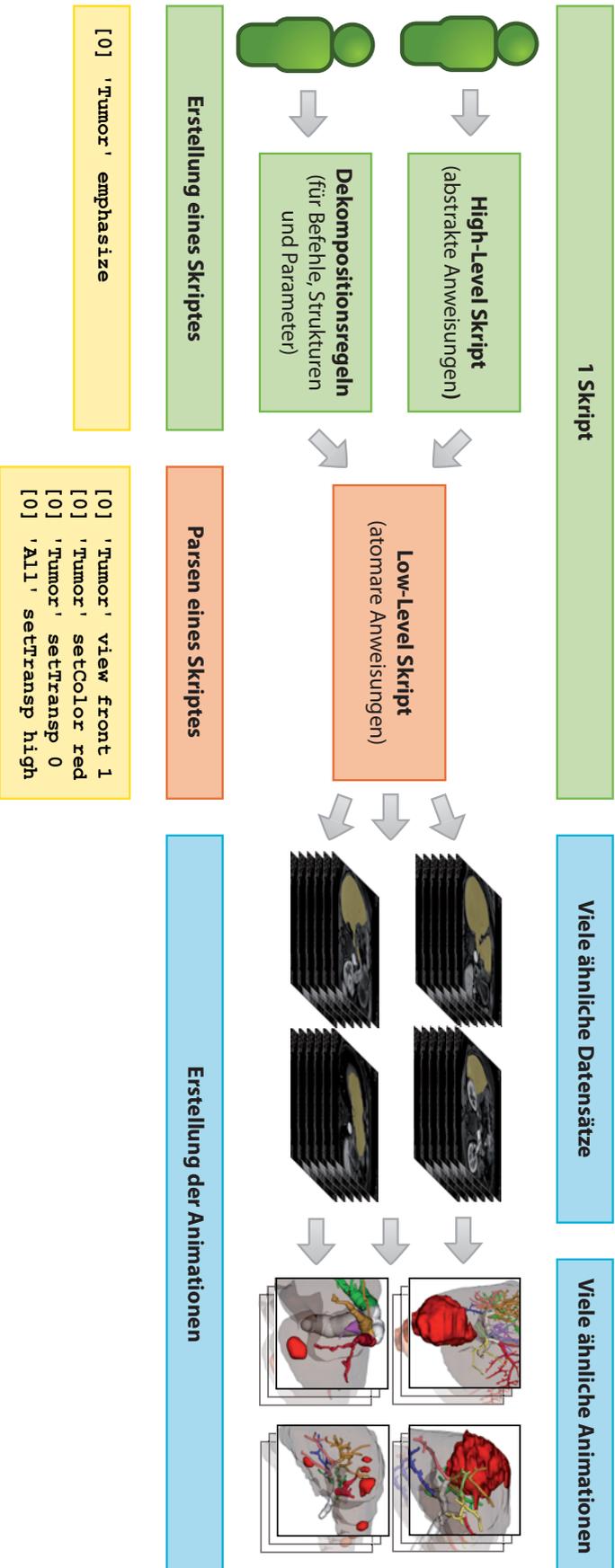


Abbildung 3.2: Konzept der Umwandlung eines High-Level-Skriptes in ein Low-Level-Skript unter Verwendung von Dekompositionsregeln für Anweisungen, Objekte und Parameter. Das Low-Level-Skript enthält direkt ausführbare Anweisungen, auf deren Basis dann die Animation generiert wird.

```
[0,10] 'Vessel' setColor red
...
[0,10] 'Vein' setColor 255,0,0
[0,10] 'Artery' setColor 255,0,0
```

Listing 3.2: Dekomposition der Strukturgruppe `Vessel` in `Vein` und `Artery` sowie des Parameters `red` in dessen RGB-Wert.

werden diese durch ihre jeweiligen `BoundingBoxen` approximiert. Das ist für die meisten Strukturen ausreichend, da diese eine zumeist kompakte Form aufweisen. Bei elongierten Strukturen kann diese Vorgehensweise jedoch zu Problemen führen.

Die Angabe der Positionierung der Kamera erfolgt in Polarkoordinaten auf einer umgebenden Kugel. Zusätzlich wird ein `Zoomfaktor` angegeben, der bei einem Wert von 1 die Struktur vollständig im Viewport der Kamera abbildet und bei kleineren Werten die Struktur größer, bei größeren Werten entsprechend kleiner, darstellt.

```
[0,10] 'Cam' view 'Tumor' front normalZoom
...
[0,10] 'Cam' view 'Tumor' 90 0 1
```

Listing 3.3: Anweisung zur Positionierung der Kamera in Relation zum Tumor in einem Datensatz. Die Kamera soll frontal auf den Tumor blicken und den Tumor vollständig zeigen (`normalZoom`). Die Parameter werden dabei auf Low-Level-Ebene in Polarkoordinaten auf einer die Szene umgebenden Kugel übersetzt.

3.3.3 Beispiele für Animationen

Im Folgenden werden zwei Beispiele für Animationen auf der Basis von Animationskripten vorgestellt. Das erste Beispiel entstammt der Planung von Halslymphknotenausräumungen, wie sie im `NECKSURGERYPLANNER` Anwendung finden (TJETJEN u. a. [2006a]). Zunächst wird die Szene kurz eingeführt (u.a. durch eine Rotation um alle Strukturen). Anschließend werden die Lymphknoten entsprechend ihrer Größe hervorgehoben, um so dem Chirurg einen ersten schnellen Überblick über die Lage, gerade der vergrößerten Lymphknoten zu geben (Listing 3.4 und Abb. 3.3).

Das zweite Beispiel (Listing 3.5 und Abb. 3.4) zeigt die Einführung einer Szene in einem Lernprogramm für Leberchirurgen, dem `LIVERSURGERYTRAINER` (Kapitel 10). Dabei wird zunächst die Szene in einer Standardfärbung dargestellt und anschließend der Tumor hervorgehoben und um diesen rotiert. Abschließend werden die portalveösen Territorien angezeigt, wobei die vom Tumor betroffenen nur als Silhouette dargestellt werden.

```
// Rotation der Szene / Anwendung der Standardparameter
// für Farbe und Transparenz aller Strukturen
[1,7] 'Scene' introduce

// Hervorhebung der Lymphknoten entsprechend ihrer Größe
[8,10] 'Lymphnode' showBySize

// Zoom zum größten Lymphknoten der rechten Halsseite
[11,14] 'Cam' showLK largest right
```

Listing 3.4: Ein Animationskript zur Hervorhebung vergrößerter Halslymphknoten. Dazu wird zunächst ein Überblick über die Szene gegeben. Anschließend werden die Lymphknoten entsprechend ihrer Größe unterschiedlich stark hervorgehoben. Zum Schluß zoomt die Kamera auf den größten Lymphknoten auf der rechten Halsseite.

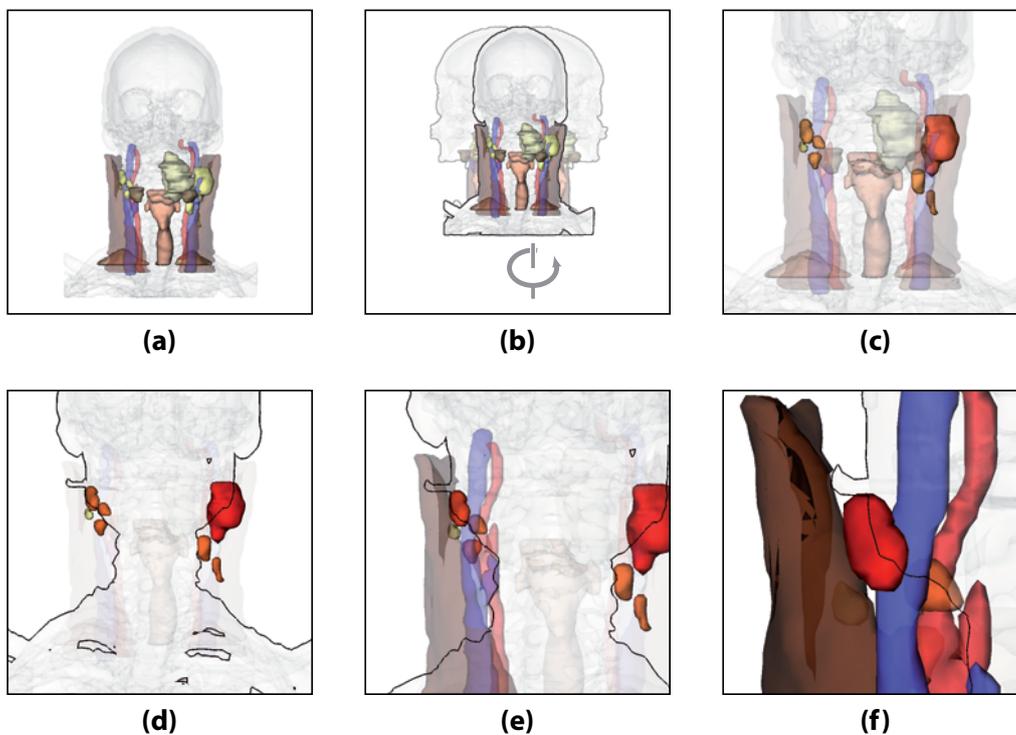


Abbildung 3.3: Frames einer Animation, die entsprechend des Skriptes in Listing 3.4 zur Hervorhebung vergrößerter Halslymphknoten erzeugt wurde.

```
[0] 'Scene' init
[1,4] 'Tumor' emphasize
[5,8] 'Tumor' rotate axial 360
[9,12] 'PVTerritories' show infiltration
```

Listing 3.5: Skript zur Einführung einer Szene im LiverSurgeryTrainer. Die Szene wird initialisiert, anschließend wird der Tumor hervorgehoben und die Kamera um ihn rotiert. Abschließend werden die portalvenösen Territorien gezeigt.

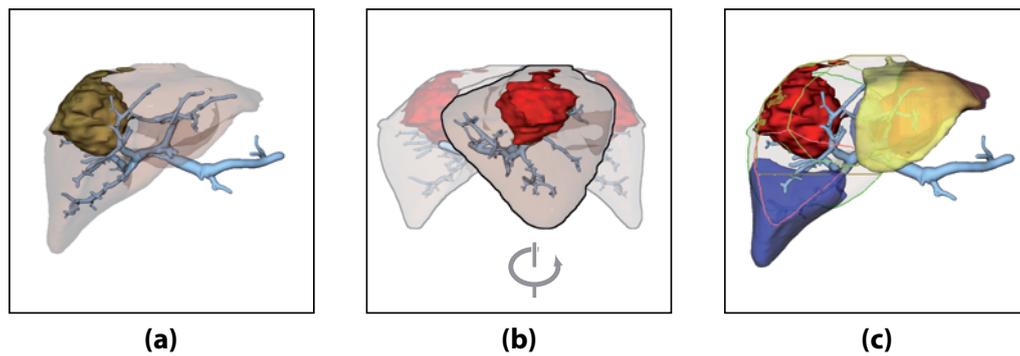


Abbildung 3.4: Frames einer Animation, die entsprechend des Skriptes in Listing 3.5 zur Einführung einer Szene im LiverSurgeryTrainer erzeugt wurde.

3.4 Zusammenfassung und Analyse des Frameworks

Durch die stetige Evaluierung des skriptbasierten Animationsframeworks konnten Erfahrungen gesammelt und verschiedene Aspekte direkt abgeleitet werden, die in Entwicklungen neuer Techniken einfließen, denen sich die vorliegende Arbeit ausführlich widmen wird. Dazu gehört die Erkenntnis, dass eine Platzierung der Kamera allein durch abstrakte Richtungsvorgaben („von vorn“, „von links“) für eine automatische Animationsgenerierung nicht ausreichend ist. Zusätzlich ist die Ermittlung möglichst günstiger Sichtpunkte auf eine Struktur nötig (Kapitel 4). Weiterhin hat sich der Glaube, Ärzte würden selbstständig Skripte zur Animationsgenerierung schreiben, als Trugschluss erwiesen. Daher wurde mit den *Keystates* ein völlig neues Konzept entwickelt, um durch eine zusätzliche Abstraktionsschicht, es dem Arzt auf intuitive und visuelle Art und Weise zu ermöglichen, Animationen selbsttätig zu definieren und zu erzeugen (siehe Kapitel 8).

Als ein zusätzliches positives Ergebnis der ersten Arbeiten am skriptbasierten Animationsframework lässt sich die Verwendung der Skriptsprache zur **standardisierten Generierung von Standbildern** festhalten. Durch das automatische Einblenden der jeweiligen Strukturen in den passenden Farben und Transparenzen und der Wahl eines einheitlichen Sichtpunktes können vergleichbare Bilder eines Patienten (z.B. vor und nach einer Operation) sowie vieler verschiedener Patienten ohne Mehraufwand erzeugt werden. Dies hat sowohl die Arbeit im Forschungskontext erheblich beschleunigt, als auch Eingang in Planungssysteme in der klinischen Routine gefunden.

Zusammenfassend ist festzustellen, dass in diesem Kapitel die Bedeutung von Animationen sowohl für die interaktive Exploration als auch für Videos zur Präsentation und Dokumentation analysiert wurde. Das skriptbasierte Framework zur automatischen Animationsgenerierung bildet die Grundlage vieler weiterer Techniken, die in den folgenden Kapiteln vorgestellt werden.

4

Automatische Bestimmung von Sichtpunkten

Ein wesentlicher Aspekt bei der Erstellung von Darstellungen ist der gewählte Sichtpunkt. Ist er ungünstig gewählt, kann der Betrachter möglicherweise nicht die gewünschten Informationen aus der Darstellung entnehmen. In extremen Fällen kann es sogar dazu kommen, dass der Betrachter falsche Schlüsse zieht, weil vom aktuellen Sichtpunkt aus wichtige Informationen verdeckt sind. Ohne einen „guten“ Sichtpunkt auf eine Szene können auch noch so ausgefeilte Visualisierungstechniken ihr Potential nicht entfalten. Gute Sichtpunkte manuell im Rahmen einer interaktiven Exploration zu erzeugen, ist nicht nur für ungeübte Nutzer, die nicht täglich längere Zeit mit 3D-Darstellungen arbeiten, ein aufwendiger und energieverbrauchender Prozess.

So haben eigene Untersuchungen gezeigt, dass es für einen HNO-Chirurgen einigen Aufwandes bedarf, die virtuelle Kamera genau an der Stelle zu positionieren, von welcher aus er einen unverdeckten Blick auf den minimalen Abstand eines Lymphknotens und eines Muskels hat. Für einen Leberchirurgen bringt es z.B. zeitliche Vorteile, wenn er nicht lange nach dem einen Sichtpunkt suchen muss, von dem aus ein intra-hepatischer Tumor nicht vollständig von Gefäßästen verdeckt ist. Die automatische Bestimmung guter Sichtpunkte ist weiterhin für Verfahren unerlässlich, bei denen keine Interaktionen des Nutzers erwünscht sind – z.B. bei der automatischen Erstellung von Animationen oder in intra-operativen Szenarien.

Im Rahmen dieser Arbeit wurde daher ein neuartiges Verfahren entwickelt, das die automatische Bestimmung guter Sichtpunkte auf eine (medizinische) Darstellung erlaubt.

Gliederung des Kapitels

Zunächst werden Anforderungen an ein Verfahren zur automatischen Sichtpunktbestimmung für medizinische Visualisierungen erläutert und eine Klassifizierung von Sichtpunkten wird vorgenommen (Abschnitt 4.1). Anschließend werden verwandte Arbeiten analysiert (Abschnitt 4.2) und Grundlagen der virtuellen Kamerasteuerung vorgestellt (Abschnitt 4.3), bevor im Kern des Kapitels das neu entwickelte Verfahren zur automatischen Sichtpunktbestimmung vorgestellt wird (Abschnitt 4.4). Abschließend wird eine Evaluierung präsentiert (Abschnitt 4.5) und in einem Ausblick wird ein Blick auf mögliche zukünftige Entwicklungen gegeben (Abschnitt 4.6).

Verschiedene Ansätze zur Anwendung des Verfahrens werden später in Kapitel 5 vorgestellt.

Eigene Publikationen

Dieses Kapitel basiert u.a. auf folgenden eigenen Publikationen:

MÜHLER u. a. [2007b] Mühler, K.; Neugebauer, M.; Tietjen, C. & Preim, B. *Viewpoint Selection for Intervention Planning*. In: **Proc. of IEEE/Eurographics Symposium on Visualization (EuroVis)**, 2007

MÜHLER u. a. [2008a] Mühler, K.; Hansen, C.; Neugebauer, M. & Preim, B. *Automatische Kamerapositionierung für intra-operative Visualisierungen in der onkologischen Leberchirurgie*. In: **Proc. of Bildverarbeitung für die Medizin (BVM)**, 2008

NEUGEBAUER u. a. [2007] Neugebauer, M.; Mühler, K.; Tietjen, C. & Preim, B. *Automatische Kamerapositionierung in komplexen medizinischen 3D-Visualisierungen*. In: **Proc. of Bildverarbeitung für die Medizin (BVM)**, 2007

4.1 Anforderungsanalyse und Einordnung

Die automatische Bestimmung guter Sichtpunkte ist im Rahmen dieser Arbeit unter drei Gesichtspunkten von Bedeutung:

1. Bei der **automatischen Erstellung von Animationen** ist die Ermittlung von Sichtpunkten auf die Darstellung essenziell, sobald es zu Bewegungen der Kamera kommt. Die Ermittlung muss automatisch erfolgen und sollte gute Resultate liefern. Die Frage „*Was ist ein guter Sichtpunkt?*“ ist dabei von zentraler Bedeutung und wird im Laufe dieses Kapitels ausführlich diskutiert werden.
2. Die **individuelle Exploration dreidimensionaler Darstellungen** ist ein Kernbestandteil der chirurgischen Operationsplanung, jedoch auch ein sehr aufwendiger. Gerade die Manipulation der virtuellen Kamera ist nicht

nur für ungeübte Benutzer aufwendig und enervierend. Dies geht sowohl aus vielen Gesprächen mit Anwendern, Beobachtungen wie auch aus Studien (BADE u. a. [2005]) hervor. Daher birgt an dieser Stelle eine Unterstützung des Benutzers ein großes Potential. Ein Aspekt einer solchen Unterstützung ist die automatische Bestimmung von guten Sichtpunkten auf Teile der Darstellung. Solche Sichtpunkte können ein guter Ausgangspunkt für weitere Explorationen des Benutzers sein, welcher davon ausgehend oft nur noch kleine Änderungen an der Darstellung vornehmen muss, um zur gewünschten Visualisierung zu gelangen.

3. In einer Umgebung wie dem **Operationssaal**, in der nur eingeschränkte Interaktionen möglich sind, stellt eine automatische Bestimmung von Sichtpunkten auf verschiedene Strukturen in den Planungsdaten eine wesentliche Erleichterung dar und erlaubt dem Operateur die Fokussierung auf die eigentliche Beantwortung seiner Fragestellung an die Visualisierung.

Medizinische Darstellungen, wie sie im Rahmen dieser Arbeit verwendet werden (siehe Abschnitt 2.3), basieren auf polygonalen Repräsentationen einer Vielzahl von Strukturen. Diese Darstellungen werden in den hier genutzten Szenarien (Abschnitt 2.1 und 2.2) von außen betrachtet, sind also kompakt. Der Betrachter einer solchen Darstellung (z.B. der planende Chirurg) hat je nach (medizinischer) Fragestellung verschiedene Anforderungen an einen Sichtpunkt:

1. Der Betrachter möchte **die Darstellung als Ganzes** betrachten, beispielsweise um sich einen Überblick über die vorhandenen Strukturen zu verschaffen.
2. Der Betrachter möchte **eine Struktur** genau ansehen und ist z.B. an speziellen Oberflächen- oder Formmerkmalen interessiert – Tumore, Lymphknoten oder Rundherde wären einige Beispiele dafür.
3. Der Betrachter möchte **eine Struktur im Kontext anderer Strukturen** betrachten. Kontextstrukturen können z.B. größere benachbarte Strukturen wie Knochen sein, die zwar abgeschwächt dargestellt werden, aber wichtige Hinweise über die Lage der betrachteten Struktur geben können.
4. Der Betrachter möchte **zwei oder mehrere Strukturen und ihre räumliche Relation** betrachten. Dies kann beispielsweise der minimale Abstand zweier Strukturen sein.

Je nach Fragestellung des Betrachters sind andere Kriterien zur Bewertung eines Sichtpunktes zu wählen. Die qualitative Bewertung eines Sichtpunktes ist unerlässlich, so er denn automatisch bestimmt werden soll. Diskussionen mit Medizinern und Beobachtungen von Anwendern lassen folgende Kriterien für eine Bewertung von Sichtpunkten auf medizinische Darstellungen als sinnvoll erscheinen:

- Die Zielstruktur sollte, soweit möglich, **unverdeckt** von anderen Strukturen sichtbar sein (Abb. 4.1(a)). Dies gilt auch für Gruppen von Strukturen, wie mehrere Lymphknoten oder Lebersegmente.
- Ist eine Verdeckung unvermeidlich, z.B. weil die Zielstruktur komplett von anderen Strukturen umschlossen ist, so sollte sie aus der Sichtrichtung von für den Kontext möglichst unwichtigen Strukturen verdeckt sein (Abb. 4.1(b)). Diese können unter Umständen ausgeblendet werden.
- Die Zielstruktur sollte möglichst von einer Seite betrachtet werden, die ein **hohes Maß an Informationen** bietet. Dies kann eine große Fläche sein (Abb. 4.1(c)), aber auch andere Features, wie eine starke Krümmung oder eine bestimmte Form, können für den Betrachter interessant sein.
- Gerade im medizinischen Umfeld hat sich gezeigt, dass es **Vorzugsblickrichtungen** gibt. So möchte ein HNO-Chirurg Strukturen des Halses aus einer möglichst äquatorialen Sichtrichtung betrachten und niemals von oben oder unten (Abb. 4.1(d)). Es ist zu vermuten, dass es solche Vorzugsblickrichtungen auch für viele Bereiche außerhalb der Medizin gibt.
- Die Zielstruktur sollte möglichst **zentral** im Bildraum liegen (Abb. 4.1(e)) und sowohl nah am Betrachter als auch nah zum vorhergehenden Sichtpunkt. Letzteres vermeidet unnötig lange Kameraflüge zwischen einzelnen Sichtpunkten.
- Da automatisch berechnete Sichtpunkte bei der Exploration Ausgangspunkte für weitere Veränderungen der Kamera durch den Betrachter sein können, sollten die Sichtpunkte **stabil** sein. Das heißt, kleine Änderungen des Sichtpunktes sollten keine großen Auswirkungen auf die Sichtbarkeit der Zielstruktur haben (Abb. 4.1(f)).

Ein Verfahren zur automatischen Bestimmung von Sichtpunkten auf medizinische Darstellungen muss diese Kriterien berücksichtigen. Neben ihrer reinen Berechnung ist auch die Abhängigkeit dieser Kriterien von Anwendungen und medizinischen Fragestellungen zu beachten: Für einen HNO-Chirurgen sind Vorzugsblickrichtungen wichtig, während ein Leberchirurg durch den umgebenden Charakter der Leber es immer mit Verdeckungen seiner Zielstrukturen zu tun hat. Weiterhin haben selbst innerhalb einer Anwendungsdomäne verschiedene Fragestellungen unterschiedliche Auswirkungen auf die Wichtigkeit von Kontextstrukturen. So können je nach Fragestellung und Anwendung völlig unterschiedliche Sichtpunkte als „gut“ bewertet werden.

Anhand der allein im Bereich der medizinischen Visualisierung auftretenden Unterschiede bei der Bewertung der Güte eines Sichtpunktes ist die Hinzunahme von domänenspezifischem Wissen zur Bewertung von Sichtpunkten unerlässlich. Die Analyse verwandter Arbeiten in Abschnitt 4.3 wird zeigen, dass diese Einsicht nicht selbstverständlich ist. Das nötige domänenspezifische Wissen kann im Fall der vorliegenden Visualisierungen zur Interventionsplanung implizit aus den

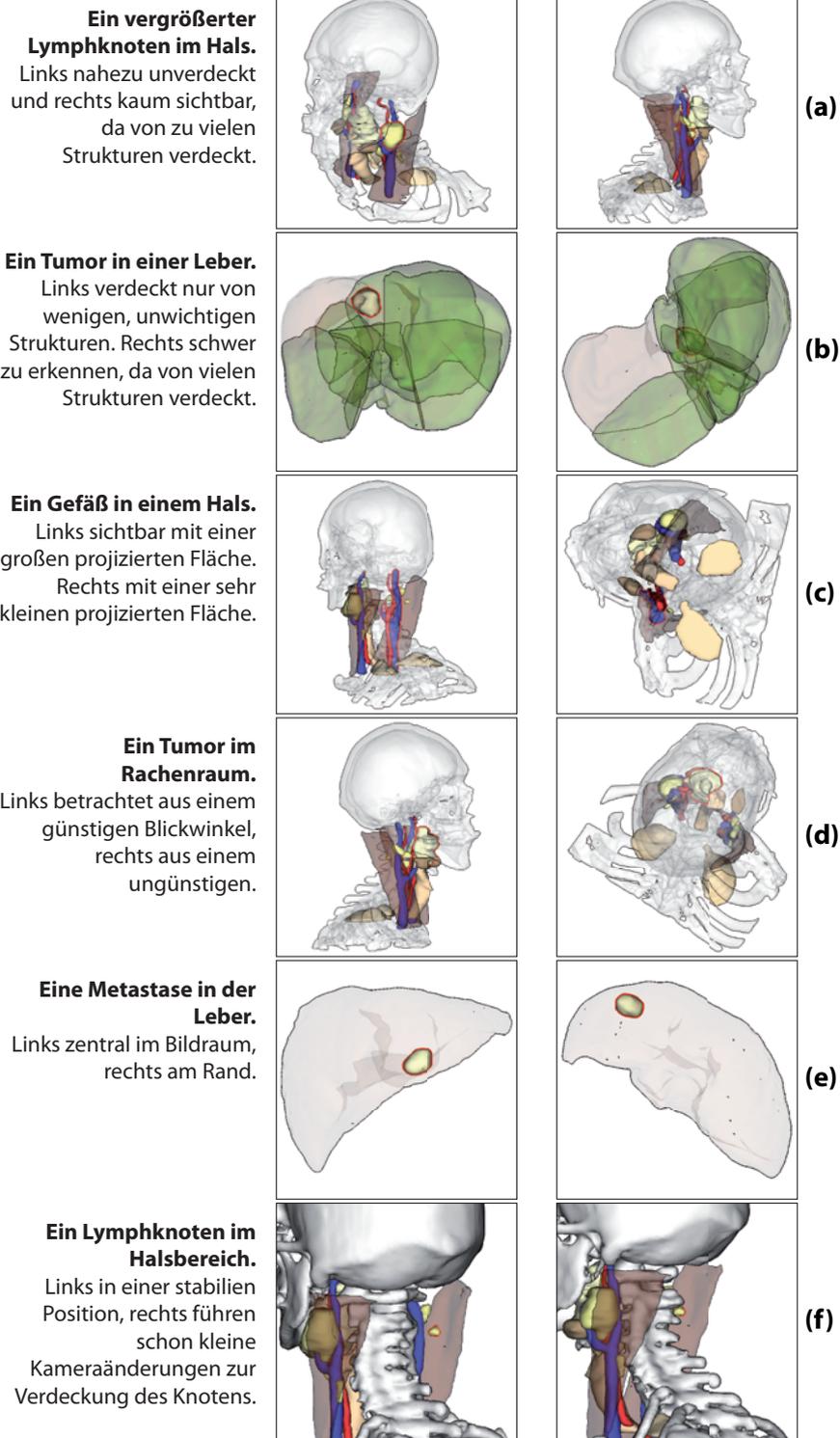


Abbildung 4.1: Vergleich von guten (links) und schlechten (rechts) Sichtpunkten auf verschiedene Strukturen von Interesse (rot umrandet). Es wurde jeweils genau eine Eigenschaft des Sichtpunktes berücksichtigt: (a) Verdeckung; (b) Wichtigkeit verdeckender Strukturen; (c) Hohes Maß an Information (hier: große Fläche); (d) Vorzugsblickrichtung; (e) Zentralität; (f) Stabilität;

vorhandenen Segmentierungen gezogen werden. Es erschwert aber auch die Anwendung automatischer Verfahren zur Sichtpunktbestimmung in Bereichen, in denen diese kontextuellen Informationen fehlen – so z.B. in der Diagnose, bei der meist auf direkten Volumendarstellungen gearbeitet wird.

Zusammenfassend lässt sich feststellen, dass ein Verfahren benötigt wird, welches flexibel an verschiedene Anwendungsszenarien und Fragestellungen anpassbar ist. Eine solche Initialisierung könnte ähnlich einem Template mit verschiedenen Sets von Parametern erfolgen. Die Datengrundlage eines solchen Verfahrens müsste jedoch so allgemein sein, dass ein Wechsel der Initialisierung (oder des Templates) auch zur Laufzeit möglich wäre, um unmittelbar auf Nutzeranfragen reagieren zu können.

4.2 Grundlagen der Kamerasteuerung

Zur besseren Einordnung und zum Verständnis des in diesem Kapitel später vorgestellten Verfahrens werden im Folgenden gängige Metaphern zur Kamerasteuerung in 3D-Darstellungen kurz erläutert, die sich im Laufe der Zeit etabliert haben:

- a) „Eyeball in Hand“-Metapher
- b) „Scene in Hand“-Metapher
- c) „Flying Vehicle Control“-Metapher

Die Metaphern unterscheiden sich in der Art und Weise, wie die Freiheitsgrade der Eingabegeräte auf die Freiheitsgrade der Kamera abgebildet werden. Sieht man von der Möglichkeit einer Kamera ab, einen Öffnungswinkel sowie eine near- bzw. far-Clipping-Plane zu haben, so gibt es sechs Freiheitsgrade (drei Positionsvektoren und drei Rotationsvektoren). Am geeignetsten erscheinen für die Aufgabe der Kamerasteuerung daher direkte Eingabegeräte, wie das PHANTOM der Firma Sensable mit ebenfalls sechs Freiheitsgraden. So können Position und Rotation der Kamera direkt manipuliert werden. Allerdings ist dabei ein haptisches Feedback unverzichtbar (vergleiche Evaluierung von [WARE & OSBORNE \[1990\]](#)). Die „**Eyeball in Hand**“-Metapher beschreibt eine solche Kamerasteuerung, bei der der Nutzer quasi ein Auge direkt manipulativ in der Szene bewegen kann. Ohne ein Eingabegerät mit sechs Freiheitsgraden und haptischem Feedback ist eine „Eyeball in Hand“-Steuerung jedoch nur schwer umzusetzen.

Eine Alternative stellt daher die „**Scene in Hand**“-Metapher dar. Bei dieser wird die Kamera auf einer die Szene umgebenden Kugel um den Szenenmittelpunkt rotiert, wobei die Blickrichtung jederzeit auf diesen gerichtet ist. Da für eine solche Steuerung nur zwei Freiheitsgrade nötig sind (die Polarkoordinaten der Kamera auf der Kugeloberfläche), bietet sich hierfür die Steuerung mit einer Maus an. Es gibt allerdings verschiedene Modelle, wie eine solche Rotation der Kamera um

die Szene mit der Maus gesteuert wird (z.B. *Bell's Virtual Trackball* oder den *ArcBall* SHOEMAKE [1992]). Jedes dieser Modelle bietet gewisse Vor- und Nachteile. Eine umfassende Evaluierung von BADE u. a. [2005] zeigte, dass je nach Erfahrungsgrad der Nutzer unterschiedliche Techniken in einer anderen Performanz bei der Szenenexploration mündeten und dass auch bei der individuellen Bewertung der Techniken größere Unterschiede auftraten. Dennoch scheint die Kamerasteuerung über die „Scene in Hand“-Metapher für kompakte Visualisierungen, wie sie auch in dieser Arbeit verwendet werden, die sinnvollste zu sein.

Ein wenig anders stellt sich die Kamerasteuerung bei der Exploration elongierter Strukturen, wie Gefäßen oder bei endoskopischen Visualisierungen, dar. Hier liegt der Einsatz der „**Flying Vehicle Control**“-Metapher nahe. Dabei wird die Kamera ähnlich einem Flugzeug gesteuert: Sie hat eine Vortriebsgeschwindigkeit und der Nutzer steuert über Winkeländerungen die Flugrichtung sowie die Geschwindigkeit. Eine Steuerung der Richtung kann gänzlich entfallen, wenn die Kamera einem vorgegebenen Pfad folgt (z.B. der Mittellinie eines Gefäßes oder der Luftröhre). Dann kann die Eingabe des Nutzers zur Blickwinkelsteuerung verwendet werden: Der Nutzer kann zu den Seiten blicken, ohne den vorgegebenen Pfad verlassen zu müssen. Auch wenn solch eine Steuerung mit der Maus als Eingabegerät möglich ist, haben Untersuchungen gezeigt, dass eine Nachempfindung des realen endoskopischen Eingriffs mittels eines Gerätes mit haptischem Feedback wie dem PHANTOM für Chirurgen angenehmer ist (KRÜGER u. a. [2008]).

4.3 Verwandte Arbeiten

Die Forschung auf dem Gebiet der (automatischen) Steuerung der Kamera in 3D-Darstellungen ist ein weites Feld. CHRISTIE u. a. [2008] gaben dazu einen sehr guten und ausführlichen Überblick. Ein wichtiger Aspekt der Kamerasteuerung ist die Bestimmung guter Sichtpunkte und damit verbunden die Bestimmung von Sichtbarkeiten. ELMQVIST & TSIGAS [2008] haben daher 50 verschiedene Verfahren und Strategien – darunter das im Rahmen dieser Arbeit entstandene Verfahren (MÜHLER u. a. [2007b]) – zum Umgang mit Sichtbarkeiten und Verdeckungen einer genaueren Analyse unterzogen und eine Taxonomie erstellt. Sie konnten dabei die Verfahren nach ihren Eigenschaften in fünf Gruppen aufteilen:

1. Mehrfache Sichten (*Multiple Viewports*)
2. Röntgeneffekte (*Virtual X-Ray*)
3. Tourenplaner (*Tour Planner*)
4. Volumetrische Werkzeuge (*Volumetric Probe*)
5. Projektionsverzerrungen (*Projection Distorter*)

Eigenschaften, nach denen die Verfahren analysiert wurden, waren u.a.:

- „Wird die 3D-Darstellung erhalten oder verändert?“
- „Muss die Kamera bewegt werden, wenn ja, geschieht dies automatisch oder manuell?“
- „Wie viele Viewer werden verwendet?“
- „Wie werden Effekte zur Tiefenwahrnehmung und zur Erfassung räumlicher Zusammenhänge bewahrt?“

Für den Bereich der medizinischen Visualisierung sind im Wesentlichen Verfahren interessant, die den Gruppen „Tourenplaner“ und „Röntgeneffekte“ zugeordnet werden können. Verfahren, die sich einer räumlichen Veränderung bedienen (sei es durch aktiven Eingriff, wie bei den „Volumetrischen Werkzeugen“, oder durch Projektionsverzerrungen), scheiden für eine medizinische Anwendung in den meisten Fällen aus,¹² da die reale Lage und Relation aller Strukturen zueinander zu jeder Zeit gewahrt sein müssen. Die Nutzung mehrere Viewer zielt mehr in den Bereich der Anwendungsentwicklung und Oberflächenkonzeption und soll daher hier nicht betrachtet werden. Verfahren zur Röntgentechnik, bei denen Strukturen durch intelligentes Ausblenden verdeckender Teile der Darstellung sichtbar gemacht werden, kommen in der medizinischen Visualisierung öfter zur Anwendung, z.B. die Cut-Away-Technik (TIETJEN u. a. [2009]), sind jedoch ebenfalls nicht Teil der hier betrachteten Verfahren. Vielmehr lassen sich die Verfahren zur automatischen Bestimmung guter Sichtpunkte aus Sicht der medizinischen Visualisierung in zwei Bereiche einteilen:

- a) Verfahren für polygonale Darstellungen (Abschnitt 4.3.1) und
- b) Verfahren für direkte Volumendarstellungen (Abschnitt 4.3.2).

Aufgrund der Bedeutung für das im Rahmen dieser Arbeit entwickelte Verfahren werden in Abschnitt 4.3.3 Verfahren zur Bestimmung der Sichtbarkeit von Strukturen in polygonalen Szenen gesondert betrachtet. Verschiedene Anwendungen von Verfahren zur Sichtpunktbestimmung werden in Abschnitt 4.3.4 vorgestellt.

4.3.1 Sichtpunktbestimmung in polygonalen Darstellungen

Bei der folgenden Diskussion von Verfahren zur Sichtpunktbestimmung in polygonalen Darstellungen lag der Fokus auf der Frage, welches Maß genutzt werden kann, um die Güte eines Sichtpunktes zu bestimmen.

KHAWAJA u. a. [1996] präsentierten einen Ansatz zur Bestimmung guter Sichtpunkte auf Bauteilanordnungen in CAD-Systemen. Sie berechneten einen guten Sichtpunkt in zwei Schritten: Zunächst wurden Sichtpunkte ermittelt, von denen

¹²Neuere Entwicklungen, wie die *FlatMaps* von NEUGEBAUER u. a. [2009], stellen bisher eine Ausnahme für sehr spezielle Anwendungsfälle dar.

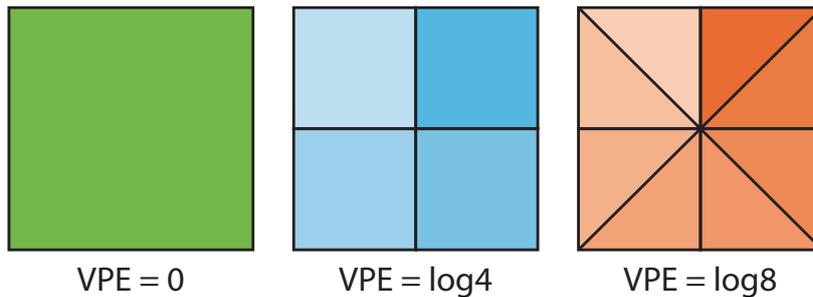


Abbildung 4.2: Beispiel für die Abhängigkeit der Viewpoint Entropy (VPE) von der Triangulierung. Obwohl jeweils die gleiche Fläche zu sehen ist, erhält man für die VPE (Gleichung 4.1) je nach Triangulierung unterschiedliche Werte.

aus die Abstände zwischen den Bauteilen gut zu sehen waren. Das waren Sichtpunkte, die möglichst senkrecht auf die minimalen Distanzen zwischen den Bauteilen blickten. Anschließend wurden die Sichtpunkte hinsichtlich der Anzahl der sichtbaren Objekte, der Polygone pro Objekt und der projizierten Fläche pro Polygon betrachtet. Dabei wurden Sichten mit vielen Objekten und großen Flächen bevorzugt. Unklar blieb, wie bei einer völligen Verdeckung der Sicht, z.B. durch umschließende Objekte, verfahren wurde.

Als Maß für die Güte eines Sichtpunkte in polygonalen Darstellungen führten VÁZQUEZ u. a. [2001] in Anlehnung an die Shannon-Entropy¹³ die *Viewpoint Entropy* (VPE) ein. Sie definieren die Viewpoint Entropy H_v als

$$H_v = - \sum_{i=0}^{N_p} \frac{a_i}{a_t} \log \frac{a_i}{a_t} \quad (4.1)$$

wobei N_p die Anzahl aller Polygone einer Szene ist, a_i die auf eine umgebende Kugel projizierte Fläche des Polygons i aus Sicht des Sichtpunkte v , a_o ist die projizierte Fläche des Hintergrundes und a_t die Gesamtfläche der umgebenden Kugel. Die Viewpoint Entropy ist abhängig von verschiedenen Triangulierungen einer Szene. So bevorzugt sie Sichten mit einer sehr feinen Triangulierung (Abb. 4.2). Eine solche Abhängigkeit macht das Verfahren für die praktische Nutzung unbrauchbar – eine Sichtpunktbestimmung sollte immer unabhängig von der Art und Qualität der polygonalen Netze sein.

Daher führten SBERT u. a. [2005] ein neues Maß basierend auf der Kullback-Leibler-Divergenz (COVER & THOMAS [1991]), die *Viewpoint Kullback-Leibler Distance* KL_v , ein. Dieses Distanzmaß wird zwischen der normalisierten Verteilung aller projizierten Flächen von allen Sichtpunkten aus und der aktuellen projizierten Fläche berechnet:

¹³Die Shannon-Entropy ist ein Maß für den mittleren Informationsgehalt oder die Unsicherheit einer Zufallsvariable.

$$KL_v = \sum_{i=1}^{N_p} \frac{a_i}{a_t} \log \frac{\frac{a_i}{A_t}}{\frac{a_i}{A_t}} \quad (4.2)$$

wobei A_i die ursprüngliche Fläche des Polygons i (a_i ist dessen projizierte Fläche) und A_t die ursprüngliche Fläche aller Polygone einer Szene ist. Gute Sichtpunkte werden bei [SBERT u. a. \[2005\]](#) bei einer möglichst kleinen *Viewpoint Kullback-Leibler Distance* erreicht.

Einen anderen Weg, die *Viewpoint Entropy* von der Triangulierung der Szene unabhängig zu machen, präsentierten [VÁZQUEZ & ANDÚJAR \[2007\]](#). Sie erweiteren die *Viewpoint Entropy* um die Silhouettenlänge von Objekten und da diese symmetrisch für gegenüberliegende Sichtpunkte ist, nutzen sie weiterhin einen Tiefenpuffer, der bei gleicher Silhouetten-Entropie unterschiedliche Tiefenpuffer-Entropien liefert und so eine Unterscheidung der Sichtpunkte zuließ. [SOKOLOV & PLEMENOS \[2005\]](#) erweitern die *Viewpoint Entropy* auf die Krümmung der Oberfläche. Gute Sichtpunkte sind nun jene, die sowohl eine große Krümmung der sichtbaren Oberfläche, als auch große Flächen aufweisen.

Generell ist das Verfahren der *Viewpoint Entropy* kritisch zu sehen. Auch wenn das Problem der Abhängigkeit von der Triangulierung gelöst werden konnte, scheinen viele Erweiterungen am eigentlichen Problem vorbeizugehen: Der Frage, was ein *guter* Sichtpunkt ist und ob sich diese Frage überhaupt allgemeingültig beantworten lässt. Der Autor ist der Ansicht, dass je nach Anwendungsdomäne und Fragestellung an eine Visualisierung andere Sichtpunkte vom Betrachter als *gut* eingeschätzt werden würden.

Gute Sichtpunkte auf Szenen aus mehreren Objekten ermittelten [SOKOLOV u. a. \[2006b\]](#). Sie führten eine Wichtigkeit von Objekten ein, die sie direkt von der Größe der BoundingBoxen abhängig machen. Die Güte eines Sichtpunktes wird auf der Basis der *Viewpoint Entropy* ermittelt, allerdings werden anstelle der einzelnen Polygone Objekte und deren projizierte Fläche betrachtet. Ein guter Sichtpunkt ist demnach einer, von dem aus viele wichtige Objekte mit einer möglichst großen Fläche zu sehen sind.

Eine weitere Anwendung der *Viewpoint Entropy* lieferten [VÁZQUEZ u. a. \[2006\]](#), indem sie gute Sichten auf Molekülmodelle berechneten. Dabei wurde zwischen einem chemischen Blick (möglichst alle Atome sollten sichtbar sein) und einem physikalischen Blick (die Verbindungen zwischen den Atomen sollten gut sichtbar sein) auf die Darstellung unterschieden. Dieser Ansatz zeigt, dass semantische, an den Anwendungsfall gebundene Informationen essenziell für ein Verfahren zur Sichtpunktbestimmung sind.

[FU u. a. \[2008\]](#) stellten einen neuen Ansatz vor, um gute Sichtpunkte auf „*manmade objects*“ (von Menschen erschaffene Objekte) zu finden. Sie gingen dabei davon aus, dass solche Objekte zumeist auf einer großen Grundfläche standen – womit sich Sichten von schräg oben automatisch ermitteln ließen. Weitere Parameter wie die Symmetrie und Sichtbarkeit von Flächen und deren Wichtigkeit ließen

sie durch einen lernfähigen Algorithmus ermitteln. So konnten automatisch gute Sichtpunkte auf eine ganze Reihe von Einzelobjekten ermittelt werden.

Gute Sichtpunkte sind oft auch in Abhängigkeit weiterer verwendeter Visualisierungstechniken zu sehen. So haben [LI u. a. \[2007\]](#) ein Verfahren zur Sichtbestimmung vorgestellt, das darauf abzielte, Sichten auf das Innenleben komplexer Modelle, wie Flugzeugturbinen unter Zuhilfenahme von Cut-Aways, zu liefern. Da beim Cut-Away verdeckende Strukturen teilweise entfernt werden müssen, sollte deren Anzahl möglichst gering sein. Weiterhin sollte die gewählte Sicht ein sogenannter *canonical view*, also ein für den Nutzer vertrauter Sichtpunkt sein ([BLANZ u. a. \[1999\]](#)). Dazu hatte ein Autor verschiedene Kandidaten von Sichtpunkten für jedes Modell zuvor manuell bestimmt. Aus diesen Kandidaten wurde dann zur Laufzeit derjenige Sichtpunkt mit den wenigsten verdeckenden Strukturen vor dem Zielobjekt automatisch gewählt. Durch die manuelle Definition von Sichtpunkten ist das Verfahren von [LI u. a. \[2007\]](#) für eine automatische Sichtpunktbestimmung unbrauchbar.

Um die Szene nicht mehr aufwendig von allen in Frage kommenden Sichtpunkten aus rendern zu müssen, schlugen [VÁZQUEZ & SBERT \[2003\]](#) einen adaptierten Algorithmus zur Bestimmung der Viewpoint Entropy vor. Dabei wurden zu Beginn nur sechs Sichtpunkte an den Seiten eines das Objekt umgebenden Würfels gewählt, um die Viewpoint Entropy zu bestimmen. Die Werte für dazwischen liegende Sichtpunkte wurden interpoliert. Es wurde angenommen, dass alle Polygone, die von den beiden Punkten, zwischen denen interpoliert wurde, aus sichtbar sind, auch am neuen Sichtpunkt zu sehen sein werden. Dies tritt jedoch so gut wie nie auf. Da durch diese Annahme die Viewpoint Entropy zwischen zwei Sichtpunkten steigen kann, stoppt der Algorithmus, wenn kein neues Maximum der Viewpoint Entropy mehr erreicht wird. Weil der Algorithmus von [VÁZQUEZ & SBERT \[2003\]](#) nur auf sechs aus den realen Gegenbenheiten einer Szene berechneten Sichtpunktgüten basiert, stellen die Daten für alle anderen Sichtpunkte eine solch große Abstraktion dar, dass das Verfahren nur auf einzelne, sehr einfache Objekte angewendet werden kann und bei komplexen Objekten mit Selbstverdeckungen oder gar bei Darstellungen aus mehreren sich verdeckenden Objekten versagt.

[STOEV & STRASSER \[2002\]](#) versuchten, gute Sichtpunkte auf Terrains mit Simulationen von Völkerwanderungen zu erzeugen. Ein Ansatz, der rein auf einer maximal projizierten Fläche der Zielstruktur basiert, liefert hier schlechte Ergebnisse: Das Terrain wird von senkrecht oben gezeigt, was zu einer stark eingeschränkten Tiefenwahrnehmung führt. [STOEV & STRASSER \[2002\]](#) führten daher zusätzlich ein Maß für die Tiefenwirkung der Darstellung aus Sicht der Kamera in ihre Berechnungen ein.

4.3.2 Sichtpunktbestimmung in direkten Volumendarstellungen

Eine frühe Methode zur Unterstützung des Nutzers in direkten Volumendarstellungen durch vorbestimmte Sichtpunkte stellten [HASTREITER u. a. \[1998\]](#) vor. Für die Diagnose intracranialer Aneurysmen boten sie dem Arzt (Neuroradiologen bzw. Neurochirurgen) verschiedene Sichtpunkte an, von denen aus der Chirurg seine Exploration starten konnte. Die Sichtpunkte waren datensatzunabhängig, orientierten sich also nicht an der jeweiligen Visualisierung, sondern basierten vielmehr auf empirischen Erfahrungen.

Die Viewpoint Entropy, wie sie von [VÁZQUEZ u. a. \[2001\]](#) für polygonale Darstellungen eingeführt wurde, wurde auf verschiedene Weise für Volumendarstellungen adaptiert (u.a. von [BORDOLOI & SHEN \[2005\]](#), [TAKAHASHI u. a. \[2005\]](#) und [VIOLA u. a. \[2006\]](#)).

[BORDOLOI & SHEN \[2005\]](#) haben versucht, ohne zusätzliche Informationen über eine Darstellung gute Sichtpunkte in Volumendarstellungen zu identifizieren. Sie argumentierten, dass bei Volumendarstellungen vorbestimmte, gute Sichtpunkte besonders wichtig sind, da Volumendarstellungen für die interaktive Exploration oft noch zu langsam dargestellt werden können. Das Gütemaß eines Sichtpunktes basiert bei [BORDOLOI & SHEN \[2005\]](#) auf der Wichtigkeit (*noteworthiness*) von Voxeln: Voxel werden als wichtig eingeschätzt, wenn sie a) möglichst opak sind und b) eine seltene Farbe¹⁴ haben. Zusätzlich zu diesen Größen werden möglichst instabile Sichtpunkte gesucht, da diese als Startpunkt einer Exploration viele Veränderungen in der Darstellung versprechen. Dies ist eine interessante Vermutung, die aber noch einer ausführlichen Evaluierung bedarf. Sie widerspricht teilweise den in dieser Arbeit gemachten Erfahrungen mit Nutzern, bei denen kleine manuelle Änderungen mit einer großen Veränderung in der Darstellung eher für Verwirrung sorgten.

Weiterhin versuchten [BORDOLOI & SHEN \[2005\]](#) unterschiedliche Sichtpunkte zu finden, wobei sie sich zur Distanzmessung zwischen den verschiedenen Sichtpunktgüten der Jensen-Shannon-Divergenz¹⁵ bedienen. Ähnliche Sichtpunkte werden zu Regionen zusammengefasst und anschließend wird genau ein Repräsentant pro Region dem Betrachter angeboten.

Eine Erweiterung dieser Methode stellen [CHAN u. a. \[2008\]](#) vor, wobei sie besonderes Augenmerk auf die räumlichen Relationen zwischen verschiedenen Objekten legten. Sie machten vier mögliche Varianten an Relationen aus: separierte Objekte; Objekte, die sich berühren; Objekte, die sich überlagern; Objekte die sich umschließen. Solche Relationen wurden von [CHAN u. a. \[2008\]](#) automatisch ermittelt und daraufhin neben der Transferfunktion auch der Sichtpunkt angepasst, so dass eine gute Sicht auf den jeweiligen Relationsbereich (also z.B. die Überlappung) möglich war.

¹⁴Farbe und Transparenz werden dabei über eine Transferfunktion zugewiesen.

¹⁵Die Jensen-Shannon-Divergenz ist von der Kullback-Leibler-Divergenz abgeleitet ([LIN \[1991\]](#)).

VIOLA u. a. [2006] generierten automatisch gute Sichtpunkte auf Objekte in Volumendarstellungen, wobei auch hier Objekte durch eine Wichtigkeit ihrer Voxel gekennzeichnet sind. Neben der guten Sichtbarkeit möglichst vieler, wichtiger Voxel spielen zwei weitere Parameter eine Rolle bei der Sichtpunktbestimmung: Ein Gewicht des Bildraumes, bei dem die Objektvoxel möglichst zentral liegen sollen, und ein Gewicht des Objektraumes, bei dem die Voxel sich möglichst nah an der Kamera befinden sollen. Die Sichtbarkeit der Objekte wird dabei durch Cut-Away-Techniken sichergestellt.

Ein sehr aufwendiges Verfahren zur Sichtpunktbestimmung in Volumendarstellungen von Angiographien stellten CHAN u. a. [2006] vor. Sie ließen den Nutzer zunächst für ihn wichtige Voxel im Schichtbild-Datensatz markieren und jeweils eine Wichtigkeit angeben. Darauf aufbauend wurde versucht, Sichtpunkte zu finden, wo möglichst viele dieser Voxel entsprechend ihrer Wichtigkeit zu erkennen waren. Es ist allerdings fraglich, ob der Nutzer in der Zeit der Voxelmarkierung nicht selbst zu einer interessanten Sicht hätte gelangen können.

Mit dem Einsatz der Shannon-Entropy als Informationsmaß eines Sichtpunktes versuchten VÁZQUEZ u. a. [2008] gute Sichtpunkte auf medizinische Volumenvisualisierungen zu ermitteln. Dabei wird für jeden der drei Farbkanäle RGB über eine Histogramm-Analyse das Auftreten verschiedener Werte analysiert. Die Shannon-Entropy ist 0, wenn alle Pixel die gleiche Farbe haben, und erreicht ihr Maximum, wenn alle Pixel unterschiedlicher Farbe sind. Die Nutzbarkeit des Verfahrens, gerade auch im medizinischen Kontext, ist allerdings äußerst zweifelhaft. So ist für VÁZQUEZ u. a. [2008] nach der Shannon-Entropy die Sicht auf einen Schädel von oben der schlechteste Sichtpunkt und die Sicht von der Seite der beste. Jedoch kann gerade für einen Neurochirurgen oder in der Notfallmedizin genau diese Sicht die wichtigste und damit beste sein.

4.3.3 Bestimmung der Sichtbarkeit von Strukturen in polygonalen Darstellungen

Die Sichtbarkeit einer Zielstruktur ist vordringlichstes Merkmal eines jeden Sichtpunktes in einer Szene aus mehreren Strukturen. Daher werden im Folgenden verschiedene Möglichkeiten zur Bestimmung von Sichtbarkeiten einzelner Strukturen betrachtet.

Klassifikation: Man kann für die im Rahmen dieser Arbeit genutzten Darstellungen grob zwischen lokalen und globalen Verfahren zur Sichtbarkeitsanalyse unterscheiden (vergleiche Klassifikation von BITTNER & WONKA [2003]). Bei lokalen Verfahren, der sogenannten *Point Visibility*, wird die Sichtbarkeit einer Struktur von genau einem Punkt aus ermittelt. Oft wird nur eine binäre Entscheidung über die Sichtbarkeit geliefert, ohne zu ermitteln, in welchem Maße und von welchen Strukturen eine Zielstruktur möglicherweise verdeckt wird. Globale Verfahren zeichnen sich durch eine Betrachtung der gesamten Szene aus allen möglichen Sichtpunkten aus. Sie sind damit mächtiger als lokale Verfahren, bieten sie doch

erst die Möglichkeit, eine Kamera an einem Ort gewünschter Sichtbarkeit zu platzieren. Die Mächtigkeit globaler Verfahren schlägt sich aber auch in der größeren Komplexität und den erhöhten Performanzanforderungen nieder.

Lokale Verfahren

Nach [COHEN-OR u. a. \[2003\]](#) kann eine weitere Unterteilung anhand der Genauigkeit der gelieferten Ergebnisse vorgenommen werden. So wird zwischen exakten und approximativen Verfahren unterschieden. Eine Approximation bei globalen Verfahren kann bspw. durch eine Beschränkung der Kandidaten für mögliche Sichtpunkte erfolgen. Anwendungsbeispiele für **approximative, lokale Verfahren** liefern Computerspiele, bei denen eine echtzeitfähige Positionierung der Kamera des Spielers in einer sich verändernden Umgebung entscheidend ist. [HALPER u. a. \[2001\]](#) präsentierten daher den *Visibility Solver*. Dabei wurden in einem begrenzten Bereich die z-Buffer der jeweiligen Darstellung ausgewertet, um die Sichtbarkeit einer Zielstruktur zu ermitteln. Die Kamera wird auf die am nächsten gelegene Position bewegt, von der aus die Zielstruktur sichtbar ist. Das Verfahren von [HALPER u. a. \[2001\]](#) wird gemeinhin zu den lokalen Verfahren gezählt, da es keinerlei Informationen über die verdeckenden Strukturen liefert und sich auf einen sehr kleinen Positionierungsbereich beschränkt.

Ein **exaktes, lokales Verfahren** präsentierten [CHU & CHUNG \[2002\]](#). Ein Roboterarm sollte ein Zielobjekt greifen, welches von mehreren anderen Objekten umgeben und teilweise verdeckt wurde. Dazu musste die Kamera am Greifer das Zielobjekt ständig im Blick haben. Um Bereiche zu identifizieren, von denen aus das Zielobjekt sichtbar war, wurden die Geometrien der umgebenden Objekte auf eine den Aufbau umgebende Hemisphere projiziert. Der Arm und damit die Kamera wussten somit, welche Punkte sie nicht anfahren konnten und konnten aus den freien Positionen je nach Bewegungsmöglichkeiten des Roboterarmes neue Positionen für diesen ermitteln. Das Verfahren von [CHU & CHUNG \[2002\]](#) ist exakt, da es auf analytisch ermittelten Verdeckungsbereichen basiert.

Globale Verfahren

Verfahren, die **global** die Sichtbarkeiten **approximativ** ermitteln, sind bspw. Verfahren zur Radiosity ([HANRAHAN u. a. \[1991\]](#)). Bei Radiosity-Darstellungen muss zur Vorberechnung die Sichtbarkeit jedes Elementes von jedem anderen Element aus ermittelt werden. Dabei können Approximationen wie der Hemicube von [COHEN u. a. \[1988\]](#) zum Einsatz kommen.

Eine **exakte** Variante **globaler** Verfahren stellte das Visibility Skeleton von [DURAND u. a. \[1997\]](#) dar. Sie machten sich zunutze, dass zwei Punkte gegenseitig sichtbar sind, wenn kein anderes Element der Szene von ihrer Verbindungslinie geschnitten wird. Verändert sich der Kamerastandpunkt, können solche Linien sich plötzlich schneiden oder Eckpunkte von Polygonen können durch einen Wechsel

auf die andere Seite einer Verbindungslinie die Sichtbarkeitsverhältnisse ändern. Solche visuellen Ereignisse bilden die Kanten des Visibility-Skeleton-Graphs. Die Schnittpunkte von je zwei solcher Kanten (beide Ereignisse treten gleichzeitig am selben Ort auf) bilden die Knoten des Graphen. Das Verfahren von [DURAND u. a. \[1997\]](#) beschreibt somit global in der Datenstruktur des Visibility Skeletons die Sichtbarkeitsverhältnisse aller Objekte einer Szene. Durch die Nutzung geometrischer Konstrukte ist das Verfahren analytisch und damit als exakt einzustufen.

Bewertung für medizinische Visualisierungen

Für das Auffinden von guten Sichtpunkten auf eine Zielstruktur, wie es hier angestrebt wird, sind globale Verfahren das Mittel der Wahl. Durch das Vorliegen von Verdeckungsinformationen zu jeder Struktur aus jeder möglichen Sichtrichtung ist ein schneller Wechsel der Fokusstruktur jederzeit möglich. Als Nachteil ist der größere Aufwand bei der vollständigen Generierung der Sichtbarkeitsinformationen zu sehen. Der Aufwand lässt sich allerdings durch eine angemessene Approximation, z.B. der Kandidaten für mögliche Sichtpunkte, in Grenzen halten.

4.3.4 Anwendungen von Verfahren zur Sichtpunktbestimmung

Aus den vorliegenden Arbeiten ergibt sich ein weites Feld an Anwendungen für die automatische Bestimmung guter Sichtpunkte. Einige sollen im Folgenden kurz vorgestellt werden.

Sichtbarkeit auf Objekte von Interesse kann auch durch eine teilweise Anpassung der Szene erfolgen. So rendern [TAKAHASHI u. a. \[2006\]](#) Landmarken in einem Navigationssystem (z.B. Berge) derart, dass sie diese aus verschiedenen Blickwinkeln zeigen und so temporär den Blick auf wichtige Strukturen, wie z.B. zu verfolgende Routen, zulassen. Solche Veränderungen der Szene sind für medizinische Visualisierungen aus schon dargestellten Gründen der Szenenmanipulation nicht akzeptabel.

Die Sichtbarkeit von Objekten spielt eine große Rolle in der Vorstufe zum Rendering komplexer Szenen: Wenn mit geringem Aufwand ermittelt werden kann, welche Objekte nicht sichtbar sind, müssen diese erst gar nicht aufwendig gerendert werden. [COHEN-OR u. a. \[2003\]](#) geben dazu einen guten Überblick.

Verschiedene Ansätze für gute Sichtpunkte zielen auf deren automatische Generierung ab, um mit möglichst wenigen Sichtpunkten ausreichend viele Bilder für eine automatische Objekterkennung zu generieren ([DEINZER u. a. \[2003\]](#), [DERICHS u. a. \[2005\]](#) und [KOVACIC u. a. \[1998\]](#)).

Algorithmen zur automatischen Kameraplatzierung finden weiterhin Anwendung in der Gestaltung von 3D-Filmen sowie Computerspielen. Hier orientiert sich die Kameraposition an klassischen, cineastischen Einstellungen, wie „Close-

Up“, „Objekt im Profil“ oder „Objekt im Hintergrund“. CHRISTIE u. a. [2005] stellen dazu ein Framework vor, welches für animierte 3D-Darstellungen die Kamera nach jeweils für angemessen befundenen cineastischen Charakteristika positionierte. HALPER u. a. [2001] konzentrierten sich dabei im Wesentlichen auf ein frame-kohärentes Überblenden zwischen verschiedenen Kamerastandpunkten unter Berücksichtigung möglicher kommender Aktionen eines Spielers in einer 3D-Welt.

Um automatische 3D-Darstellungen mit dazugehörigen Texten zu verbinden, haben GÖTZELMANN u. a. [2007b] ein Verfahren entwickelt, bei dem aus einer aktuell gewählten Sicht die sichtbaren, opaken Strukturen entsprechend ihrer Fläche bestimmt werden. Daran anschließend wurde versucht, einen Textabschnitt zu finden, in dem genau diese Strukturen vorkommen, wobei die Häufigkeit des Auftretens im Text proportional zur gezeigten Fläche sein sollte.

Abschließend sei hier noch auf die Arbeit von TAN u. a. [2001] verwiesen. Sie berechneten gute Sichtpunkte auf einzelne Objekte zwar nicht automatisch, haben aber mit der „Ghost Copy“-Technik¹⁶ eine sehr interessante, interaktive Lösung präsentiert. Dabei wird dem Nutzer das Objekt von Interesse gesondert gezeigt, wobei der Nutzer das Objekt so manipuliert, dass es sich für ihn aus einer guten Sicht zeigt. Anschließend wird die Kamera animiert zu einem Sichtpunkt bewegt, der das Objekt genau aus dieser Richtung zeigt.

4.3.5 Diskussion und Zusammenfassung

Die vorliegenden Arbeiten, gerade auch für polygonale Darstellungen, zeichnen sich zumeist durch den völligen Verzicht auf anwendungs- oder domänenspezifischen Zusatzwissens aus. So vermitteln insbesondere Verfahren, die die Viewpoint Entropy nutzen, abwandeln oder verfeinern, den Eindruck, sie könnten gute Sichtpunkte auf jegliche Objekte vom Motorblock bis zum menschlichen Organ liefern. Dies mag bei einfachen Anforderungen, wie z.B. dem Wunsch, ein 3D-Objekt von schräg oben zu betrachten, ausreichend sein. Diese Ansätze versagen jedoch, sobald es um komplexere Objekte oder gar Szenen aus vielen, sich verdeckenden Objekten geht. So können je nach Anwendung und Nutzergruppe ganz unterschiedliche Sichtpunkte auf ein und dasselbe Objekt als gut oder schlecht gelten: Ein Neurochirurg schaut sich das 3D-Modell des Gehirns eines Patienten sicher aus völlig anderen Sichtrichtungen und unter anderen Aspekten an als bspw. der Patient selbst oder ein Medizinstudent in der Anatomieausbildung. Eine Hinzunahme von Wissen, welches den Nutzungskontext einer 3D-Darstellung widerspiegelt, ist daher unabdingbar und sollte Bestandteil eines Verfahrens zur automatischen Bestimmung von Sichtpunkten sein.

¹⁶Diese Technik ist patentiert (ROBERTSON u. a. [2006]).

4.4 Konzept der parametrisierbaren Kameraposition

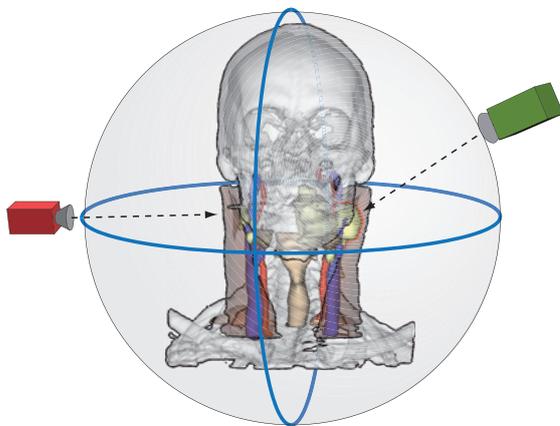
Aus der Anforderungsanalyse ergibt sich, dass die Güte eines Sichtpunktes von verschiedenen Parametern abhängt. Die Parameter haben je nach Fragestellung und Anwendung einen unterschiedlichen Einfluss bzw. eine unterschiedliche Wertigkeit. Im Rahmen dieser Arbeit wurde daher ein Verfahren entwickelt, welches die Güte eines Sichtpunktes anhand verschiedener gewichteter Parameter automatisch berechnen kann und damit in der Lage ist, Vorschläge für gute Sichtpunkte zu liefern. Der unter den jeweiligen Bedingungen beste Sichtpunkt wird automatisch aus einer endlichen Menge an Sichtpunktkandidaten gewählt. Diese Kandidaten sind gleichmäßig auf einer die Szene umgebenden Kugel verteilt. Der Ablauf des Verfahrens lässt sich grob in folgende Schritte unterteilen:

1. Bestimmung der Kandidaten für Sichtpunkte auf einer die Szene umgebenden Kugel (Abschnitt 4.4.1),
2. Vorberechnung bestimmter Parameter wie Sichtbarkeiten und projizierte Flächen (Abschnitt 4.4.1),
3. Bestimmung eines guten Sichtpunktes auf beliebige Strukturen und Strukturgruppen zur Laufzeit durch eine gewichtete Summierung aller Parameter (Abschnitte 4.4.2 und 4.4.3).

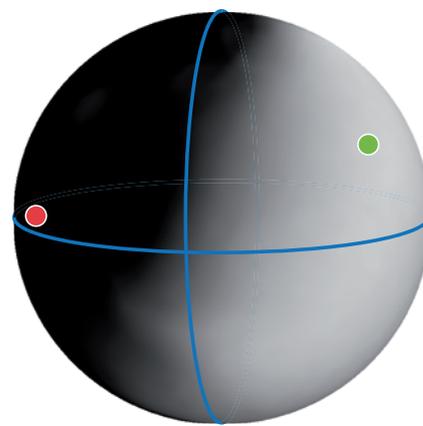
Jeder Parameter zur Bestimmung der Güte eines Sichtpunktes wird durch eine diskrete zweidimensionale Funktion (im Folgenden als Parameterfeld bezeichnet) repräsentiert:

$$p(\alpha, \beta) = \text{norm}_{[0,1]}(q) \quad (4.3)$$

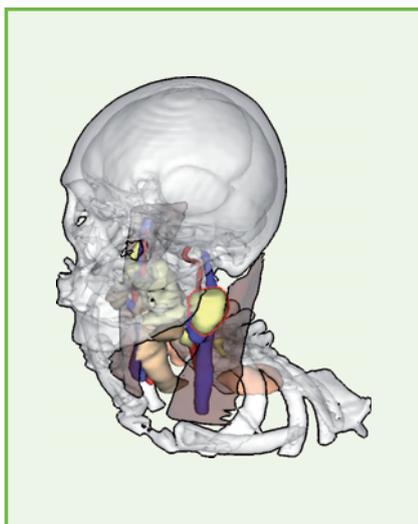
Das Parameterfeld beschreibt die skalaren Werte q eines Parameters an jeder Kameraposition, welche durch die Längen- und Breitengrade α und β eindeutig bestimmt ist. Jedes Parameterfeld ist auf den Bereich $[0, 1]$ normiert ($\text{norm}_{[0,1]}$), wobei Werte nahe 1 einen guten Sichtpunkt hinsichtlich des jeweiligen Parameters repräsentieren. In Abbildung 4.3 ist ein Parameterfeld beispielhaft für den Parameter der Sichtbarkeit eines vergrößerten Lymphknotens in einem Halsmodell zu sehen. Helle Werte stellen einen hohen Wert im Parameterfeld, hier eine gute Sichtbarkeit, dar. Von schwarz gefärbten Sichtpunkten aus ist der Lymphknoten sehr schlecht oder gar nicht zu sehen. Im Folgenden werden die Schritte des Verfahrens hin zur Bestimmung eines guten Sichtpunktes im Detail vorgestellt.



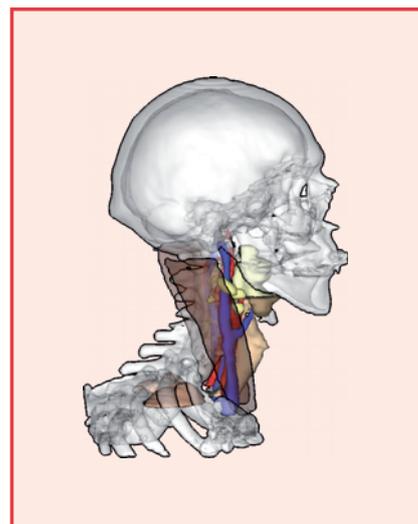
Kamerapositionen für die größte und geringste sichtbare Fläche des Lymphknotens



Parameterfeld mit dem besten und schlechtesten Sichtpunkt



Lymphknoten vom besten Sichtpunkt aus (größte sichtbare Fläche)



Lymphknoten vom schlechtesten Sichtpunkt aus (geringste sichtbare Fläche)

Abbildung 4.3: Darstellung eines Parameterfeldes für die Sichtbarkeit. Die Zielstruktur, ein vergrößerter Lymphknoten (rote Silhouette) ist im linken unteren Bild sehr gut zu sehen. Im rechten unteren Bild wurde ein Sichtpunkt gewählt, auf dem die Struktur sehr schlecht zu erkennen ist. Rechts oben ist die vordere Hälfte des Parameterfeldes zu sehen. Helle Werte repräsentieren eine gute Sichtbarkeit, dunkle eine schlechte.

4.4.1 Vorberechnungen

Da viele der im Verfahren genutzten Parameter auf Informationen zur Sichtbarkeit und Verdeckung von Strukturen basieren, werden diese Daten in einem Vorverarbeitungsschritt berechnet. Weil das Verfahren einen globalen Ansatz verfolgt, also alle möglichen Kamerastandpunkte in den Prozess zur Sichtpunktbestimmung einbezogen werden sollen, ist eine Bestimmung der Sichtbarkeitsverhältnisse nicht zur Laufzeit der eigentlichen Szenenexploration möglich.

Es wird für jede Kameraposition die Größe der projizierten Fläche einer jeden Struktur ebenso erfasst wie auch, welche Strukturen eine jede Struktur in welchem Maße verdeckt. Die Strukturen werden dazu einzeln in den z -Buffer gerendert und dieser ausgewertet. Genauer wird im Folgenden auf nachstehende Aspekte der Vorberechnung genauer eingegangen:

1. Ermittlung von Kandidaten für Sichtpunkte
2. Wahl einer angemessenen Auflösung des Viewports
3. Erstellung und Auswertung der z -Buffer
4. Erzeugung der Datenbasis
5. Laufzeitbetrachtungen

Eine Übersicht über die Schritte der Vorberechnung ist in Abb. 4.4 zu finden.

Ermittlung von Kandidaten für Sichtpunkte

Durch die Kompaktheit der Szenen bietet sich die Betrachtung einzelner Strukturen von einer umgebenden Kugel aus an. Um sowohl den Aufwand bei der Vorberechnung als auch bei der späteren Berechnung der Parameterfelder zu begrenzen, ist eine Diskretisierung der Kugeloberfläche notwendig. Dabei kann, ausgehend von einem Isokaeder durch dessen fortlaufende Unterteilung, eine immer größere Anzahl von Sichtpunkten erzeugt werden (Abb. 4.5). Die Sichtpunkte sind dabei gleichmäßig verteilt und bieten somit eine homogene Abdeckung der gesamten Kugeloberfläche. Die Unterteilung sollte nicht zu grob gewählt werden, da sonst durch die größeren Abstände zwischen den Sichtpunkten einzelne Strukturen in ihrer Sichtbarkeit nicht erfasst werden könnten (Abb. 4.6).

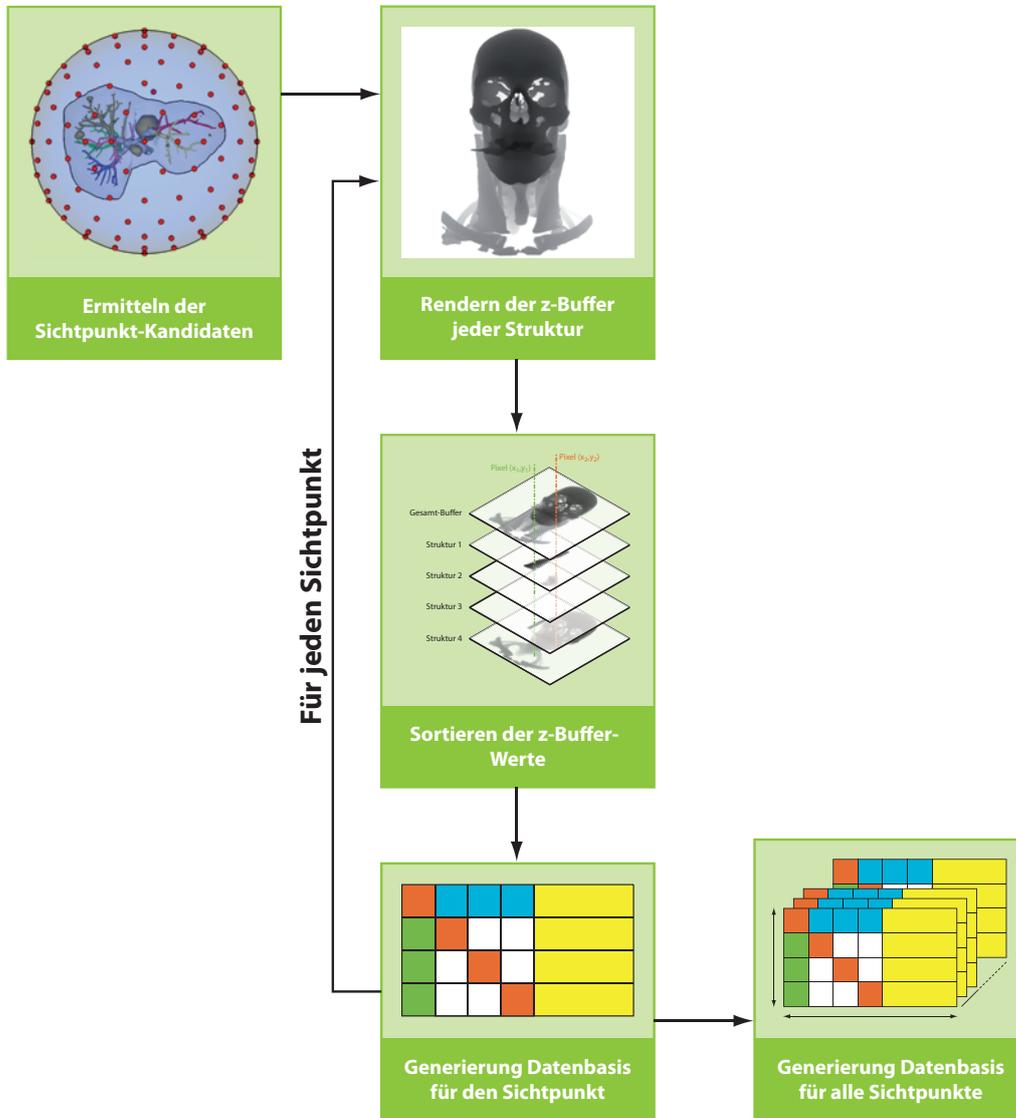
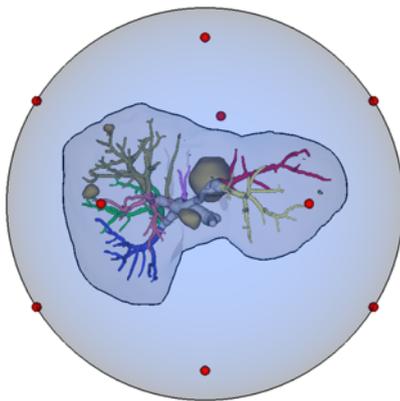
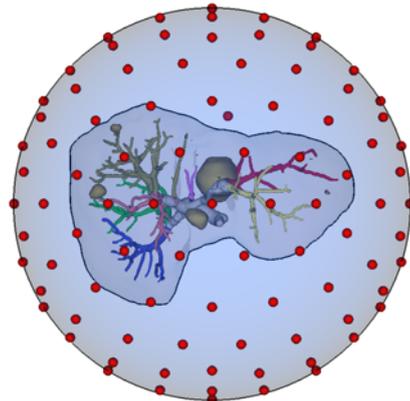


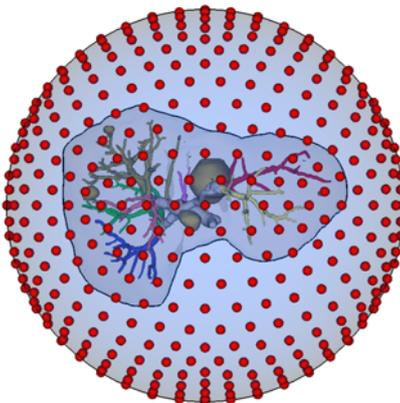
Abbildung 4.4: Schritte der Vorberechnung zur Ermittlung der Datenbasis für die Sichtpunktbestimmung. Nachdem die Kandidaten für die Sichtpunkte ermittelt wurden, werden für jeden Sichtpunkt die z-Buffer aller Strukturen gerendert, sortiert und eine Datenstruktur erzeugt. Zum Abschluss werden die Daten aller Sichtpunkte in einer gemeinsamen Datenstruktur vereinigt.



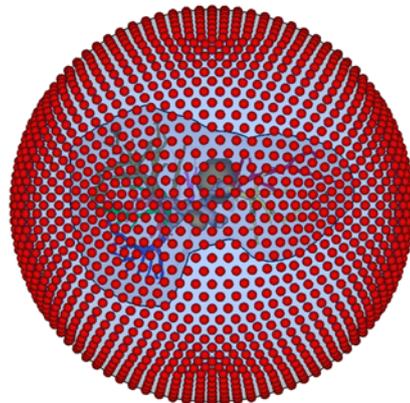
(a) Isokaeder:
12 Sichtpunkte



(b) 2-fache Unterteilung:
162 Sichtpunkte

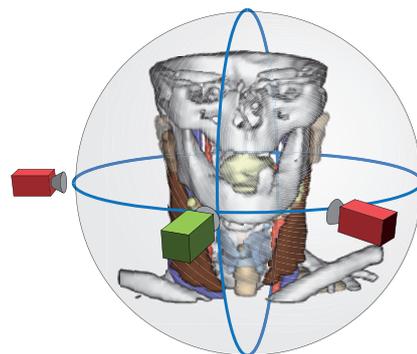


(c) 3-fache Unterteilung:
642 Sichtpunkte

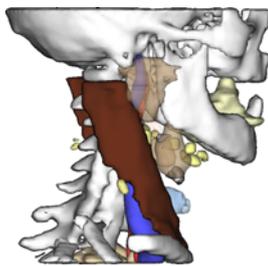


(d) 4-fache Unterteilung:
2562 Sichtpunkte

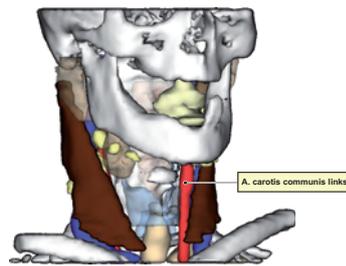
Abbildung 4.5: Unterteilung einer Kugeloberfläche für verschiedene Sichtpunkte. Dabei werden durch fortlaufende Unterteilung eines Isokaeders (a) durch hinzukommende Eckpunkte neue Sichtpunkte erzeugt.



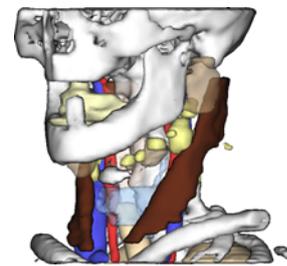
(a) 3 Sichtpunkte auf eine Darstellung



(b) Sichtpunkt 1



(c) Sichtpunkt 2



(d) Sichtpunkt 3

Abbildung 4.6: Effekte bei zu großen Abständen zwischen zwei Sichtpunkten. Werden nur die Sichtpunkte 1 und 3 (rote Kameras in (a)) berücksichtigt, so wird die Sichtbarkeit des Gefäßes (*Aorta carotis communis links*) von Sichtpunkt 2 aus nicht erfasst.

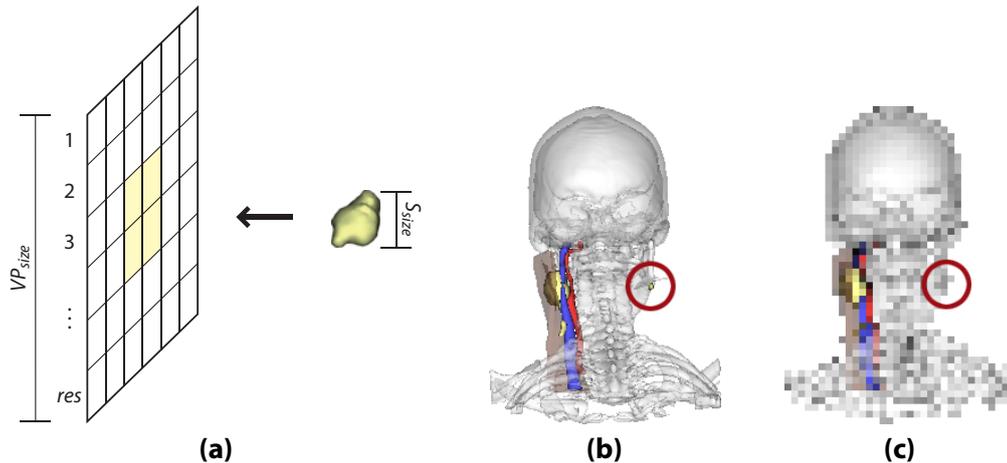


Abbildung 4.7: Angemessene Wahl der Viewport-Auflösung. Eine Struktur sollte mit mindestens einem Pixel im Viewport erfasst werden. Dazu muss die Auflösung res des Viewports so gewählt werden, dass $\frac{VP_{size}}{res} \leq S_{size}$ ist (a). Die Auflösung in (b) wurde ausreichend gewählt (300×300 Pixel), der Lymphknoten (roter Kreis) wurde erfasst. Die Auflösung in (c) ist zu gering (50×50 Pixel), da hier der Lymphknoten nicht mehr erfasst werden würde.

Wahl einer angemessenen Auflösung des Viewports

Zur Ermittlung der Sichtbarkeiten werden die Strukturen der Szene in einer orthogonalen Projektion auf den quadratischen Viewport¹⁷ der jeweiligen Kameraposition projiziert. Dabei werden sie entsprechend der Auflösung des Viewports in Pixel gerastert. Damit kleine Strukturen (z.B. kleine Lymphknoten) erfasst werden können, müssen sie mit mindestens einem Pixel gerastert werden. Die Viewport-Auflösung darf daher in Abhängigkeit von der Größe der kleinsten Struktur nicht zu klein gewählt werden. Wenn VP_{size} die Größe des Viewports, S_{size} die minimale Ausdehnung der kleinsten Struktur und res die Auflösung des Viewports in Pixeln ist, so muss $\frac{VP_{size}}{res} \leq S_{size}$ sein (Abb. 4.7).

Erstellung und Auswertung der z-Buffer

Um zu ermitteln, von welchen anderen Strukturen eine Struktur in welchem Maße verdeckt wird, werden im beschriebenen Verfahren z -Buffer genutzt. Jede Struktur wird einzeln in einen z -Buffer gerendert. Für jeden Pixel des Viewports werden die jeweiligen Werte aller z -Buffer an dieser Stelle sortiert (Abb. 4.8). Die an erster Stelle der sortierten Liste stehende Struktur bekommt einen unverdeckten Pixel zugeordnet. Allen anderen Strukturen in der Liste wird jeweils ein verdeckender Pixel jeder in der Liste vor ihr liegender Struktur zugeordnet. Ist dies für alle Pixel erfolgt, liegen für jede Struktur folgende Daten vor:

¹⁷Bei einem quadratischen Viewport werden weder horizontale noch vertikale Viewports bevorzugt. Das Seitenverhältnis des Viewports später beim Nutzer ist bei der Generierung der Datenbasis noch nicht bekannt.

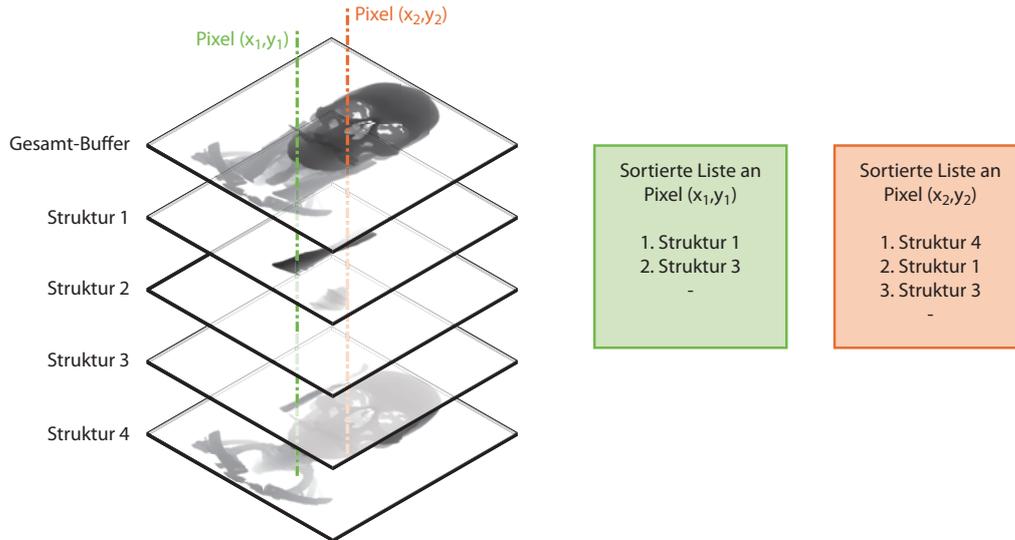


Abbildung 4.8: Sortierung der z-Buffer. Die z-Buffer der einzelnen Strukturen werden an jeder Pixelposition (x, y) sortiert und so die Anzahl der sichtbaren und verdeckten Pixel pro Struktur ermittelt. Beispielfhaft ist dies hier für zwei Positionen dargestellt.

- Anzahl ihrer Pixel (egal ob verdeckt oder unverdeckt),
- zu jeder verdeckenden Struktur die Anzahl der verdeckenden Pixel,
- der minimale und maximale Tiefenwert zur Ermittlung der Kameranähe einer Struktur,
- die Zahl der Silhouettenpixel einer Struktur als Pixel mit mindestens einem Hintergrundpixel in ihrer 4er Nachbarschaft,
- eine Maßzahl über die Zentralität einer Struktur, deren Berechnung Gleichung 4.12 in Abschnitt 4.4.2 zu entnehmen ist.

Aus Gründen der Optimierung wird zusätzlich zu den z-Buffern der einzelnen Strukturen ein zusätzlicher z-Buffer mit allen Strukturen gerendert. Dieser wird genutzt, um eine Sortierung der Buffer nur an den Pixeln durchzuführen, an denen mindestens eine Struktur abgebildet wird.

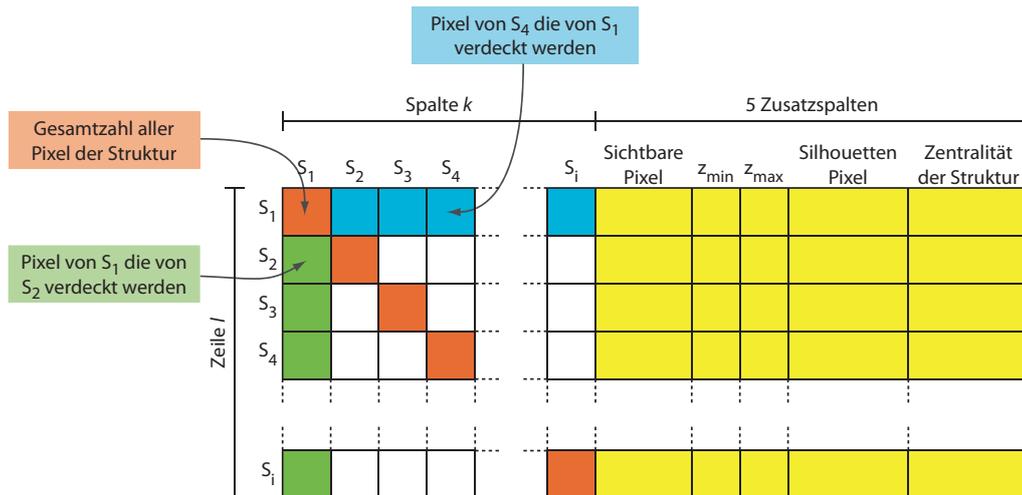


Abbildung 4.9: Datenstruktur zur Speicherung der Sichtpunktdaten. Die gewonnenen Daten in Form ausgezählter Pixel der z -Buffer werden für jeden potentiellen Sichtpunkt in einer zweidimensionalen Datenstruktur gespeichert. In jeder Zeile finden sich für jede Struktur S die Anzahl der Pixel, die diese Struktur bei anderen Strukturen verdeckt (die ersten i Spalten) sowie weitere Daten, wie die Anzahl der Silhouetten-Pixel, in zusätzlichen Spalten.

Erzeugung der Datenbasis

Die aus der Sortierung der z -Buffer-Werte gewonnenen Daten werden in einer zweidimensionalen Datenstruktur der Größe $i + 5 \times i$ gespeichert, wobei i die Anzahl der Strukturen ist. In jeder Spalte k ($6 \leq k \leq i + 5$) sind die Anzahl der Pixel enthalten, die von der jeweiligen Struktur in Zeile l verdeckt werden (Abb. 4.9). Im Eintrag $k = l$ wird die Anzahl aller Pixel der Struktur k vermerkt (ohne Berücksichtigung von Verdeckungen). In den 5 zusätzlichen Spalten werden zu jeder Struktur in l vermerkt:

1. die Zahl unverdeckter Pixel,
2. die kleinsten und größten z -Werte,
3. die Zahl der Silhouettenpixel,
4. die Maßzahl über die Zentralität einer Struktur.

Die Datenstruktur in Abb. 4.9 wird für jeden Sichtpunkt erzeugt. In der Gesamtheit aller Sichtpunkte ergibt sich damit eine dreidimensionale Datenstruktur, wie sie in Abb. 4.10 dargestellt ist, der Größe $i \times i + 5 \times n$ mit n als Anzahl der Sichtpunkte.

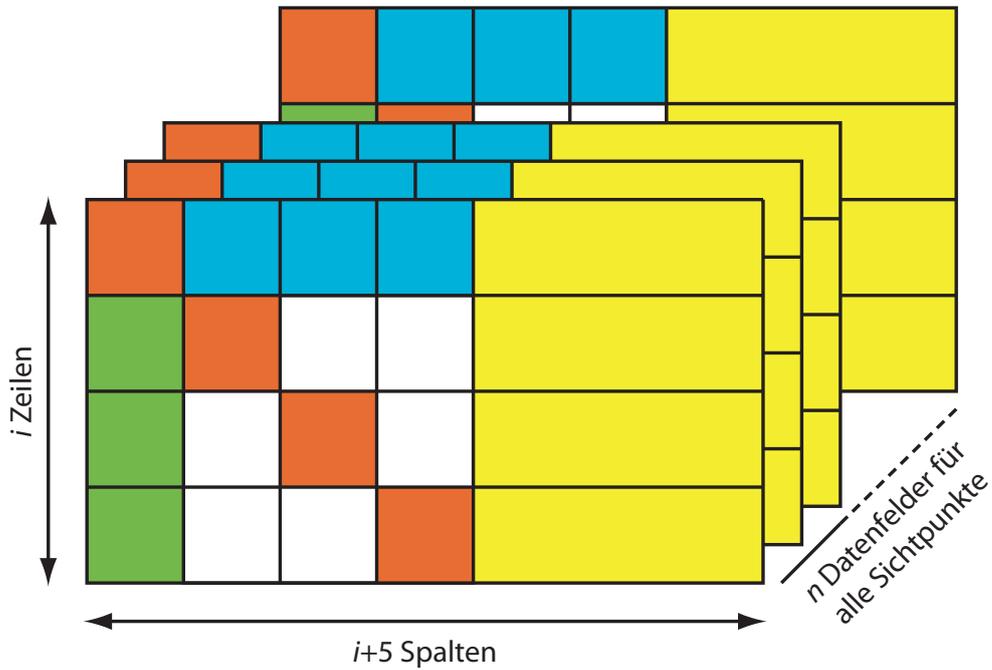


Abbildung 4.10: Datenstruktur zur Speicherung aller Daten aller Sichtpunkte. Für jeden Sichtpunkt existiert eine zweidimensionale Matrix, wie sie in Abbildung 4.9 schematisch dargestellt ist.

Laufzeitbetrachtungen

Die Laufzeit der Vorberechnungen ist von drei Faktoren abhängig:

1. Anzahl der Strukturen
2. Auflösung des Viewports
3. Anzahl der Sichtpunkte

Die Wahl der letzten beiden Punkte hat direkten Einfluss auf die Genauigkeit des Verfahrens – eine größere Anzahl an Sichtpunkten erlaubt später eine exaktere Positionierung der Kamera und eine größere Viewport-Auflösung liefert genauere Aussagen über Sichtbarkeiten und Verdeckungen. Auf die Anzahl der Strukturen hat man keinen Einfluss – diese ist durch die Szene vorgegeben. Eine angemessene Wahl der Viewport-Auflösung wie auch der Anzahl der Sichtpunkte kann daher performancesteigernd wirken. Als in der Anwendung ausreichend hat sich eine Kombination aus 162 Sichtpunkten bei einer Viewport Auflösung von 200×200 Pixeln herausgestellt. Die Auflösung genügte sowohl zur ausreichenden Darstellung auch kleiner Strukturen, wie Lymphknoten, und die Zahl der Sichtpunkte wurde mit 162 indirekt als ausreichend ermittelt, indem für eine Vielzahl von Datensätzen (ca. 90) bei nahezu allen Anfragen ein guter Sichtpunkt gefunden werden konnte. Auf einem Intel Pentium 4 mit 3,4 GHz und 3 GB RAM werden mit diesen Parametern für einen Datensatz aus 21 Strukturen 84s benötigt (weitere Daten siehe Tabelle 4.1).

Anzahl der Strukturen	Viewport-Auflösung	Sichtpunkte	Benötigte Zeit
21	300 × 300	42	45s
21	400 × 400	162	295s
21	300 × 300	162	171s
21	200 × 200	162	84s
10	300 × 300	42	23s
10	300 × 300	162	94s
10	200 × 200	162	49s

Tabelle 4.1: Übersicht über die Berechnungszeit für die Vorberechnungen. Es wurden verschiedene Datensatzgrößen, Viewport-Auflösungen und Sichtpunktmen-gen auf einem Intel Pentium 4 mit 3,4 GHz und 3 GB RAM getestet. Die empfohlenen Einstellungen sind hervorgehoben.

4.4.2 Genutzte Parameter

Sichtbare Fläche

Einer der wichtigsten Parameter ist der der sichtbaren Fläche einer Struktur. Eine Struktur soll für den Betrachter oft aus einer Sichtrichtung mit der größtmöglichen Fläche sichtbar sein. Für ein elongiertes Gefäß ist dies eine Position, von der aus es in seiner ganzen Ausdehnung sichtbar wird, im Vergleich zu einer Position, die nur den sehr kleinen Querschnitt des Gefäßes zeigt (Abb. 4.11). Die Daten für die Berechnung der sichtbaren Fläche einer Struktur p_{vis} fließen direkt aus der Datenbasis der Vorberechnung ein, wobei s_{unv} die Anzahl der unverdeckten Pixel einer Struktur aus Sicht der Kameraposition (α, β) ist:

$$p_{\text{vis}}(\alpha, \beta) = s_{\text{unv}}(\alpha, \beta) \quad (4.4)$$

Wichtigkeit verdeckender Strukturen

Nicht immer sind Sichtpunkte möglich, von denen aus die Zielstruktur mit einer großen, unverdeckten Fläche sichtbar ist. Ein gutes Beispiel sind Strukturen innerhalb des Leberparenchyms, die aus allen Sichtrichtungen von der umgebenden Kugel aus vom Parenchym verdeckt werden. In solchen Fällen kann der Umstand der unterschiedlichen Wichtigkeit von Strukturen für den Kontext einer Darstellung genutzt werden. Kontextuell weniger wichtige Strukturen können beispielsweise sehr transparent dargestellt werden, so dass man durch sie hindurchsehen kann. Auch können bei solchen Strukturen andere Techniken wie Cut-Aways



Abbildung 4.11: Ein guter (grün) und ein schlechter (rot) Sichtpunkt auf ein Gefäß des Halses (*Vena jugularis*) unter Berücksichtigung der projizierten Fläche.

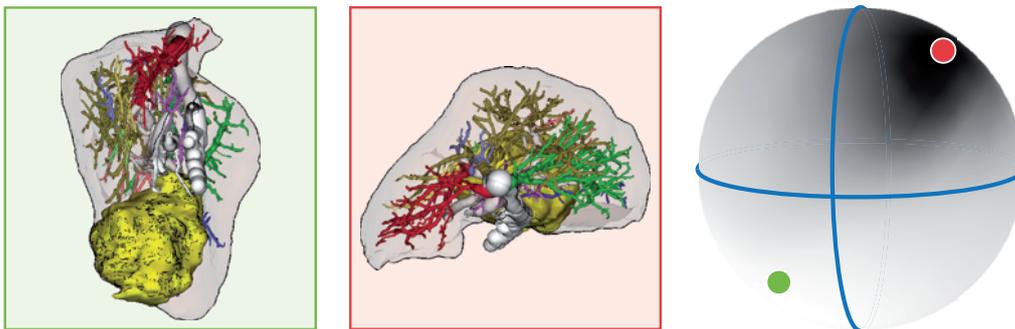


Abbildung 4.12: Ein guter (grün) und ein schlechter (rot) Sichtpunkt auf einen Tumor (gelbe Struktur) innerhalb einer Leber. Der Tumor ist von allen Seiten verdeckt. Durch die geringere kontextuelle Wichtigkeit der Leber kann jedoch ein Sichtpunkt gewählt werden, bei dem der Tumor zwar komplett verdeckt wird – allerdings nicht von kontextuell sehr wichtigen (opaquen) Gefäßen.

(TJETJEN u. a. [2009]) eher angewendet werden. Daher wird im vorgestellten Verfahren der Parameter der kontextbezogenen Wichtigkeit verdeckender Strukturen p_{imp} genutzt. Er beschreibt die Größe der projizierten Fläche einer Struktur s in Pixeln aus Sicht der jeweiligen Kameraposition (α, β) im Verhältnis zur verdeckenden Fläche v einer Struktur i in Pixeln in Abhängigkeit von der jeweiligen kontextuellen Wichtigkeit w_s ($0 \leq w_s \leq 1$) der verdeckenden Struktur i :

$$p_{\text{imp}}(\alpha, \beta) = 1 - \text{norm}_{[0,1]} \sum_{i=1}^n w_{s_i} \frac{v_i(\alpha, \beta)}{s(\alpha, \beta)} \quad (4.5)$$

Wird eine Struktur von keiner anderen verdeckt, ist p_{imp} an dieser Position 1. Wird sie nur von wenigen unwichtigen Strukturen zu einem geringen Teil verdeckt, sind für p_{imp} Werte nahe 1 zu erwarten. Für den Fall einer kompletten Verdeckung der Struktur ist über die Steuerung der Wichtigkeit w_s immer noch die Möglichkeit gegeben, wichtige (opaque) von unwichtigen (transparenten) Verdeckungen zu unterscheiden (Abb. 4.12).

Der Parameter der kontextbezogenen Wichtigkeit kann als Erweiterung des Parameters der Sichtbarkeit verstanden werden. Bei einer maximalen Wichtigkeit von 1 aller Strukturen sind die beiden Parameterfelder ähnlich.

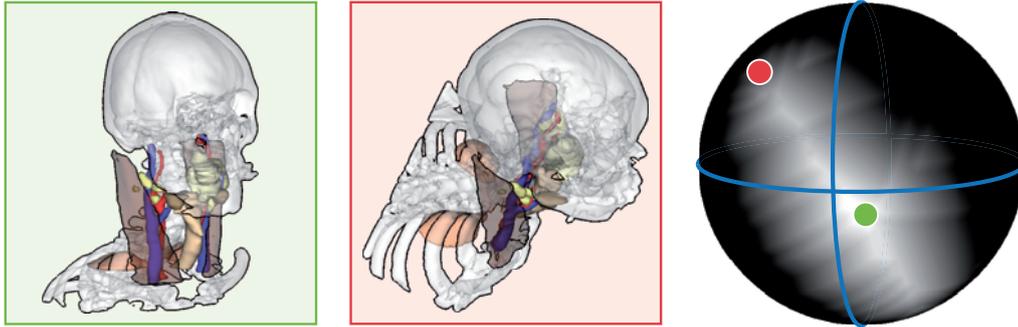


Abbildung 4.13: Ein stabiler (grün) und ein instabiler (rot) Sichtpunkt auf einen Lymphknoten im Hals (rote Silhouette). Bei kleinen Veränderungen des schlechten Sichtpunktes würde der Lymphknoten sofort von umliegenden Strukturen verdeckt werden. Bei dem guten Sichtpunkt kann die Kamera verändert werden, ohne dass sich die Sichtbarkeit des Lymphknotens wesentlich verschlechtert.

Stabilität der Sichtbarkeit

Auf eine automatische Sichtpunktbestimmung folgt oft eine weitere individuelle Exploration der Darstellung durch den Nutzer. Der Sichtpunkt sollte daher eine gewisse Stabilität hinsichtlich der Sichtbarkeit der Zielstruktur aufweisen. Kleine Änderungen an der Kameraposition durch den Nutzer sollten keine großen Änderungen in der Sichtbarkeit der Struktur zur Folge haben. So kann das Maximum der Sichtbarkeit einer Struktur in ungünstigen Fällen genau von einem Sichtpunkt aus erreicht werden, bei dem durch eine kleine Öffnung in den umgebenden Strukturen (z.B. Knochen) auf die Zielstruktur geblickt wird. Kleine Abweichungen von diesen Sichtpunkten würden zu einer kompletten Verdeckung der Zielstruktur führen (Abb. 4.13). Um dies zu verhindern, sollten bei vielen Sichtpunkten mit ähnlich hohen Sichtbarkeitswerten diejenigen gewählt werden, die von mehreren Sichtpunkten mit ebenso hohen Sichtbarkeitswerten umgeben sind.

Die Parameterfelder für die Stabilität von Sichtpunkten $p_{\text{vis_sta}}$ bzw. $p_{\text{imp_sta}}$ werden auf der Basis der sichtbaren Fläche p_{vis} und der Wichtigkeit verdeckender Strukturen p_{imp} erzeugt. Dazu werden die vorliegenden Parameterfelder anhand jeweils eines Schwellenwertes ($\text{tresh}_{\text{vis}}$ bzw. $\text{tresh}_{\text{imp}}$) binarisiert (Abb. 4.14). Die Schwellenwerte legen fest, wie hoch die Sichtbarkeit einer Struktur mindestens sein muss, um bei der Stabilitätsberechnung berücksichtigt zu werden. Ein Wert von 0,7 auf den $[0, 1]$ normierten Parameterfeldern hat sich empirisch als sinnvoll erwiesen. Auf die binarisierten Felder wird eine Distanztransformation d angewendet, um den Abstand jedes im binarisierten Feld mit 1 bewerteten Sichtpunktes zum Rand einer Region zu bestimmen. Sichtpunkte im Zentrum einer Region erhalten somit höhere und bessere Werte als Sichtpunkte am Rand einer Region (Abb. 4.14):

$$p_{\text{vis_sta}}(\alpha, \beta) = \text{norm}_{[0,1]}(d(\text{bin}_{[\text{tresh}_{\text{vis}}]}(p_{\text{vis}}(\alpha, \beta)))) \quad (4.6)$$

$$p_{\text{imp_sta}}(\alpha, \beta) = \text{norm}_{[0,1]}(d(\text{bin}_{[\text{tresh}_{\text{imp}}]}(p_{\text{imp}}(\alpha, \beta)))) \quad (4.7)$$

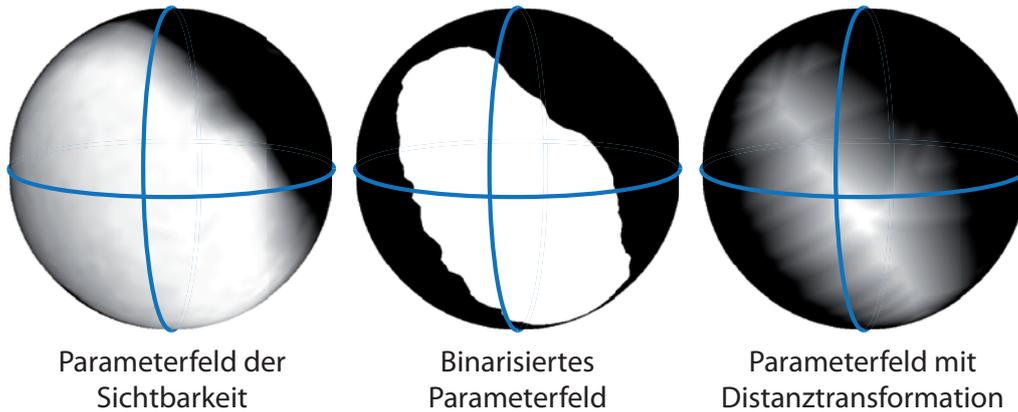


Abbildung 4.14: Schrittweise Erzeugung des Parameterfeldes für den Parameter der Stabilität. Das Parameterfeld der Sichtbarkeit wird an einem Schwellenwert binarisiert und mithilfe einer Distanztransformation werden Sichtpunkte im Zentrum von Bereichen großer Sichtbarkeit ermittelt.

Anzahl verdeckender Strukturen

Für Zielstrukturen, die von mehreren Strukturen umgeben sind, ist die Anzahl der jeweils verdeckenden Strukturen bei der Ermittlung eines guten Sichtpunktes wichtig. Kommen in solchen Fällen Techniken zum Ausblenden oder teilweisen Aufschneiden der verdeckenden Strukturen zum Einsatz (z.B. Cut-Aways oder Ghost-Views), so sollten möglichst wenig Strukturen davon betroffen sein. Die Anzahl der verdeckenden Strukturen n aus Sicht einer Kameraposition (α, β) wird aus der Datenbasis der Vorberechnungen abgeleitet und für das Parameterfeld p_{occ} normiert:

$$p_{occ}(\alpha, \beta) = norm_{[0,1]}(n) \tag{4.8}$$

Vorzugsblickrichtung

Ein Sichtpunkt, der auf Basis der bisherigen Parameter als gut bewertet werden würde, kann dennoch für den Betrachter ein schlechter sein, weil er die Zielstruktur aus einem ungewohnten Blickwinkel zeigt. Die bevorzugte Blickrichtung des Nutzers fließt daher in Form der Vorzugsblickrichtung p_{pref} in die Berechnung ein. Das Parameterfeld p_{pref} ist ein künstlich erzeugtes Feld, welches an Sichtpunkten, die der Nutzer bevorzugt, Werte nahe 1 enthält und ungewünschte Sichtpunkte mit Werten nahe 0 bewertet. Da eine individuelle Erfassung der Vorzugsblickrichtung im Alltag zu aufwendig wäre, stellen anwendungsorientierte Verallgemeinerungen eine gute Lösung dar. So bevorzugen HNO-Chirurgen Sichtpunkte in einer äquatorialen Ebene um eine Halsdarstellung und lehnen Sichtpunkte von oben oder unten auf die Szene ab (Abb. 4.15). Ein Parameterfeld für diesen Fall enthält hohe Werte für Sichtpunkte in Äquatornähe und abfallende Werte zu den Polen.

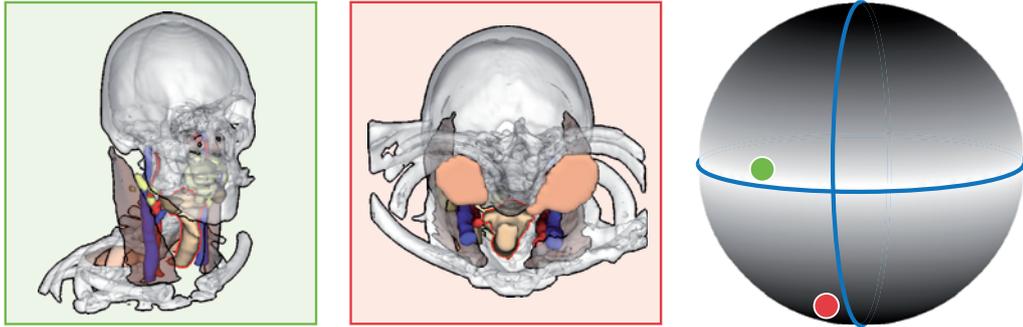


Abbildung 4.15: Zwei Sichten auf die Luftröhre (*Trachea*). Bei dem schlechten (roten) Sichtpunkt wurde nur die sichtbare Fläche berücksichtigt. Die Blickrichtung von unten wird jedoch von HNO-Chirurgen abgelehnt. Daher wurde bei dem guten (grünen) Sichtpunkt zusätzlich die äquatoriale Vorzugsblickrichtung berücksichtigt. Das Parameterfeld p_{pref} mit der Vorzugsblickrichtung ist rechts dargestellt.



Abbildung 4.16: Zwei gute Sichtpunkte auf Strukturen des Knies. Bei beiden wurde neben der sichtbaren Fläche auch die Vorzugsblickrichtung des Orthopäden (von superior bzw. anterior) berücksichtigt. Das entsprechende Parameterfeld p_{pref} ist rechts dargestellt. Vom grünen Sichtpunkt aus ist das hintere Kreuzband (*Ligamentum cruciatum posterius*) sehr gut zu erkennen. Beim gelben Sichtpunkt wird von oben auf den Innenmeniskus (*Meniscus medialis*) geschaut, um ihn auf mögliche Risse hin zu untersuchen.

Ein weiteres Beispiel ist die Betrachtung einer 3D-Darstellung des Knies durch einen Orthopäden. Möchte dieser ein Knie vorzugsweise von vorn (*anterior*) oder von oben (*superior*) betrachten, ließe sich das entsprechende Parameterfeld p_{pref} mit zwei Sichtpunkten mit dem Wert 1 und von diesen mit der Distanz abfallenden Werten generieren (Abb. 4.16).

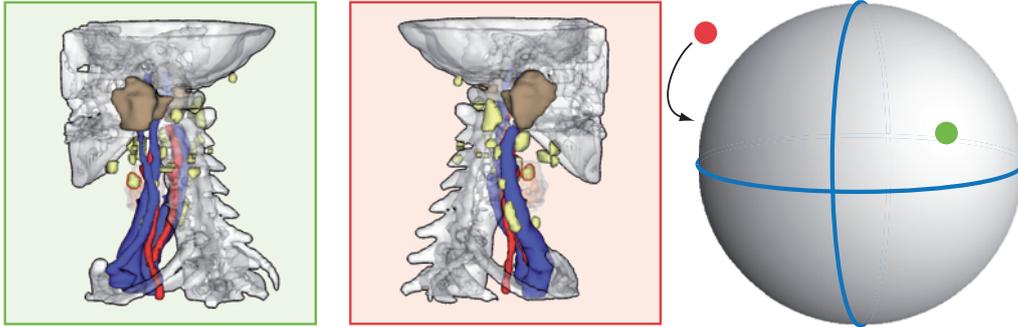


Abbildung 4.17: Zwei Sichtpunkte auf einen Lymphknoten (rote Silhouette) im linken Halsbereich. Der gute Sichtpunkt (grün) hat eine geringe Entfernung zur Zielstruktur. Bei dem schlechten Sichtpunkt (rot) ist der Lymphknoten selbst gut zu sehen – allerdings muss der Betrachter durch die gesamte Szene hindurch auf die andere Seite des Halses blicken, da der Lymphknoten hier auf der anderen Seite liegt.

Entfernung zwischen zwei Kamerapositionen bei Kameraflügen

Die automatische Bestimmung von Sichtpunkten im Laufe einer Animation oder während der Exploration der Szene ist immer mit einem Kameraflug zum neuen Sichtpunkt verbunden. Dies erfordert eine Neuorientierung des Betrachters und birgt die Gefahr des Orientierungsverlustes. Daher sind bei ansonsten ähnlichen Eigenschaften die Sichtpunkte zu bevorzugen, die näher an der aktuellen Kameraposition liegen. Dies wird mit dem Parameterfeld p_{cam} erreicht, welches die Distanzen d der einzelnen Sichtpunkte zur aktuellen Kameraposition $(x_{\text{cam}}, y_{\text{cam}}, z_{\text{cam}})$ auf $[0, 1]$ normiert enthält, wobei mit 1 der Sichtpunkt mit dem geringsten Abstand zur aktuellen Kameraposition bewertet wird:

$$p_{\text{cam}}(\alpha, \beta) = \text{norm}_{[0,1]}(d((\alpha, \beta), (x_{\text{cam}}, y_{\text{cam}}, z_{\text{cam}}))) \quad (4.9)$$

Entfernung der Zielstruktur zur Kamera

Selbst wenn die Zielstruktur gut sichtbar ist, kann sie am entgegengesetzten Ende der Szene und damit weit vom Betrachter entfernt liegen. Dies erschwert die Bewertung der Umgebung und der räumlichen Zusammenhänge um die Zielstruktur (Abb. 4.17). Daher sorgt der Parameter p_{dis} für eine bessere Bewertung von Sichtpunkten, die näher an der Position der Zielstruktur $(x_{\text{target}}, y_{\text{target}}, z_{\text{target}})$ liegen:

$$p_{\text{dis}}(\alpha, \beta) = d((\alpha, \beta), (x_{\text{target}}, y_{\text{target}}, z_{\text{target}})) \quad (4.10)$$

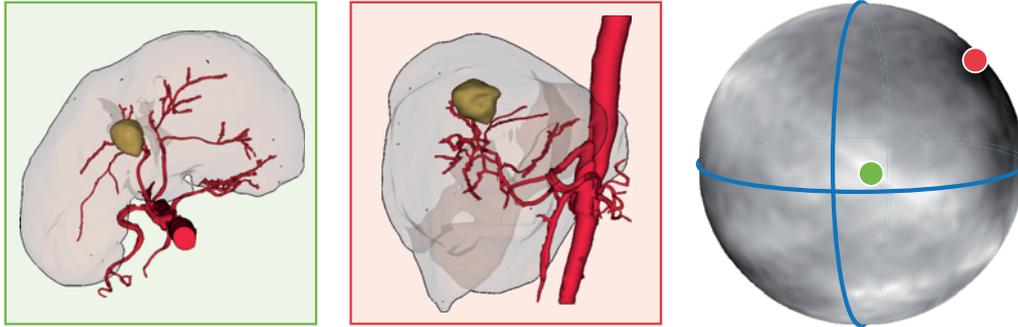


Abbildung 4.18: Ein guter (grün) und schlechter (rot) Sichtpunkt auf die Leberarterie (*Arteria hepatica propria*), wobei die Silhouettenlänge besonders berücksichtigt wurde. Eine lange Silhouette führt zumeist auch zu einer größeren Anzahl von sichtbaren Verzweigungen und Teilästen des Gefäßbaumes (grüner Sichtpunkt). Dies ist allein bei einer möglichst großen projizierten Fläche (roter Sichtpunkt) nicht garantiert.

Länge der Silhouette

Bei der Beurteilung von Gefäßbäumen, z.B. bei der Tumorsektion der Leber, ist es für den planenden Chirurgen besonders wichtig, möglichst viele Verzweigungen und Äste des Gefäßbaumes zu sehen, auch wenn diese eine sehr kleine Fläche aufweisen. Von einem Gefäßbaum sind mutmaßlich die meisten Äste zu sehen, wenn die Silhouette¹⁸ der projizierten Fläche lang ist. Daher ist bei der Bestimmung von guten Sichtpunkten auf Gefäßbäume der Parameter der Silhouettenlänge p_{sil} von Bedeutung (Abb. 4.18). Als Silhouette werden dabei alle Pixel der projizierten Fläche der Zielstruktur betrachtet, die in ihrer 4er Nachbarschaft ein Hintergrundpixel haben. Die Daten zur Silhouettenlänge in Pixeln sil werden im Zuge der Vorberechnungen erfasst:

$$p_{\text{sil}}(\alpha, \beta) = \text{norm}_{[0,1]}(sil) \quad (4.11)$$

Zentrierung der Zielstruktur im Bildraum

Damit der betrachteten Zielstruktur eine hohe Aufmerksamkeit entgegenkommt, ist ihre Positionierung möglichst in der Mitte des Bildes wichtig. Ein Beispiel dafür ist in Abbildung 4.19(a) zu sehen. Dies lässt sich durch eine Ausrichtung der Kamera auf die Zielstruktur, vereinfacht auf das Zentrum ihrer BoundingBox erreichen (Abb. 4.19(c)). Liegt die Zielstruktur allerdings am Rand des Bildraumes der Kamera, verschieben sich durch einen solchen Schwenk meist große Teile der Szene an den Bildrand oder gar darüber hinaus (Abb. 4.19(c)). Die Szene „hängt“

¹⁸ Als Silhouette werden dabei Randpixel im Bildraum betrachtet, und nicht die Punkte der Objektfläche, an denen der Sichtvektor senkrecht zur Oberflächennormalen steht. Dies spart zusätzliche Berechnungen im Objektraum.

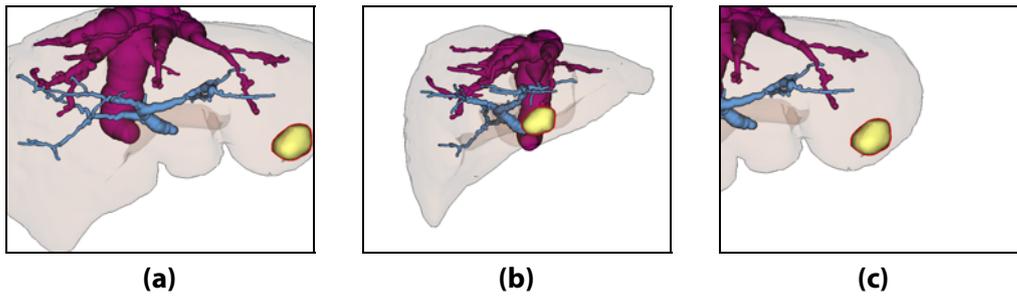


Abbildung 4.19: Darstellung der Bedeutung der Lage einer Zielstruktur möglichst im Zentrum des Bildraumes. Kamerapositionen, bei denen sich die Zielstruktur am Rand des Bildraumes befindet (a), sind schlechter als solche, bei denen die Zielstruktur im Zentrum liegt (b). Eine reine Ausrichtung der Kamera auf die Zielstruktur reicht oft nicht aus, da es so zu einer Verschiebung der übrigen Szene aus dem Bildraum heraus kommen kann (c).

förmlich seitlich an der im Bildraum zentrierten Zielstruktur. Um dies zu vermeiden, ist eine möglichst zentrale Positionierung der Zielstruktur im Bildraum schon von Beginn an wünschenswert (Abb. 4.19(b)). Dies bedeutet, dass bei der Generierung der Datenbasis (siehe Abschnitt 4.4.1), wenn die Kamera zum Szenenmittelpunkt ausgerichtet ist, Daten gewonnen werden müssen, die Aussagen über die „Zentralität“ der Zielstruktur erlauben. Zu diesem Zweck wird die mittlere Entfernung d_{cen} aller Pixel einer Struktur (unabhängig von deren Sichtbarkeit) vom Bildraummittelpunkt ermittelt:

$$d_{cen} = \frac{\sum_{i=1}^n 1 - \frac{\sqrt{x_i^2 + y_i^2}}{\sqrt{\left(\frac{w}{2}\right)^2 + \left(\frac{h}{2}\right)^2}}}{n} \quad (4.12)$$

Dabei ist n die Anzahl der projizierten Pixel der Struktur für die aktuelle Kameraposition. x und y sind die Koordinaten des jeweiligen Pixels i im Bildraum, wobei der Koordinatenursprung in der Mitte des Bildraumes liegt. w und h sind Breite und Höhe des Bildraumes. Pro Kameraposition (α, β) ergibt sich so ein Parameter p_{cen} , anhand dessen später eine Bewertung aller Kamerapositionen bezüglich der „Zentralität“ der Zielstruktur möglich ist:

$$p_{cen}(\alpha, \beta) = norm_{[0,1]}(d_{cen}) \quad (4.13)$$

4.4.3 Berechnung guter Sichtpunkte zur Laufzeit

Gewichtete Addition der Parameter

Die vorgestellten Parameter werden in Form ihrer Parameterfelder zusammengeführt, um aus dem Ergebnis den besten Sichtpunkt für die Zielstruktur zu bestimmen. Die Parameterfelder werden in ein finales Parameterfeld p_{final} addiert, wobei über einen Wichtungsfaktor w jeder Parameter nach den anwendungsspezifischen Anforderungen gewichtet werden kann (siehe Gleichung 4.14 und Abb. 4.20). Das finale Parameterfeld enthält für jeden Sichtpunkt einen skalaren Ergebniswert, der die Güte des Sichtpunktes widerspiegelt. Der beste Sichtpunkt ist das Maximum im finalen Parameterfeld. Dieser wird dem Nutzer präsentiert, indem die Kamera animiert an diese Stelle innerhalb der Szene bewegt wird. Das geschieht auf der Oberfläche der umgebenden Kugel. Nähere Betrachtungen zur Thematik der Kameraflüge werden in Kapitel 6 vorgenommen.

$$\begin{aligned}
 p_{\text{final}}(\alpha, \beta) = & w_{\text{vis}} * p_{\text{vis}}(\alpha, \beta) + w_{\text{imp}} * p_{\text{imp}}(\alpha, \beta) + \\
 & w_{\text{vis_sta}} * p_{\text{vis_sta}}(\alpha, \beta) + w_{\text{imp_sta}} * p_{\text{imp_sta}}(\alpha, \beta) + \\
 & w_{\text{occ}} * p_{\text{occ}}(\alpha, \beta) + w_{\text{pref}} * p_{\text{pref}}(\alpha, \beta) + \\
 & w_{\text{cam}} * p_{\text{cam}}(\alpha, \beta) + w_{\text{dis}} * p_{\text{dis}}(\alpha, \beta) + \\
 & w_{\text{sil}} * p_{\text{sil}}(\alpha, \beta) + w_{\text{cen}} * p_{\text{cen}}(\alpha, \beta)
 \end{aligned} \tag{4.14}$$

Die Wichtung von Parametern ist ein wesentlicher Bestandteil des Verfahrens. So erlauben die Gewichte auf Basis der gleichen Daten (Sichtbarkeit, Vorzugsblickwinkel, usw.) die Berechnung von auf die jeweiligen Anwendung oder Fragestellung des Arztes zugeschnittenen Sichtpunkten. So spielt der Vorzugsblickwinkel für den planenden HNO-Chirurgen eine wesentlich größere Rolle als für einen Leberchirurgen. Da bei Visualisierungen der Leber oder anderer Organe diese meist alle anderen Strukturen verdeckt, spielt hier der Parameter der verdeckenden Wichtigkeit eine große Rolle. Eine Übersicht über die Wichtungen für verschiedene Anwendungsfelder kann Tabelle 4.2 entnommen werden.

Das Verfahren erlaubt weiterhin nicht nur anwendungsspezifische Wichtungen der einzelnen Parameter, sondern auch solche, die an die anatomische Zugehörigkeit einer Struktur gebunden sind. So ist bei den Gefäßen durch ihre starke Verästelung die Silhouettenlänge als Parameter wichtiger als beispielsweise bei runden, kompakten Strukturen wie Lymphknoten.

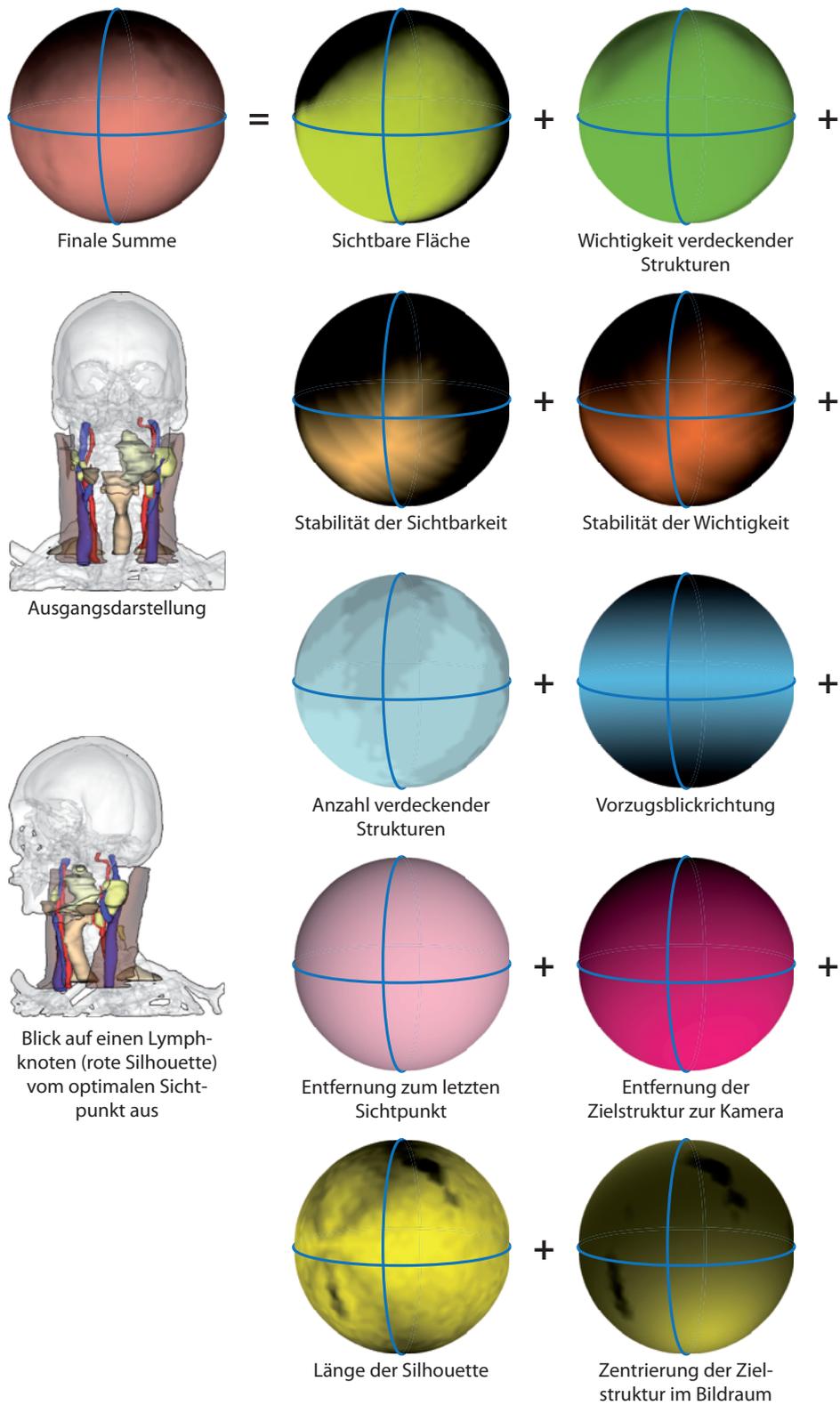


Abbildung 4.20: Gewichtete Addition aller Parameterfelder für einen Lymphknoten (rote Silhouette). Die Farben entsprechen den Farben der Terme in Gleichung 4.14. Helle Bereiche der Parameterfelder repräsentieren gute Sichtpunkte.

Gewicht	Hals	Leber
w_{vis}	0,3	0
w_{imp}	0,5	1
w_{vis_sta}	0,3	0
w_{imp_sta}	0,5	0,5
w_{occ}	0,2	0,1
w_{pref}	0,5	0,2
w_{cam}	0,5	0,5
w_{dis}	0,5	0,5
w_{sil}	0,2	1
w_{cen}	0,5	0,5

Tabelle 4.2: Gewichte der Parameter, wie sie für zwei Szenarien der Hals- und Leberchirurgie genutzt werden. Die Gewichte wurden empirisch aus vielen Gesprächen mit Ärzten und genauen Beobachtungen von 3D-Navigationen gewonnen.

Besondere Funktion des Vorzugsblickwinkels

Eine Besonderheit bei der Ermittlung des Ergebnisfeldes kann unter bestimmten Umständen der Parameter des Vorzugsblickwinkels darstellen. Neben der Bevorzugung bestimmter Blickwinkel kann er auch dazu genutzt werden, um bestimmte Bereiche von Sichtpunkten komplett zu sperren. Wird das finale Parameterfeld abschließend mit dem binarisierten Parameterfeld des Vorzugsblickwinkels p_{pref} multipliziert,¹⁹ so werden Sichtpunkte, die in p_{pref} mit 0 bewertet sind, auch im finalen Parameterfeld mit 0 bewertet und fallen so für eine Wahl als neuer Sichtpunkt aus. Ein solcher Fall tritt bspw. in Lernsystemen auf, in denen der Lernende nur bestimmte, vom Ausbilder oder Autor für gut befundene Blickwinkel präsentiert bekommen soll.

Laufzeit

Die Berechnung des finalen Parameterfeldes und die Wahl des besten Sichtpunktes erfolgt in Echtzeit, so dass der neue Sichtpunkt dem Nutzer direkt auf eine Interaktion hin (z.B. die Selektion der Zielstruktur) angeboten werden kann. Dies ist umso mehr von Bedeutung, als dass der gesamte Vorgang schneller ablaufen sollte, als ihn der Arzt manuell erledigen könnte.

¹⁹Wobei alle Sichtpunkte in p_{pref} mit einem Wert größer 0 mit 1 bewertet werden und nur die mit 0 bewerteten Sichtpunkte unverändert bleiben.

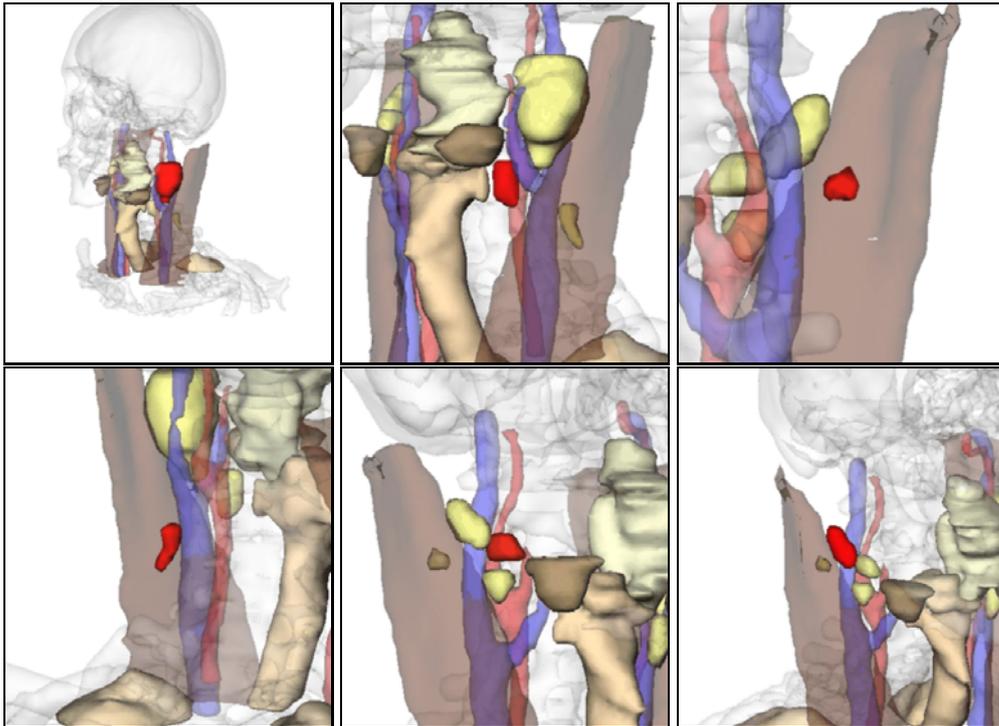


Abbildung 4.21: Auszüge aus einer Animation zur Hervorhebung aller vergrößerten Lymphknoten über automatisch bestimmte gute Sichtpunkte.

4.4.4 Beispiel

Die automatische Bestimmung von Sichtpunkten kommt unter anderem bei der Generierung von Animationen zum Einsatz. In [Abbildung 4.21](#) sind Auszüge aus einer Animation zur Hervorhebung aller vergrößerten Lymphknoten im Hals zu sehen. Dabei wurde für jeden Lymphknoten ein guter Sichtpunkt bestimmt und von der Kamera nacheinander angefahren. Wie solche Animationen weiter optimiert werden können, wird in [Kapitel 6](#) beschrieben.

4.5 Evaluierung des Verfahrens

Das Verfahren zur Sichtpunktbestimmung wurde einer ersten Evaluierung unterzogen, die im Folgenden vorgestellt wird.

4.5.1 Durchführung

In einer internetbasierten Umfrage wurden den Teilnehmern insgesamt neun Zielstrukturen in HNO-Visualisierungen aus jeweils zwei verschiedenen Sichtpunkten präsentiert. Die Sichtpunkte wurden einmal mit dem adaptierten Verfahren der Viewpoint Entropy nach [SBERT u. a. \[2005\]](#) und einmal mit dem neuen Verfahren dieser Arbeit erzeugt (Abb. 4.22). Die Teilnehmer sollten für jeden Sichtpunkt die Sichtbarkeit der Zielstruktur und die Nützlichkeit des Blickwinkels auf einer Skala von 1 (sehr gut) bis 5 (sehr schlecht) bewerten. Der Vergleich mit Sichtpunkten auf der Basis der Viewpoint Entropy wurde gewählt, weil dies die zum Zeitpunkt der Evaluierung beste bekannte Technik zur Bestimmung von Sichtpunkten war. Es wurden statische Bilder der Szenen genutzt, weil dies eine Befragung über das Internet zuließ.²⁰ Dies ermöglichte das Erreichen eines größeren Personenkreises auch unter Personen mit guten medizinischen Vorkenntnissen. Mit einer ortsgebundenen Befragung wäre der Kreis der Testpersonen wesentlich kleiner ausgefallen.

4.5.2 Ergebnisse und Diskussion der Evaluierung

Die 50 Teilnehmer entstammten einem heterogenen Feld, 22 besaßen gute bzw. sehr gute medizinische Vorkenntnisse (z.B. Ärzte). Sie hatten ein Durchschnittsalter von 28 Jahren (19–45), 13 waren Frauen und 37 Männer.

Auch wenn Psychologen oft eine Normalverteilung der Ergebnisse ab einer Probandenzahl von 30 annehmen, haben [BAER u. a. \[2009\]](#) gezeigt, dass dennoch in jedem Fall auf eine Normalverteilung der jeweiligen Stichprobe zu testen ist. Daher wurden die einfaktoriell abhängigen Stichproben für die neu entwickelte Technik sowie die Sichtpunkte mit Viewpoint Entropy mithilfe des Shapiro-Wilk-Tests positiv auf ihre Normalverteilung getestet (siehe Tabelle 4.3). Anschließend konnte mittels t-Test sowohl ein signifikanter Unterschied zwischen der neuen Technik und der Viewpoint Entropy bei der Bewertung der Sichtbarkeit ($p < 0,0001$) wie auch bei der Bewertung des Blickwinkels ($p < 0,0001$) hin zu einer Bevorzugung der neuen Technik festgestellt werden (siehe dazu auch die grafische Darstellung der Ergebnisse in Abbildung 4.23). Die Effektgröße δ zeigt mit $\delta_{\text{sichtb}} = 0,67$ und $\delta_{\text{blick}} = 0,79$ zwar keine sehr großen Unterschiede nach der Einteilung von [COHEN \[1992\]](#) ($\delta > 0,8$). Die Werte liegen aber deutlich über der Marke von $\delta < 0,2$, die [COHEN \[1992\]](#) für geringe Unterschiede ausgemacht hat.

²⁰Es kam ein eigens für diese Evaluierung entwickeltes Framework aus PHP-Seiten und einer MySQL-Datenbank zum Einsatz.

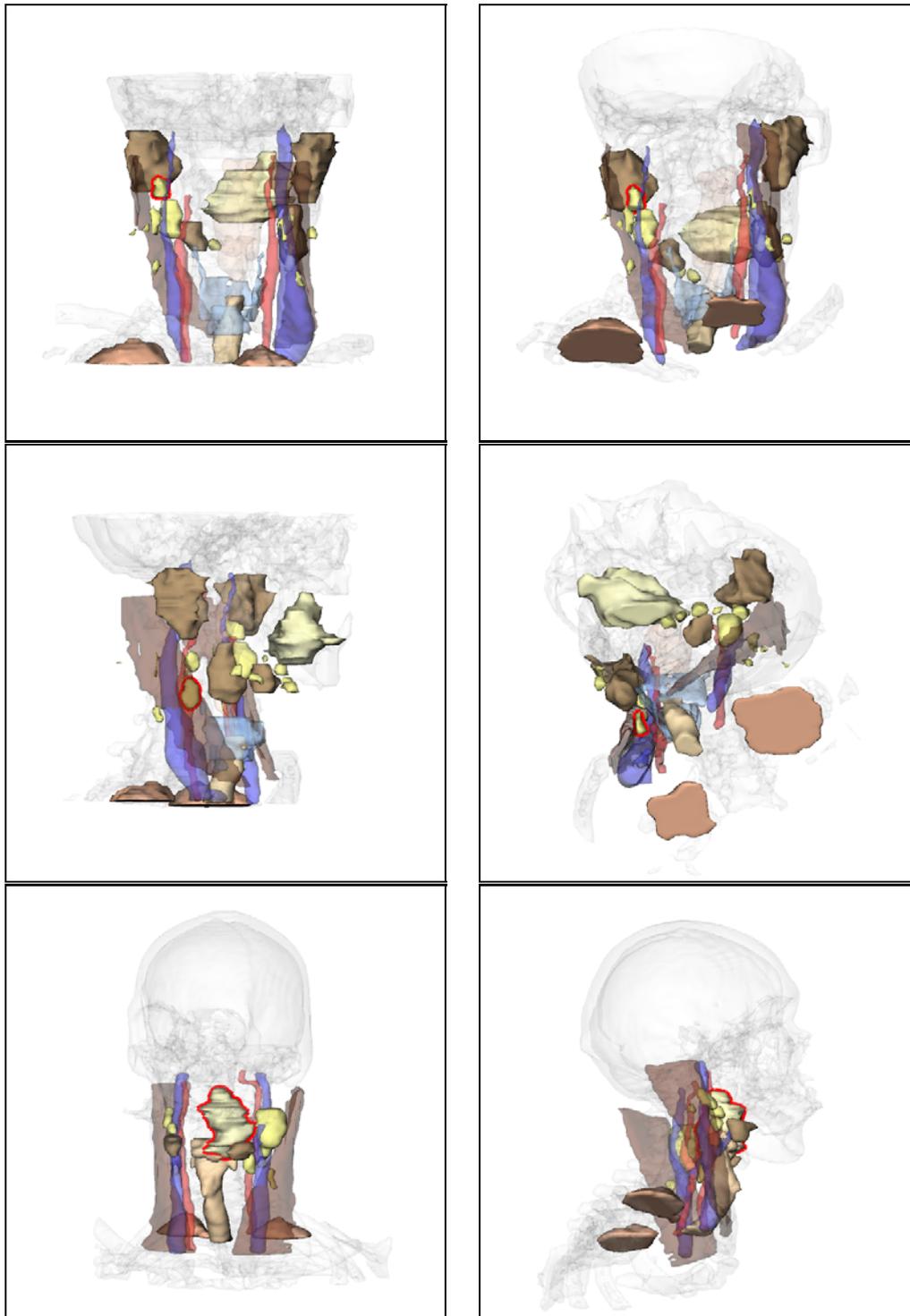
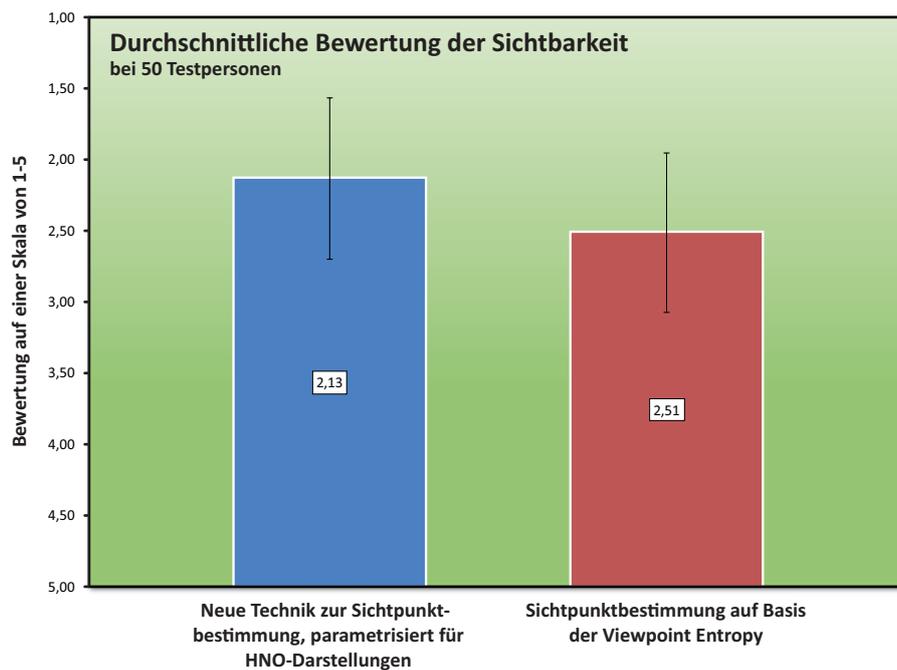
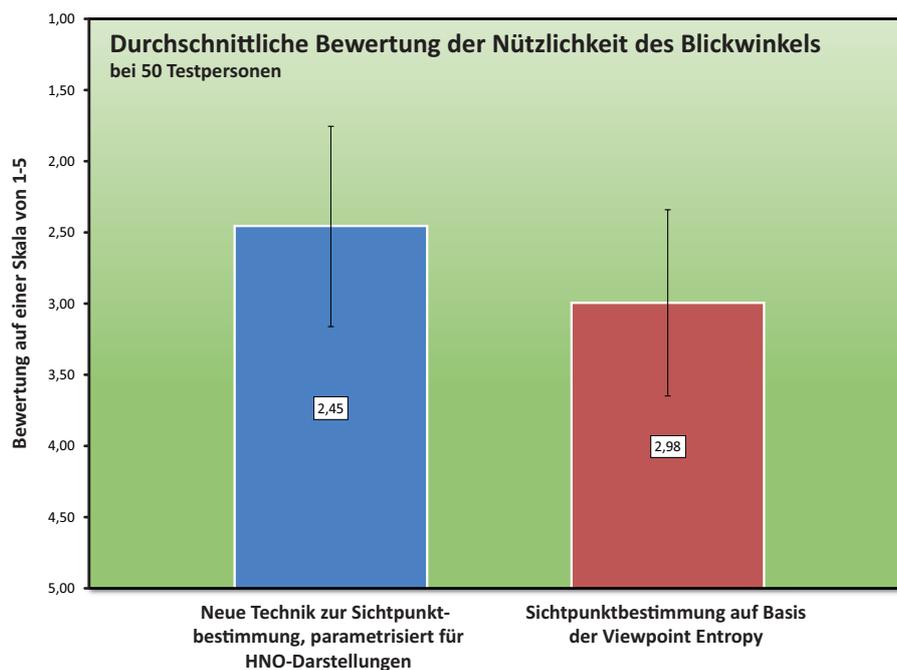


Abbildung 4.22: Beispiele für jeweils zwei verschiedene Sichtpunkte auf eine Zielstruktur (rote Silhouette) in Patientenmodellen des Halses. Der linke Sichtpunkt wurde jeweils mit dem neuen Verfahren, der rechte Sichtpunkt mit dem Verfahren der Viewpoint Entropy bestimmt.



(a)



(b)

Abbildung 4.23: Durchschnittliche Bewertung der Sichtbarkeit einer Zielstruktur (a) bzw. die Einschätzung der Nützlichkeit des Blickwinkels für die Operationsplanung (b) bei allen gezeigten Sichtpunkten auf einer Skala von 1 bis 5. Die blauen Balken repräsentieren dabei die im Rahmen dieser Arbeit neu entwickelte Technik zur Sichtpunktbestimmung. Die roten Balken stehen für Sichtpunkte, die auf Basis der Viewpoint Entropy (VPE) nach Vázquez u. a. [2001] erstellt wurden.

Bewertung der ...	Technik	Normalverteilung (Shapiro-Wilk)	Signifikanz	Effektgröße
... Sichtbarkeit der Zielstruktur	Neue Technik	$p < 0,0001$	$p < 0,0001$	$\delta_{\text{sichtb}} = 0,67$
	VPE	$p < 0,0001$		
... Nützlichkeit des Blickwinkels	Neue Technik	$p < 0,0001$	$p < 0,0001$	$\delta_{\text{blick}} = 0,79$
	VPE	$p < 0,0001$		

Tabelle 4.3: Übersicht über die Evaluierungsergebnisse der Sichtpunkte, ermittelt mit der neuen Technik und einer angepassten Variante der Viewpoint Entropy (VPE).

4.6 Zusammenfassung und Ausblick

In diesem Kapitel wurde ein Verfahren zur automatischen Bestimmung guter Sichtpunkte vorgestellt. Grundlage sind dabei kompakte polygonale Szenen mit vielen Strukturen. Es wurde eine Vielzahl an Parametern identifiziert, die je nach Anwendungsfall unterschiedliche Auswirkungen auf die Güte von Sichtpunkten haben. Jeder der Parameter wird durch ein Parameterfeld repräsentiert, das in einem Vorverarbeitungsschritt oder zur Laufzeit berechnet wird. Über die gewichtete Summe aller Parameterfelder wird der im jeweiligen Kontext beste Sichtpunkt auf eine Struktur in Echtzeit ermittelt. Der Erfolg des Verfahrens konnte in einer ersten Evaluierung gezeigt werden.

Im Folgenden werden in einem kurzen Ausblick neueste Entwicklungen im Bereich der Sichtpunktbestimmung sowie noch offene Fragen diskutiert.

4.6.1 Anwendung des Verfahrens auf direkte Volumendarstellungen

Das vorgestellte Verfahren zur Sichtpunktbestimmung für Oberflächendarstellungen ist nur bedingt auf Volumerending-Darstellungen anwendbar. In der Volumerending-Darstellung ist es der Nutzer gewohnt, die Transferfunktion individuell anzupassen. Dies kann von der Auswahl aus vordefinierten Funktionen bis hin zur kompletten Manipulation der Transferfunktionsparameter durch den Nutzer selbst reichen. In jedem Fall aber führt eine Veränderung an der Transferfunktion zu einer veränderten Darstellung. Dies gilt im besonderen Maße auch für die Sichtbarkeit. Das klassische Volumerending ist nicht strukturbasiert und

orientiert sich nur an den Grauwerten in den Bilddaten. Dadurch kann es bei Veränderungen der Transferfunktion zu mannigfaltigen Überdeckungen der Strukturen durch andere, vor allem auch nicht segmentierte Strukturen kommen. Damit sind die im Pre-Processing berechneten Sichtbarkeitsinformationen, die die Grundlage für einen Großteil der Parameter darstellen, nicht für Volumerending-Darstellungen einsetzbar.

Eine Nutzung der Sichtbarkeitsparameter wäre nur in engen Grenzen denkbar, beispielsweise dann, wenn die Manipulationen und die Art der Transferfunktion lediglich eine Visualisierung der segmentierten Strukturen erlaubt (*tagged volume rendering*). So wären die Sichtbarkeitsinformationen als Parameter zur Sichtbarkeitsbestimmung wieder nutzbar, wenn zusätzlich noch bekannt wäre, welche Strukturen aktuell dargestellt werden. Dies widerspricht aber einem wesentlichen Anliegen des Volumerendings, auch Kontextinformationen um die Strukturen herum sichtbar zu machen. Sowohl [VIOLA u. a. \[2006\]](#) als auch [BORDOLOI & SHEN \[2005\]](#) gehen von festen Transferfunktionen aus, die während der Betrachtung und Interaktion nicht manipuliert werden. Im Gegensatz dazu bietet das hier vorgestellte Verfahren abseits des Volumerendings Sichtpunktberechnungen in Echtzeit auch bei Veränderungen an der Szene, die Transparenz und Sichtbarkeit betreffend, an.

Ohne Probleme einsetzbar sind allerdings die Parameter, die nicht auf Daten der Vorberechnung angewiesen sind. Dazu zählen der Vorzugsblickwinkel, die zentrale Lage im Bildraum, die Entfernung der Struktur zur Viewplane und die Entfernung der neuen Kameraposition zur aktuellen. Besonders der Vorzugsblickwinkel kann für sich allein genommen eine große Bereicherung bei der Kameraführung auch in Volumerending-Darstellungen sein. Bei einer Sichtpunktberechnung auf Basis der genannten Parameter ist im Volumerending allerdings keine letztendliche Sichtbarkeit der Struktur garantiert. Diese hängt zusätzlich von der Transferfunktion ab und muss unabhängig von der Sichtpunktbestimmung gewährleistet werden.

Neuere Entwicklungen versuchen den gewichteten Multi-Parameter Ansatz, der in dieser Arbeit eingeführt wurde, auf Volumendarstellungen zu übertragen. So haben [KOHLMANN u. a. \[2007\]](#) bzw. [KOHLMANN u. a. \[2008\]](#) mit dem LIVESYNC-System versucht, automatisch gute Sichtpunkte auf eine im 2D-Schichtbild selektierte Struktur zu finden. Dazu wurde zunächst mit einer lokalen Formanalyse über ein Region-Growing-Verfahren die grobe Form der Struktur ermittelt: ist sie rundlich, länglich oder flächig. Davon abhängig wurde der erste Parameter der Sichtpunktbestimmung ermittelt, wobei auf flächige und längliche Strukturen möglichst senkrecht geschaut werden sollte. Weitere Parameter waren die Entfernung zum letzten Sichtpunkt, der Vorzugsblickwinkel des Arztes sowie die Sichtbarkeit der Strukturvoxel selbst. Eine Anpassung der Transferfunktion fand nicht statt, so dass die Struktur eventuell von keinem Sichtpunkt aus sichtbar war.

4.6.2 Automatische Parametrisierung von Sichtpunkten

Die Gewichte der verschiedenen Parameter wurden empirisch in Gesprächen gewonnen. Es kann jedoch eingewendet werden, dass eine analytische Bestimmung der Gewichte genauere und bessere Ergebnisse liefern könnte. Dies ist auch unter der Möglichkeit zu bedenken, das Verfahren auf andere chirurgische Disziplinen oder Anwendungen auszudehnen. Ein solches Verfahren könnte eine angemessene Anzahl von für gut befundenen Sichten automatisch analysieren und die Gewichte der jeweiligen Parameter lernen.

Dem ist entgegenzuhalten, dass es bisher keine Untersuchungen gibt, die belegen, dass kleine Verbesserungen an den automatisch gefundenen Sichtpunkten auch einen Gewinn für den Nutzer bringen. Vielmehr müsste ein solcher Gewinn zunächst erkannt werden, um den Aufwand zu rechtfertigen, der mit einer umfangreichen Erfassung guter Sichtpunkte und deren Analyse verbunden wäre. Das vorliegende Verfahren zielt vielmehr darauf ab, automatisch einen Sichtpunkt auf eine Zielstruktur zu finden und weniger dessen letztendliche feine Ausprägung in wenigen Grad nach links oder rechts zu betrachten. Dies wird durch den Eindruck bestätigt, dass auch eine starke Reduzierung der Sichtpunkt-Kandidaten (siehe Abschnitt 4.4.1) zu keinen merklichen Einbußen in der subjektiven Bewertung der gefundenen Sichtpunkte führt.

5

Anwendungen für automatische Sichtpunktbestimmungen

Aus der im vorherigen Kapitel vorgestellten Lösung zur automatischen Bestimmung von Sichtpunkten ergibt sich eine ganze Reihe von Anwendungsmöglichkeiten. Die für die Bestimmung der Sichtpunkte generierte Datenbasis bildet eine gute Grundlage für weitere, spezialisiertere Verfahren zur Steuerung der Kamera und zur dynamischen Anpassung der Szene. Eine Optimierungsmöglichkeit bei der Exploration vieler einzelner Strukturen bietet das Zusammenfassen von Sichtpunkten (Clustering) (Abschnitt 5.1). Eine Spezialanwendung im Bereich der Risikoabschätzung in der Umgebung kritischer Strukturen ist die Sicht auf minimale Distanzen zwischen Strukturen, die in Abschnitt 5.2 vorgestellt wird. Wie sich die automatische Sichtpunktbestimmung im intra-operativen Umfeld nutzen lässt, wird in Abschnitt 5.3 dargelegt. Abschließend wird in Abschnitt 5.4 die Nutzung der Sichtbarkeitsinformationen zur dynamischen Szenenanpassung vorgestellt.

Neben den in diesem Kapitel präsentierten Anwendungen und Erweiterungen des Verfahrens zur automatischen Sichtpunktbestimmung kommen das Verfahren und seine Datengrundlage weiterhin bei der Pfadplanung für Kameraflüge (Kapitel 6) und der Animationsgenerierung (Kapitel 8) zur Anwendung. Diese werden in gesonderten Kapiteln behandelt.

Eigene Publikationen

Dieses Kapitel basiert u.a. auf folgenden eigenen Publikationen:

MÜHLER u. a. [2007b] Mühler, K.; Neugebauer, M.; Tietjen, C. & Preim, B.: *Viewpoint Selection for Intervention Planning*. In: **IEEE/Eurographics Symposium on Visualization (EuroVis)**, 2007

MÜHLER u. a. [2008a] Mühler, K.; Hansen, C.; Neugebauer, M. & Preim, B. *Automatische Kamerapositionierung für intra-operative Visualisierungen in der onkologischen Leberchirurgie*. In: **Bildverarbeitung für die Medizin (BVM)**, 2008

MÜHLER & PREIM [2010a] Mühler, K. & Preim, B. *Günstige Kamerapfade für medizinische Animationen*. In: **Bildverarbeitung für die Medizin (BVM)**, 2010

5.1 Gute Sichten auf mehrere Strukturen (Clustering)

Sollen mit der Kamera viele Strukturen nacheinander aus einer guten Sicht gezeigt werden, können der zurückzulegende Kamerapfad und damit die gesamte Animation schnell zu lang werden. Daher ist es angebracht, Sichtpunkte in der Art zusammenzufassen, dass möglichst mehrere Strukturen von einem Sichtpunkt aus gleichzeitig betrachtet werden können. Ein solcher Fall kann zum Beispiel die Inspektion einer Vielzahl von Lymphknoten im Halsbereich sein. Hier muss nicht jeder Lymphknoten individuell von der Kamera angeflogen werden, wenn mehrere Lymphknoten von einer gemeinsamen Kameraposition aus gut sichtbar sind. Ein weiteres Beispiel sind Sichten auf mehrere Rundherde, beispielsweise in der Lunge. Es wurde daher ein Algorithmus entwickelt, der verschiedene Sichtpunkte zusammenfasst (siehe Algorithmus 5.1):

- Grundlage des Verfahrens sind die Parameterfelder der Sichtbarkeit jeder Struktur,²¹ wie sie im vorherigen Kapitel eingeführt wurden und die für jede Kameraposition auf der szenenumgebenden Kugel die Sichtbarkeit der Struktur gespeichert haben.
- Diese Parameterfelder werden an einem Schwellenwert von 70% Sichtbarkeit²² binarisiert, um nur noch solche Positionen mit einer Mindestsichtbarkeit zu erhalten (sichtbar=1, nicht sichtbar=0).
- Anschließend werden die Strukturen in einer Gruppe zusammengefasst, bei denen nach der Multiplikation ihrer binarisierten Parameterfelder wenigstens eine Position mit 1 bewertet wird. Dies bedeutet, dass mindestens

²¹Anstelle der reinen Sichtbarkeit wurde das Feld der Sichtbarkeit unter Berücksichtigung verdeckender Strukturen genutzt. Hier wird im Folgenden jedoch weiterhin aus Gründen der Verständlichkeit von der einfachen Sichtbarkeit gesprochen.

²²Der Schwellenwert wurde empirisch an einer Vielzahl von Beispielen bestimmt und hat sich in der bisherigen Praxis als robust erwiesen.

Eingabe : Alle Parameterfelder PM

Ausgabe : n Listen von Strukturen S die jeweils zusammen sichtbar sind und dazugehörige n Parameterfelder CLUSTER

```

tresh=0.7 ; // Schwellenwert zur Binarisierung
n=0 ; // Initialisierung des Counters zum Zählen der Cluster
für jedes  $PM_i \in PM$  tue
   $PMBIN_i \leftarrow bin(PM_i, tresh)$  ; // Binarisierung der PM
  found = false;
  für jedes  $CLUSTER_j \in CLUSTER$  tue
    wenn  $1 \in PMBIN_i * CLUSTER_j$  dann
       $CLUSTER_j \leftarrow PMBIN_i * CLUSTER_j$ ;
      Füge  $i$  in die Liste  $S_j$  ein;
      found = true;
    Ende
  Ende
  wenn not found dann
     $CLUSTER_{n+1} \leftarrow PMBIN_i$ ;
    Füge  $i$  in die Liste  $S_{n+1}$  ein;
  Ende
Ende

```

Algorithmus 5.1: Algorithmus zum Clustering von Kamerapositionen.

eine Position existiert, von der aus alle Strukturen einer Gruppe aus zu mindestens 70% sichtbar sind (Abb. 5.1).

5.2 Gute Sichten auf minimale Distanzen zwischen Strukturen

Eine Erweiterung des in Kapitel 4 beschriebenen Verfahrens stellt die Identifizierung von Sichtpunkten auf Strukturrelationen, speziell auf minimale Abstände zwischen zwei Strukturen, dar. Der minimale Abstand zwischen zwei Strukturen ist besonders bei der Risikoabschätzung von entscheidender Bedeutung. So ist für einen Leberchirurgen der Abstand eines Tumors zu großen Gefäßen, wie der mittleren Lebervene, von Interesse, um beurteilen zu können, ob dieses Gefäß bei zu geringem Abstand zum Tumor bei einer Resektion mit entnommen und anschließend rekonstruiert werden muss. Ein HNO-Chirurg muss den Abstand zwischen Lymphknoten und vitalen Strukturen, wie Muskeln oder Gefäßen, gut beurteilen können. Der minimale Abstand wird für die hier vorliegenden Fälle mit dem Verfahren von [PREIM u. a. \[2002\]](#) berechnet und durch einen Pfeil mit Maßzahl visualisiert (Abbildungen 5.2 und 5.3). Ein guter Sichtpunkt auf einen (minimalen) Abstand zweier Strukturen ist nicht ohne Weiteres mit dem Basisalgorithmus aus Kapitel 4 zu bestimmen. Es bedarf einer Erweiterung, die im Folgenden beschrieben wird.

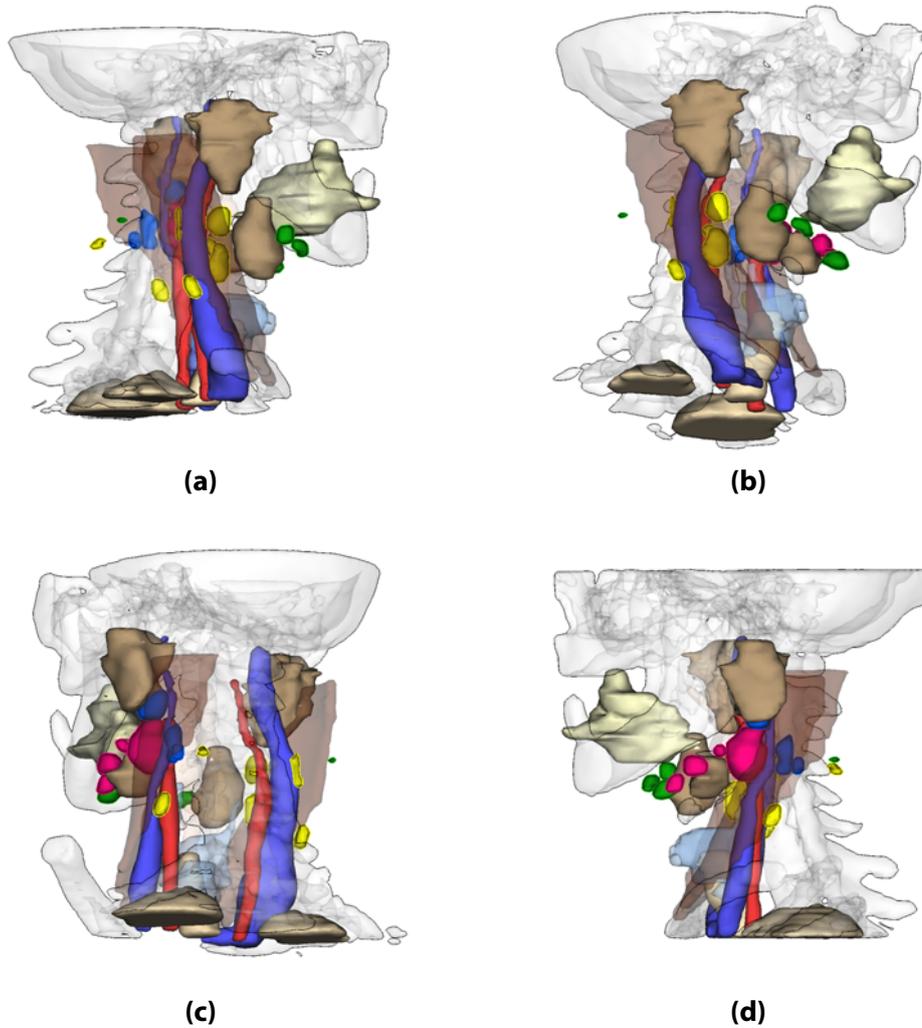


Abbildung 5.1: Vier Sichten ((a) gelb, (b) grün, (c) blau, (d) lila), die alle 18 Lymphknoten einer Darstellung zur Planung von Halslymphknotenausräumungen zeigen. Die Reduzierung von 18 auf 4 Sichten verkürzt Animationen erheblich und reduziert beispielsweise auch die Anzahl von Screenshots zur Dokumentation.

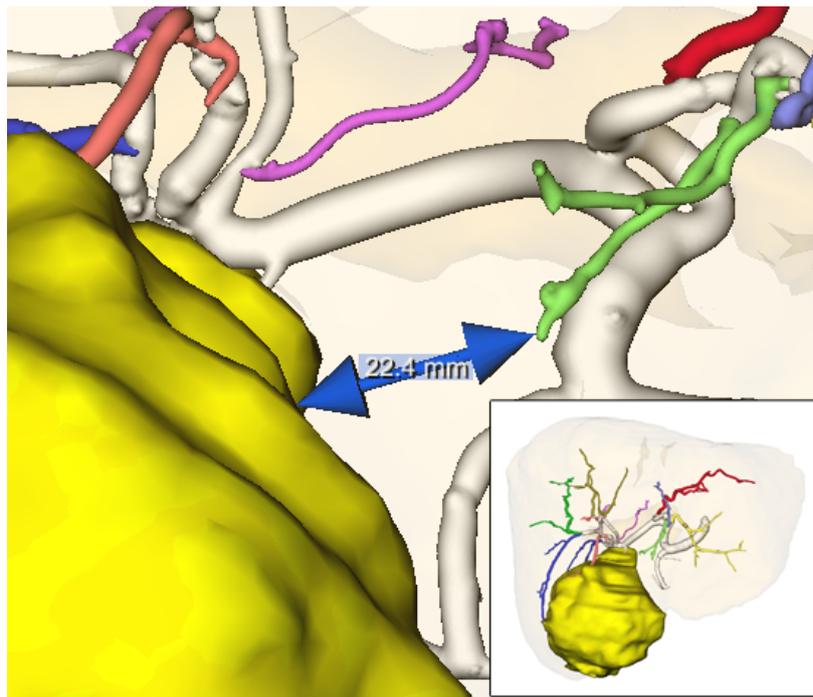


Abbildung 5.2: Gute Sicht auf einen minimalen Abstand zwischen einem Tumor und einem Gefäßast (grün) in der Leber mit der dazugehörigen Überblicksdarstellung.

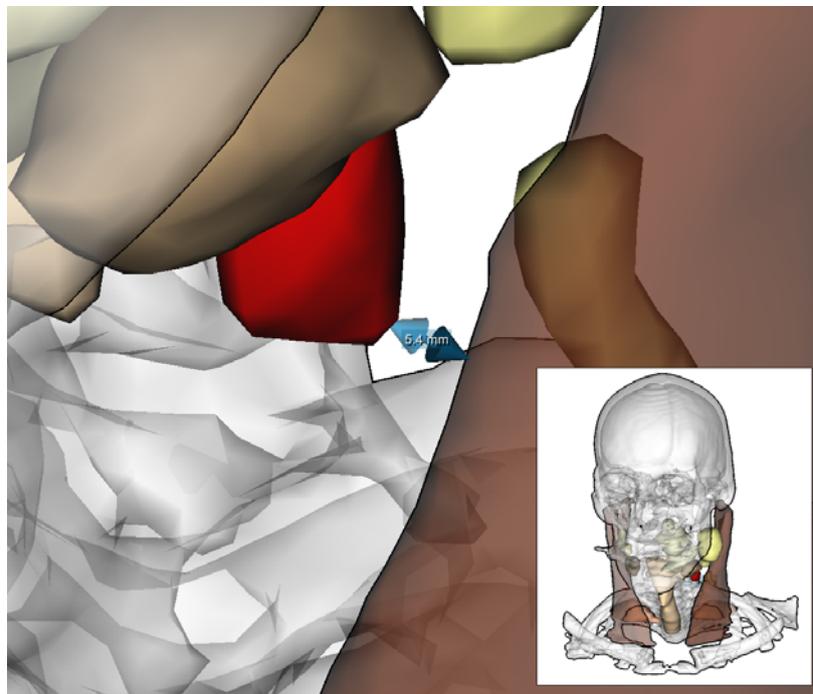


Abbildung 5.3: Gute Sicht auf einen minimalen Abstand zwischen einem Lymphknoten und einem Muskel im Halsbereich mit der dazugehörigen Überblicksdarstellung.

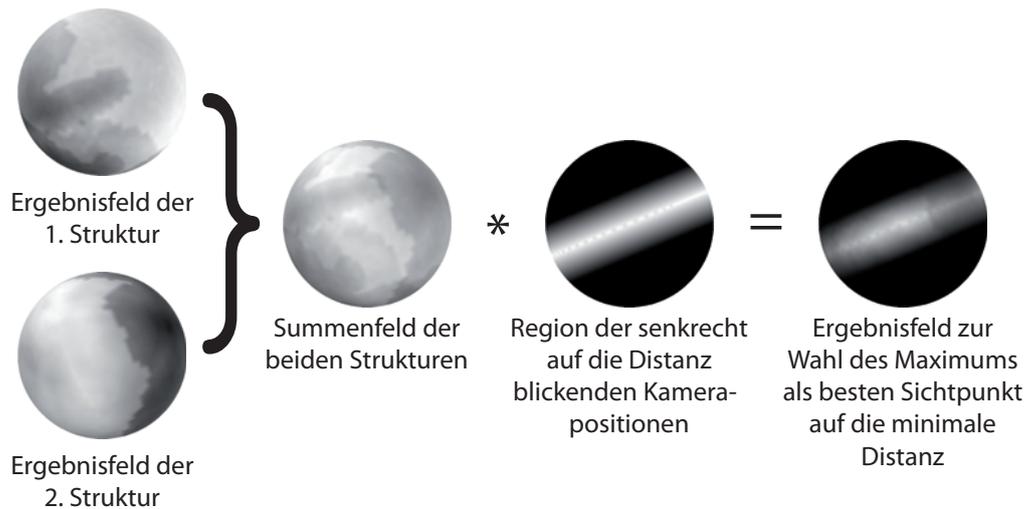


Abbildung 5.4: Zur Ermittlung eines guten Sichtpunkts senkrecht auf eine (minimale) Distanz wird zunächst ein Parameterfeld aus den Sichtbarkeitswerten der beiden beteiligten Strukturen gebildet. Dieses wird anschließend mit einem Parameterfeld der senkrechten Blickrichtungen multipliziert, um mögliche Positionen auf diesen Bereich einzuschränken.

Die beste Sicht auf einen minimalen Abstand hat man, wenn man senkrecht auf ihn blickt. Weiterhin sollten die beiden betroffenen Strukturen gut sichtbar sein. Es wird daher in einem ersten Schritt mit dem in Abschnitt 5.1 beschriebenen Verfahren ein Parameterfeld p_{pair} berechnet, welches die Güte der Sichtpunkte auf beide Strukturen beschreibt (Abb. 5.4). Ist der minimale Abstand als Vektor \vec{d}_{min} bekannt, kann ein Parameterfeld p_{senkr} ähnlich dem des Vorzugsblickwinkels bestimmt werden, in welchem alle Sichtpunkte mit 1 bewertet sind, die möglichst senkrecht auf \vec{d}_{min} blicken. Zu 0 abfallende Werte werden den Sichtpunkten entsprechend der Abweichung vom senkrechten Blick zugewiesen (Abb. 5.4).

Multipliziert man die Parameterfelder p_{pair} und p_{senkr} zu p_{minDist} :

$$p_{\text{minDist}} = p_{\text{pair}} * p_{\text{senkr}} \quad (5.1)$$

so erhält man im Ergebnis ein Feld, welches mit Werten >0 Sichtpunkte enthält, die möglichst senkrecht auf den minimalen Abstand blicken, verbunden mit ihrer Güte bezüglich der Sichtbarkeit der zwei Zielstrukturen. Ein einfacher Weg wäre die Wahl des Maximums von p_{minDist} als besten Sichtpunkt. Die Sichtbarkeit dieses Abstandes, respektive des Pfeils, ist jedoch nicht garantiert. In äußerst seltenen Fällen kann eine Struktur gerade den Pfeil verdecken, während die beiden Zielstrukturen sichtbar sind. Daher wird für einen gewählten Sichtpunkt abschließend ein Strahl in die Szene geschickt, um zu testen, ob zwischen Sichtpunkt und Pfeil wenig transparente Strukturen die Sicht verdecken. Ist dies der Fall, wird der nächst bessere Sichtpunkt in p_{minDist} gewählt und wie beschrieben verfahren.

Sollte der seltene Fall eintreten, dass der minimale Abstand von keiner Position aus möglichst senkrecht betrachtet werden kann, so werden die Informationen

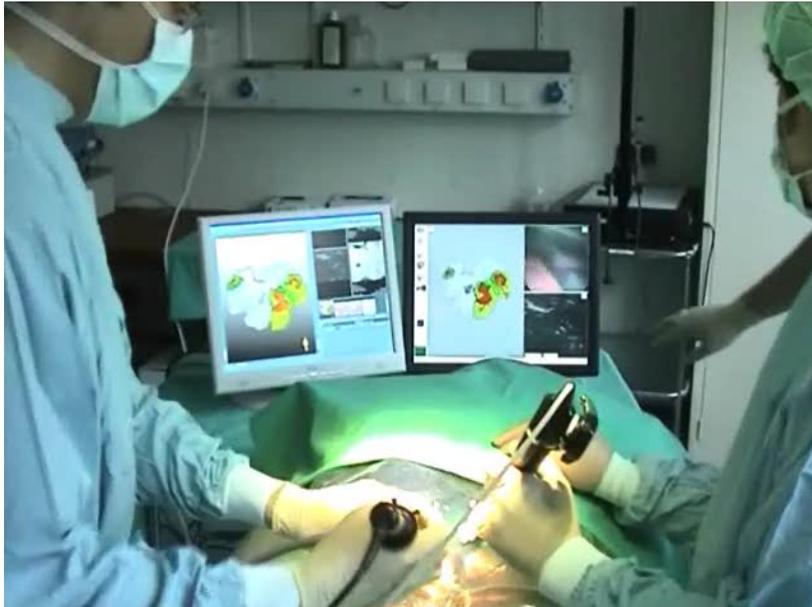


Abbildung 5.5: Intra-operative Visualisierung von Planungsdaten. Planungsmodelle und intra-operativer Ultraschall werden dem Chirurgen in einer Darstellung fusioniert präsentiert (aus [Mühler u. a. \[2008a\]](#)).

aus den zugrunde liegenden Sichtbarkeitsdaten zur Bestimmung der verdeckten Strukturen genutzt. Es wird ein Sichtpunkt mit senkrechtem Blick auf die minimale Distanz gesucht, bei dem möglichst wenig und zugleich unwichtige, verdeckende Strukturen zwischen der Kamera und dem beteiligtem Strukturpaar liegen. Diese werden dann mittels einer kurzen Animation ausgeblendet und so die Sicht auf die minimale Distanz gewährt.

5.3 Automatische Bestimmung von Sichtpunkten im intra-operativen Szenario

Zur Navigationsunterstützung während komplexer leberchirurgischer Eingriffe werden dem Chirurgen zunehmend navigierter Ultraschall gemeinsam mit dreidimensionalen Planungsmodellen auf einem Bildschirm direkt am Situs dargeboten (Abb. 5.5) und ([LANGE u. a. \[2004\]](#) sowie [MARKERT u. a. \[2006\]](#)). Im Verlauf eines Eingriffes kommt es häufig vor, dass sich die dargestellte Planungsszene durch intra-operativ detektierte Metastasen verändert ([HANSEN u. a. \[2007a\]](#)). Um dem Chirurgen jederzeit eine effektive Sicht auf die Daten zu gewährleisten, ist eine Neupositionierung der Kamera in vielen Fällen notwendig. Hierfür musste der Chirurg bisher aufwendig mit einer Maus in der Szene navigieren und wurde so unnötig abgelenkt.

Im Bereich der intra-operativen Darstellung von dreidimensionalen Planungsdaten wurden in den letzten Jahren eine Reihe an Arbeiten veröffentlicht ([LANGE](#)

u. a. [2004], HANSEN u. a. [2007b], NIMSKY u. a. [2006]). Bisher existiert jedoch kein Ansatz, der eine automatische Neupositionierung der Kamera während eines chirurgischen Eingriffes in Abhängigkeit von klinisch relevanten Parametern erlaubt.

Das im Folgenden präsentierte Verfahren nutzt die vorberechneten Sichtbarkeitsinformationen aus Kapitel 4, erweitert diese und ermittelt daraus automatisch die günstigste Kameraposition in einem intra-operativen Szenario. Ziel des Verfahrens ist es, einem Chirurgen während einer OP möglichst gute Sichten auf die fusionierten, dreidimensionalen Planungsdaten anzubieten. Eine gute Sicht wird dabei durch mehrere Parameter beeinflusst: So sollen mehrere Strukturen entsprechend ihrer aktuellen Bedeutung in einer Sicht erkennbar sein. Außerdem sollen neu hinzukommende Objekte, die erst intra-operativ in die Planungsdaten eingezeichnet werden, berücksichtigt werden.

Einbeziehung dynamischer Objekte in die Kameraberechnung

Das im vorherigen Kapitel vorgestellte Verfahren erlaubt es nicht, nach der Vorberechnung neu zur Szene hinzugekommene Objekte in die Berechnung der Kameraposition einzubeziehen. Es müssten mit den neuen Objekten wiederum die Vorberechnungen zur Sichtbarkeitsbestimmung durchlaufen werden. Um dies zu vermeiden, aber dennoch die Einbeziehung während der Operation eingezeichneter Metastasen zu ermöglichen, werden so genannte Stellvertreterobjekte (SO) genutzt. Dies sind Objekte, die schon zum Zeitpunkt der Vorberechnungen erzeugt werden und so als Vertreter dynamischer Objekte zur Sichtbarkeitsbestimmung herangezogen werden können. Zur Erzeugung der SO wird das Parenchymobjekt der Leber in ca. 30 würfelförmige Teilobjekte unterteilt (Abb. 5.6(a)). Die Größe der SO wurde empirisch ermittelt und hat sich als ausgewogen im Hinblick auf die zu erwartenden Größen der neuen Metastasen sowie der Vorberechnungszeit bei zu vielen SO erwiesen.

Anstelle einer neu hinzukommenden Metastase werden die Sichtbarkeitsinformationen der SO in deren Umgebung genutzt. Dabei wird das in Abschnitt 5.1 beschriebene Verfahren zur Sichtpunktberechnung für mehrere Objekte genutzt. Die SO für eine neue Metastase gehen entsprechend gewichtet der Größe ihrer Überlagerung mit der neuen Metastase in die Berechnung ein. So werden nur die SO genutzt, welche die neue Metastase schneiden.

Zusätzlich zu dieser Kameraposition werden weitere Positionen ermittelt, die jeweils eine gute Sicht auf *einzelne Metastasen* bieten. Zusammen werden diese Kamerapositionen als Vorschau dem Chirurgen zur Auswahl präsentiert. Dieser kann dann die gewünschte Kameraposition bequem per Touchscreen auswählen.

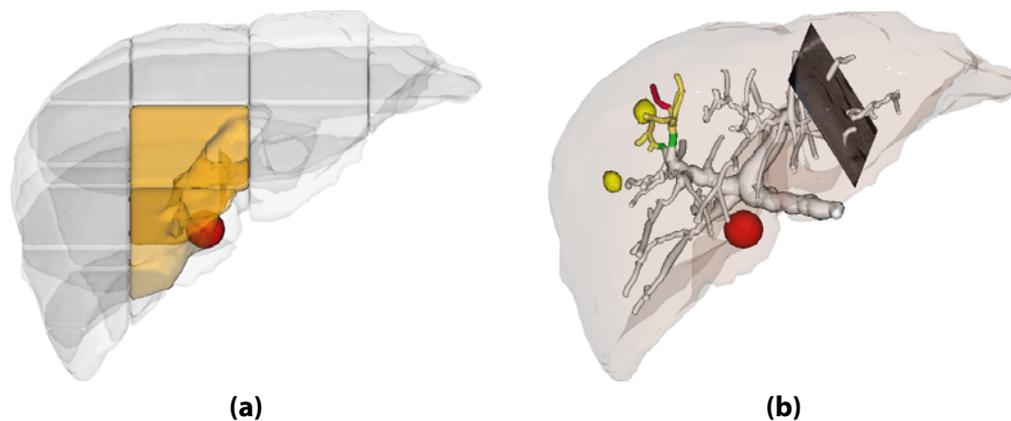


Abbildung 5.6: (a) Darstellung der Stellvertreterobjekte für die neue Metastase. Zu erkennen sind zusätzlich alle würfelförmigen Teilobjekte. (b) Optimale Sicht auf die prä-operativ detektierten Metastasen, den portalvenösen Gefäßbaum und die intra-operativ gefundene Metastase (rote Kugel). Ebenfalls zu erkennen ist das fusionierte Ultraschall-Bild.

5.4 Interaktives Blenden

Neben der Berechnung von einzelnen guten Sichtpunkten auf Strukturen finden die gewonnenen Informationen zur gegenseitigen Verdeckung der Strukturen bei einer neuen Interaktionstechnik Anwendung. Bei der Exploration einzelner Strukturen wird die Kamera durch den Nutzer meist um die Zielstruktur bewegt (*scene-in-hand* Metapher). Dabei wird die Zielstruktur zwangsläufig von anderen Strukturen ganz oder teilweise verdeckt. Dies kann den Explorationsprozess erheblich behindern. Daher ist es wünschenswert, verdeckende Strukturen während der Interaktion automatisch auszublenden.

5.4.1 Ermittlung verdeckender Strukturen

Um ein automatisches Aus- und wieder Einblenden der verdeckenden Strukturen zu erreichen, wird während der Interaktion ständig in drei Schritten überprüft, ob Strukturen die Zielstruktur verdecken und ausgeblendet werden müssen:

1. Ermittlung der Kameraposition in den vorberechneten Daten, die der aktuellen Position in ihrer Blickrichtung am nächsten kommt.
2. Ermittlung verdeckender Strukturen und Grad der Verdeckung aus den Parameterfeldern.
3. Ausblenden verdeckender Strukturen und Einblenden nicht mehr verdeckender Strukturen.

Die Annahme, dass zur Ermittlung verdeckender Strukturen diejenige vorberechnete Kameraposition herangezogen werden kann, deren Blickrichtung der aktuellen Kameraposition am ähnlichsten ist, wird durch die Tatsache unterstützt, dass eine orthogonale Projektion verwendet wird. Damit wird aus Sicht der aktuellen Kameraposition die Zielstruktur von den gleichen Strukturen verdeckt wie aus Sicht einer Kandidatenposition mit gleicher Blickrichtung.

5.4.2 Visualisierung in der Interaktion

Verdeckt eine Struktur die Zielstruktur, so wird deren Darstellungsstil angepasst. Dabei wird die verdeckende Struktur transparent dargestellt. Um die Struktur dennoch als Kontextinformation nutzen zu können, wird sie mit einer Silhouette versehen. Verdeckt eine Struktur nicht mehr die Zielstruktur, so wird sie wieder in ihrem ursprünglichen Darstellungsstil angezeigt (Abb. 5.7). Dieser Prozess erfolgt animiert, um einen weichen Übergang zwischen den Darstellungsstilen zu erhalten.

Eine Variante des interaktiven Ausblendens besteht in der Veränderung des Darstellungsstils bei Verdeckungen in Abhängigkeit vom Grad der Verdeckung. Dabei wird die Transparenz der verdeckenden Struktur davon abhängig gemacht, wie groß der Anteil der durch sie verdeckten Fläche an der gesamt sichtbaren Fläche der Zielstruktur ist. So wird bei kleinen Verdeckungen, die meist an den Rändern der Zielstruktur auftreten, die Transparenz der verdeckenden Strukturen nur geringfügig angehoben. Folglich werden in vielen Fällen unnötig umfassende Wechsel der Darstellungsstile der verdeckenden Strukturen vermieden.

Das Verfahren offenbart gute Anwendungsmöglichkeiten, wenn nicht zu viele verdeckende Strukturen als Kontextinformation erhalten werden müssen. Werden zu viele Kontextstrukturen als Silhouette oder semi-transparent vor der Zielstruktur dargestellt, so kann dies verwirrend wirken (Abb. 5.8(a)). Hier bieten sich Verfahren wie *Cut-Aways* oder *Sections Views* für eine bessere Darstellung an (Abb. 5.8(b)).

5.5 Zusammenfassung

In diesem Kapitel wurden vier Erweiterungen des Verfahrens zur automatischen Sichtpunktbestimmung vorgestellt. Mit dem Clustering von Kamerapositionen ist es möglich, Animationen in ihrer Länge zu verkürzen, wenn mehrere Strukturen von einem gemeinsamen Sichtpunkt aus gut sichtbar sind. Die automatische Generierung von guten Sichten auf minimale Distanzen ist eine gute Ergänzung bei der Exploration kritischer Strukturen. Das Verfahren zur automatischen Bestimmung von Sichtpunkten auf dynamische Strukturen zeigt zum einen die Erweiterbarkeit des Ursprungsverfahrens auch auf solche sehr speziellen Bereiche

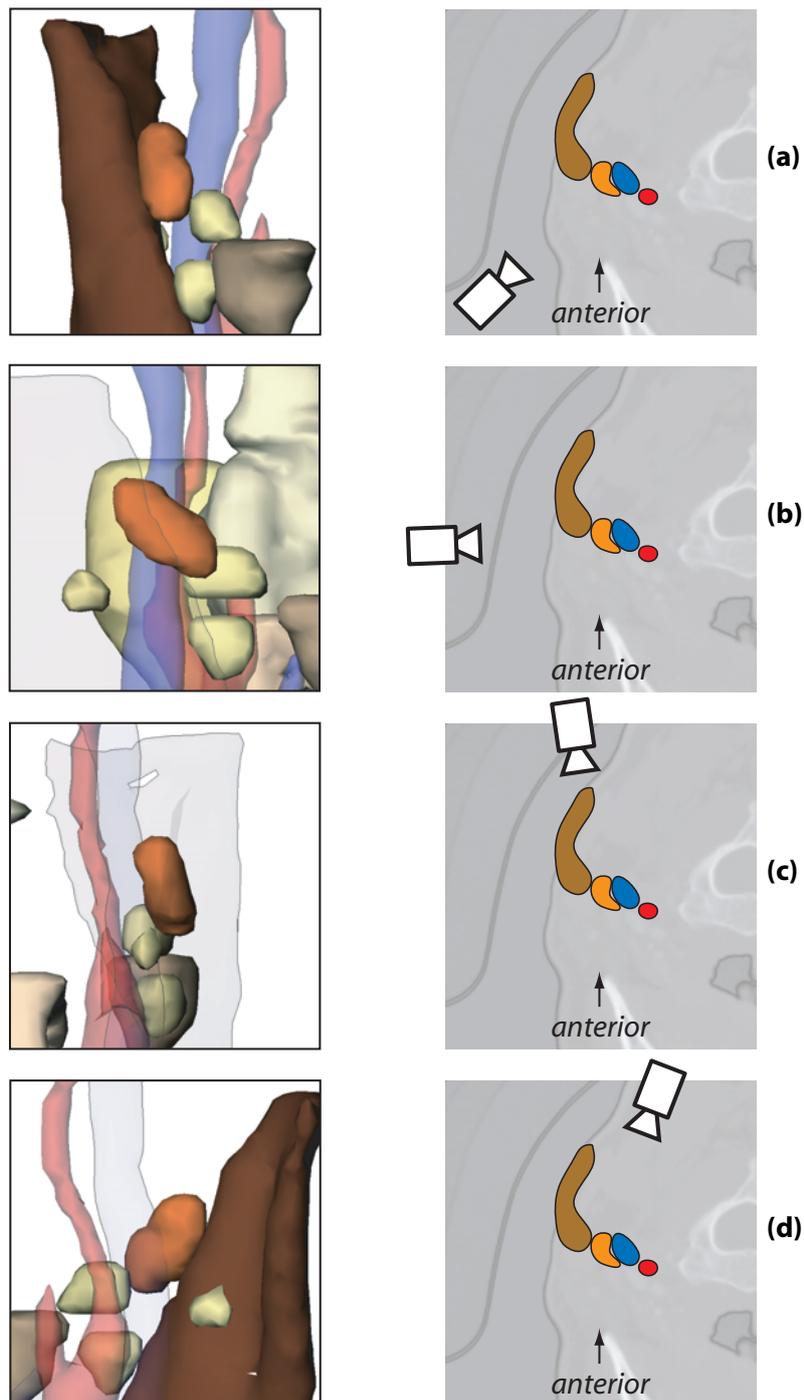


Abbildung 5.7: Verschiedene Stufen des interaktiven Blendens. Die Zielstruktur ist ein vergrößerter Lymphknoten (orange). Der Nutzer bewegt die Kamera um diesen Lymphknoten herum. **(a):** Der Lymphknoten ist ohne Verdeckungen sichtbar. **(b):** Ein Muskel (braun) verdeckt den Lymphknoten und wird daher transparent mit einer Silhouette dargestellt. **(c):** Ein Gefäß (blau) verdeckt ebenfalls den Lymphknoten und wird ausgeblendet. **(d):** Der Muskel verdeckt den Lymphknoten nicht mehr und wird wieder eingblendet.

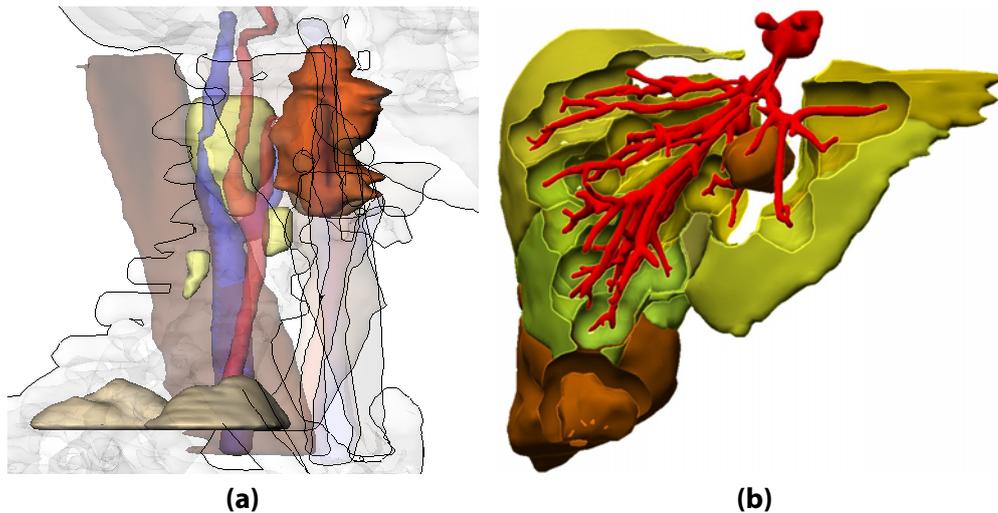


Abbildung 5.8: (a): Ein schlechtes Beispiel für viele verdeckende Strukturen als Silhouetten. (b): *Section View* auf eine intra-hepatische Metastase mit umliegenden Gefäßen (Abbildung (b) mit freundlicher Genehmigung von Christoph Kubisch).

der Anwendung, zum anderen aber auch die Grenzen des Verfahrens – für dynamische Strukturen war das Ursprungsverfahren nie gedacht und für eine breite Anwendung in diesem Bereich müssten verschiedene Überlegungen nochmals völlig neu betrachtet werden. Das Verfahren des interaktiven Blendens zeigt, dass die Sichtbarkeitsinformationen auch über die reine Bestimmung einzelner Sichtpunkte hinaus zu einer Anwendung führen können. Die neue Interaktionstechnik bettet sich harmonisch in eine Applikation ein und führt zu einer neuen (in den dargelegten Grenzen) sinnvollen Unterstützung des explorierenden Chirurgen.

6

Günstige Kamerafahrten für Animationen chirurgischer Szenen

Die Verbindung guter Sichtpunkte durch Kamerapfade spielt eine wichtige Rolle im Gesamtbild einer Animation. Dieses Kapitel analysiert daher die speziellen Anforderungen an einen Kamerapfad in Darstellungen geometrischer Modelle der Patientenanatomie und erarbeitet Lösungen zur Bewertung guter Kamerapfade sowie Verfahren zu ihrer automatischen Berechnung. Im Unterschied zu vielen anderen aktuellen Methoden der Pfadplanung liegt die Besonderheit der hier behandelten Szenen in ihrer Kompaktheit und externen Betrachtung – im Unterschied zu bspw. einer Pfadplanung durch eine 3D-Computerspielwelt.

Die Bewegung der Kamera ist wesentlich zur Unterstützung der räumlichen Wahrnehmung einer Szene. Rein statische Darstellungen vermitteln nur einen Teil der räumlichen Zusammenhänge – erst die Bewegung der Kamera erzeugt durch sich verändernde Verdeckungen und die unterschiedliche Bewegung der weit entfernten und nah am Betrachter liegenden Objekte einen ganzheitlichen, räumlichen Eindruck. Diese Bewegung der Kamera kann durch eine direkte Manipulation des Betrachters hervorgerufen werden, aber auch durch eine automatische Kamerafahrt entlang eines vorgegebenen Pfades. Hinzu kommt, dass durch eine kontinuierliche Veränderung der Kameraposition im Vergleich zu einem statischen Umschalten zwischen zwei Sichten der Betrachter die beiden Sichten in Bezug setzen und die Lage der neuen Sicht sofort erfassen kann. Der Betrachter kann die kontinuierliche Veränderung der Darstellung beobachten und muss keine abrupten Änderungen interpretieren. Daher ist es wichtig, dass der Betrachter auch während einer Kamerabewegung stets die Orientierung behält, also weiß, wo sich die Kamera im Bezug zur Szene gerade befindet.

Gliederung des Kapitels

Nach einer Motivation der Anforderungen an Kamerafahrten (Abschnitt 6.1) werden verwandte Arbeiten mit Blick auf deren Einsatzmöglichkeiten für die hier genutzten Visualisierungen vorgestellt (Abschnitt 6.2). Im Kern des Kapitels wird ein neues Verfahren zur Kamerapfadplanung (Abschnitt 6.3) vorgestellt.

Eigene Publikationen

Dieses Kapitel basiert u.a. auf der folgenden eigenen Publikation:

MÜHLER & PREIM [2010a] Mühler, K. & Preim, B. *Günstige Kamerapfade für medizinische Animationen*. In: **Proc. of Bildverarbeitung für die Medizin (BVM)**, 2010

6.1 Anforderungen an Kamerapfade

Nicht jeder beliebige Pfad zwischen zwei Sichten ist als gleich gut einzustufen. Daher werden an einen guten Kamerapfad folgende Anforderungen gestellt:

1. Der Betrachter muss zu jeder Zeit wissen, wo sich die Kamera gerade befindet.
2. Der Pfad sollte möglichst kurz sein.
3. Die Bewegung der Kamera sollte flüssig (nicht ruckartig) und angenehm sein. Dies gilt sowohl für die Geschwindigkeit der Kamera, als auch für Richtungsänderungen.
4. Während der Kamerafahrt sollten keine ungewohnten Sichten eingenommen werden.
5. Es sollte stets ein Maximum an Informationen präsentiert werden.

Es ist offensichtlich, dass nicht alle fünf Kriterien in gleichem Maße voll erfüllt werden können. Ein Pfad, der den größten Informationsgehalt bietet (Kriterium 5), ist oft nicht der kürzeste Pfad (Kriterium 2), kann zu ungewohnten Sichten führen (Kriterium 4) und ruckartige Richtungsänderungen erfordern (Kriterium 3). Im Gegensatz dazu kann eine Vermeidung ungewohnter Sichten (Kriterium 4) dazu führen, dass der Betrachter zu jeder Zeit weiß, wo sich die Kamera im Raum befindet (Kriterium 1).

Für die im Rahmen dieser Arbeit betrachteten medizinischen Visualisierungen (kompakt, von außen betrachtet) bedeutet die Anwendung der oben genannten Kriterien, dass nicht automatisch die kürzeste Distanz der beste Kamerapfad ist. Es ist offensichtlich, dass die kürzeste euklidische Verbindung zweier Sichten oft direkt durch die Szene führt und alles andere als erwünscht ist. Aber selbst die

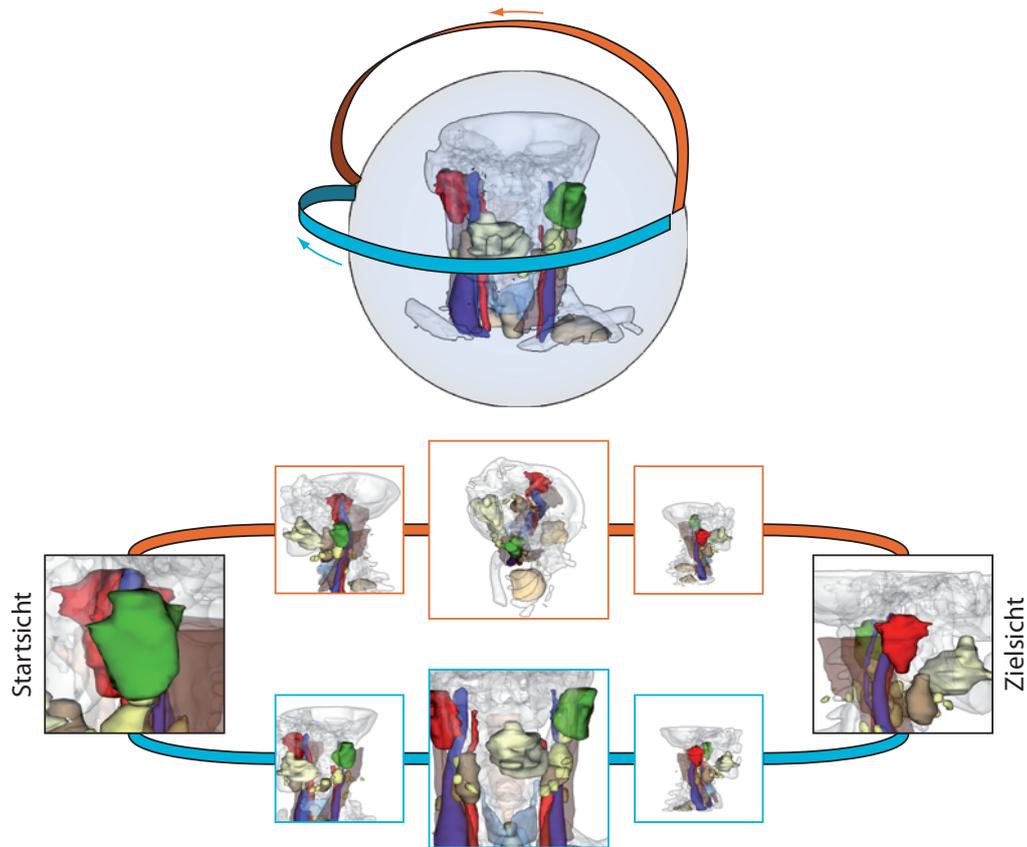


Abbildung 6.1: Vergleich des Kamerafluges mit kürzester geodätischer Distanz (oranger Pfad) mit einem Kameraflug durch gewohnte Regionen (blauer Pfad) entlang des Äquators der Szene. Zu erkennen ist, dass beim kürzeren Flug ungewöhnliche Sichten (fast senkrecht von oben) eingenommen werden, während dies beim etwas längeren Flug nicht geschieht.

kürzeste geodätische Distanz zwischen zwei Sichten auf der Oberfläche der szenenumgebenden Kugel ist nicht immer optimal. Das in Abb. 6.1 dargestellte Beispiel zeigt den kürzesten geodätischen Kamerapfad von einer Sicht auf die linke Ohrspeicheldrüse zu einer Sicht auf die rechte Ohrspeicheldrüse. Der Kamerapfad führt knapp über den Pol der umgebenden Kugel und erzeugt so Sichten auf die Szene, die für einen HNO-Chirurgen äußerst ungewohnt sind (siehe dazu auch die Betrachtungen zur Vorzugsblickrichtung in Abschnitt 4.4.2). Ein etwas längerer Kamerapfad führt dagegen entlang des Äquators und kann so die gesamte Kamerafahrt über vertraute Sichten präsentieren. In Abschnitt 6.3 wird daher ein Verfahren präsentiert, welches angemessene Kamerapfade unter Berücksichtigung einer ganzen Reihe von Parametern automatisch berechnet.

Neben der Berechnung einzelner Kamerapfade zwischen zwei Sichten ist die Betrachtung von komplexeren Pfaden im Kontext medizinischer Visualisierungen lohnenswert. Solche komplexeren Pfade können durch das „Abfahren“ einer Vielzahl von guten Sichtpunkten entstehen. Als Beispiel seien hier die Betrachtung aller vergrößerten Lymphknoten in einer Darstellung des Halses oder die Inspekti-

on aller Tumore im Kontext naher Gefäßäste in der Leber genannt. Unter Berücksichtigung der Angemessenheit eines Kamerapfades sollten solche Sichten nicht in beliebiger Reihenfolge abgefahren werden. Auch stellt sich die Frage, ob solche komplexeren Pfade nicht vereinfacht und damit ihre Gesamtlänge verkürzt werden kann. Diesem Problem widmete sich Abschnitt 5.1 mit dem Clustering von Kamerapositionen.

6.2 Verwandte Arbeiten

Ein großer Teil der Arbeiten zur Planung von Kamerafahrten ist im Bereich der Exploration von komplexen 3D-Welten in Spielen (z.B. HALPER u. a. [2001]) sowie Systemen aus vielen durch Tore oder Türen verbundenen Räumen angesiedelt (z.B. COHEN-OR u. a. [2003]). CHRISTIE u. a. [2008] geben dazu einen guten Überblick. Solche Verfahren fokussieren z.B. meist auf eine Kollisionsvermeidung, die bei den im Rahmen dieser Arbeit genutzten kompakten 3D-Modellen keine Rolle spielen. Im Kontext der vorliegenden Arbeit sind vor allem verwandte Ansätze aus dem Bereich der Kamerapfadplanung auf szenenumgebenden Kugeln sowie auf der Basis von Sichtpunktbewertungen interessant.

BARRAL u. a. [1999] bzw. BARRAL u. a. [2000] stellten ein Verfahren zur Ermittlung von Pfaden zur freien Szenenexploration vor. Dabei soll eine Szene von einer umgebenden Kugel aus möglichst umfassend von einer Kamera automatisch abgefahren werden. Als Maß für die Güte eines Sichtpunktes wird von möglichst vielen sichtbaren Dreiecken und einer großen projizierten Fläche ausgegangen. Ausgehend vom unter diesen Bedingungen besten Sichtpunkt (siehe Beschreibung dieses Verfahrens in Abschnitt 4.3.1) wird ein Pfad stückweise entlang möglichst guter Sichtpunkte gesucht. Ausgehend vom letzten Teilstück werden drei Sichtpunkte von der bisherigen Richtung weg zeigend untersucht. Der Sichtpunkt mit der besten Güte ist der nächste Anlaufpunkt der Kamera. Zusätzlich wird versucht, möglichst nicht zum Startpunkt zurückzukehren, um noch unbetrachtete Teile der Szene erreichen zu können. Das Verfahren eignet sich für die ziellose und unvoreingenommene Exploration neuer Szenen und ist nicht für Pfade ausgelegt, die einen festen Start- und Zielpunkt haben.

Ein ähnliches Verfahren wurde von SOKOLOV u. a. [2006a] präsentiert. Auch hier wird die Szene automatisch vom besten Sichtpunkt aus ziellos durch eine Kamerafahrt exploriert. Dabei wird die Kamera von anderen guten Sichtpunkten sowie großen noch „unbesuchten“ Flächen angezogen.

Für medizinische Volumendarstellungen haben VÁZQUEZ u. a. [2008] ein Verfahren vorgestellt, welches gute Sichtpunkte²³ durch einen Pfad verbindet. Das Verfahren nutzt jedoch keinerlei Zusatzinformationen, so dass selbst der Up-Vektor der Kamera nicht an die vorhandenen Verhältnisse angepasst wird und so sehr

²³Die Berechnung guter Sichtpunkte nach dem Verfahren von VÁZQUEZ u. a. [2008] ist in Abschnitt 4.3.2 beschrieben.

ungewöhnliche, weil gekippte oder auf dem Kopf stehende Kamerafahrten durchgeführt werden.

Für den speziellen Fall der automatisierten Videoerstellung zur Analyse intrakranialer Aneurysmen haben [ISERHARDT-BAUER u. a. \[2002\]](#) auf der Basis von 50 Patientendatensätzen die wichtigsten, anzufahrenden Punkte identifiziert. Durch die Drehung des dargestellten Volumens um deren Zentrum wird die Kamera auf einem Pfad um die Darstellung bewegt. Durch umfassende Rotationen, so z.B. eine 360°-Rotation zum Schluss eines jeden Videos, soll sichergestellt werden, dass bei einer Kamerafahrt alle wesentlichen Punkte abgefahren werden.

Der automatischen Steuerung von Kameras durch eine Szene aus mehreren Objekten haben sich sowohl [BECKHAUS u. a. \[2001\]](#) als auch [SOKOLOV & PLEMENOS \[2008\]](#) angenommen. [BECKHAUS u. a. \[2001\]](#) gingen von einer Szene aus, in der die einzelnen Objekte durch sie umgebende Boxen approximiert werden. Die Größe der Boxen bildet die momentane Wichtigkeit der Objekte wieder. In der Szene wird ein Potentialfeld berechnet, welches auf den Größen der Boxen basiert: Große Boxen wirken auf die Kamera anziehend, kleine (weil unwichtige) abstoßend. Das Modell erlaubt die Adaption auf eine Veränderung der Nutzerinteressen, indem die Boxen in ihrer Größe dynamisch angepasst werden können. Dieses Verfahren kam unter anderem beim 3D-Puzzle von [RITTER u. a. \[2001\]](#) zur Anwendung, bei dem die Wichtigkeit der einzelnen Objekte aus einem *Knowledge Server* bezogen wurden und zwischen verschiedenen Objekten in einer Hierarchie vererbt werden konnten.

[SOKOLOV & PLEMENOS \[2008\]](#) zeigten während einer Kamerafahrt durch eine Szene wichtige Objekte so lange, bis alle wesentlichen Informationen gezeigt wurden und schwenkten dann mit der Kamera zum nächsten wichtigen Objekt. Da es sich bei den von [SOKOLOV & PLEMENOS \[2008\]](#) verwendeten Modellen um Städteansichten handelte, war die Höhe der Kamera auf die eines virtuellen Besuchers festgelegt und musste kollisionsfrei zwischen den verschiedenen Gebäuden bewegt werden.

Neben der Exploration von 3D-Darstellungen von außen bzw. in einer Art „walk-through“-Verfahren spielen Pfadplanungen für die Navigation in geschlossenen, meist röhrenförmigen Strukturen auch in der Medizin (z.B. bei der virtuellen Endoskopie) eine wesentliche Rolle. [CHAUDHURI u. a. \[2004\]](#) hatten eine Arbeit vorgestellt, in der sie für solche geschlossenen Strukturen einen Pfad vorberechnen, der möglichst weit von den jeweiligen Wänden entfernt in der Mitte der Struktur liegt. Dabei legen sie besonderen Wert auf eine effiziente Berechnung mit einem multi-resolution Ansatz, bei dem der Nutzer entscheiden kann, wie genau, und damit wie lange, die Pfadberechnung erfolgen soll.

Für die Navigation in 2D-Darstellungen haben [VAN WIJK & NUIJ \[2004\]](#) den Zusammenhang von Kamerabewegung und -zoom mithilfe von Nutzerstudien aufwendig untersucht. Bei der Bewegung einer Kamera von einem Start- zu einem

Zielpunkt (z.B. in einer Kartendarstellung) kam eine Metrik zum gleichzeitigen Bewegen (Panning) und Zoomen der Kamera zum Einsatz. In der Nutzerstudie konnten die Probanden die Geschwindigkeit der Kamerabewegung sowie den Umfang des Heraus- und Hineinzoomens selbst regeln. Als Ergebnis stand eine große Breite an verschiedenen Nutzerwünschen, die in ihrer Betrachtung einen (wenn auch einfach gehaltenen) Einfluss des Nutzers auf die Kamerabewegungen nahe legt.

Für kaum eines der genannten Verfahren ist eine Nutzerstudie bekannt, die zeigt, dass die resultierenden Pfade wirklich als angenehm und informationsreich wahrgenommen werden. Dies ist umso bedauerlicher, als dass die grafischen Repräsentationen einiger Pfade auf den szenenumgebenden Kugeln ein eher chaotisches Richtungsverhalten an den Tag legen. Das mag in der ziellosen lokalen Suche nach den jeweils nächsten Kamerapositionen begründet sein.

6.3 Planung von Kamerapfaden zwischen zwei Sichten

Wählt der Nutzer eine Struktur aus (über eine Liste oder direkt in der Szene), so wird die Kamera automatisch zu einem guten Sichtpunkt auf diese Struktur bewegt. Der dabei zurückgelegte Pfad wird im Voraus nach bestimmten Kriterien berechnet (siehe Seite 106).

6.3.1 Informationsmaximierung

Um die während einer Kamerafahrt präsentierte Information zu erhöhen, sollte die Zielstruktur möglichst gut sichtbar sein. Beginnt der Pfad an einem guten Sichtpunkt auf eine andere Struktur (die Startstruktur), so sollte zu Beginn auch deren Sichtbarkeit gewährleistet sein. Zur Vermeidung ungewohnter Sichten kann sich der Pfad an der Vorzugsblickrichtung orientieren. Dazu kommen die Parameterfelder zur Bestimmung guter Sichtpunkte aus Kapitel 4 zur Anwendung. Die Parameterfelder der Vorzugsblickrichtung und der Sichtbarkeit werden summiert, um im Ergebnisfeld möglichst gute Kamerapositionen zu finden, die sowohl eine gewohnte Sicht wie auch eine gewisse Sichtbarkeit der Zielstruktur garantieren.²⁴ Ausgehend vom aktuellen Kamerastandpunkt wird im Parameterfeld in Richtung der Zielstruktur in einem Öffnungswinkel von 90° die nächstbeste Position gesucht (Abb. 6.2). Der Öffnungswinkel erlaubt sowohl eine gewisse Variabilität des Pfades als auch eine zielgerichtete Bewegung. Der so ermittelte Pfad ist nicht der kürzeste, doch präsentiert er schon früh die Zielstruktur und

²⁴Eine größere Wichtung der Vorzugsblickrichtung sorgt dafür, dass Positionen mit einer sehr guten Sichtbarkeit der Zielstruktur in einem unerwünschten Gebiet (niedrige Werte der Vorzugsblickrichtung) dennoch nicht angefahren werden.

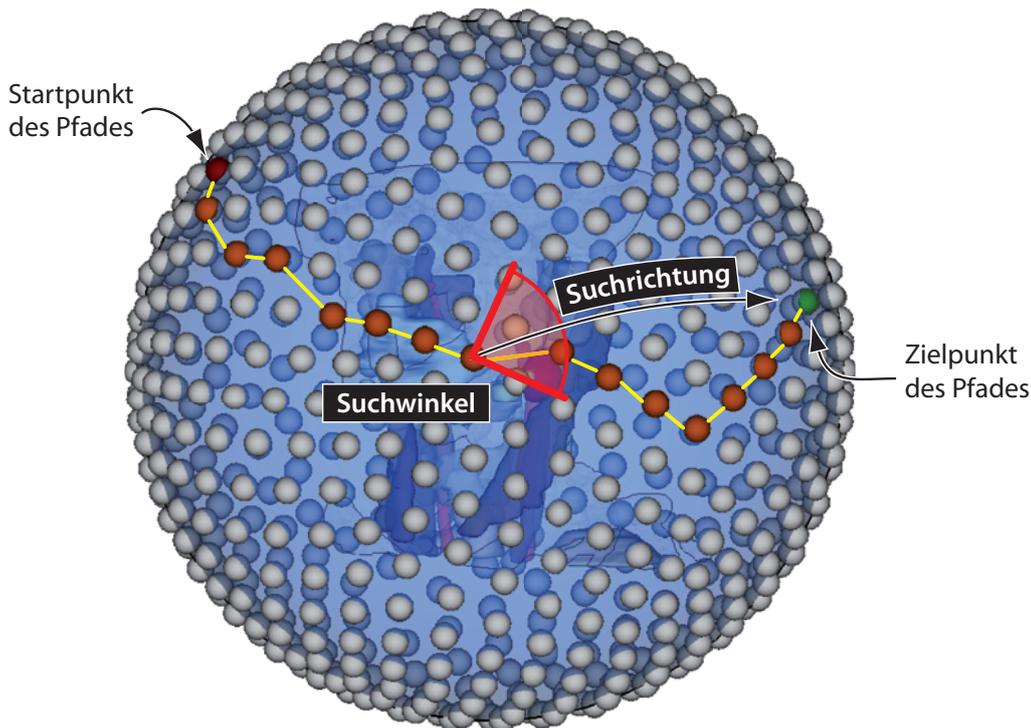


Abbildung 6.2: Ermittlung eines optimalen Kamerapfades anhand der Sichtbarkeit der Zielstruktur sowie der Vorzugsblickrichtung. Von der aktuellen Position aus wird jeweils in Richtung des Zielpunktes in einem Öffnungswinkel von 90° nach der besten nächsten Kameraposition gesucht.

verbessert somit die Orientierung wie auch die Informationsdichte während der Kamerafahrt. Durch eine Glättung des entstandenen Kamerapfades mittels Bezier-Kurven (BUCKLEY [1994]) wird erreicht, dass die Kamera an den verschiedenen Sichtpunkten nicht ruckartig ihre Richtung ändert, sondern dass eine weiche, angenehme Bewegung erfolgt.

6.3.2 Kameraausrichtung

Während einer Kamerafahrt kann die Kamera unabhängig von ihrer Position durch ihre Ausrichtung auf eine bestimmte Struktur oder in eine Richtung zusätzlich gesteuert werden. Eine reine Fixierung auf den Szenenmittelpunkt kann ein guter Ansatz sein. Jedoch bieten die zu Grunde liegenden Sichtbarkeitsinformationen die Möglichkeit, die Kamera gezielt auf die Start- oder Zielstruktur hin auszurichten. So wird die Kamera zu Beginn der Kamerafahrt noch auf die Startstruktur ausgerichtet.²⁵ Durch das Parameterfeld der Sichtbarkeit der Zielstruktur ist deren Sichtbarkeit die gesamte Kamerafahrt über bekannt. Erreicht die Sichtbarkeit der Zielstruktur einen Mindestwert von 70%, so schwenkt die Kamera

²⁵Ist keine Startstruktur bekannt, weil die Kamerafahrt an einem individuell durch den Betrachter gesetzten Kamerastandpunkt startet, wird die Kamera auf das Zentrum der Szene ausgerichtet.

zur Zielstruktur.²⁶ Eigene Untersuchungen anhand zahlreicher Test-Szenen haben gezeigt, dass ein Zurückschwenken auf die Startstruktur, wenn die Zielstruktur später einmal verdeckt sein sollte, als sehr unangenehm empfunden wird. Daher bleibt die Kamera vom ersten Schwenk an bis zum Ende der Kamerafahrt auf die Zielstruktur ausgerichtet.

6.3.3 Zoomverhalten

Zusätzlich zur reinen Strecke des Pfades und der Orientierung der Kamera ist ihr Zoom von Bedeutung. Bewegt sich die Kamera von einem Close-Up²⁷ zu einem anderen Close-Up, so ist offensichtlich, dass diese große Zoomstufe nicht die gesamte Kamerafahrt über beibehalten werden kann. Die Kamera zoomt daher zu Beginn der Kamerafahrt von der Szene weg, um zum Ende hin auf die Zielstruktur hin zu zoomen (Abb. 6.3). Die Stärke des Zooms wird dabei von mehreren Faktoren beeinflusst und begrenzt. Es gibt einen Mindestzoom, unter den nicht aus der Szene heraus gezoomt wird. Dieser Wert ist erreicht, wenn die gesamte Szene komplett sichtbar ist. Bei langen Kamerafahrten wird auf diese Stufe heraus gezoomt und der Zoom dann bis kurz vor dem Ende der Kamerafahrt nicht verändert, wo auf die Zielstruktur gezoomt wird. Da ein vollständiges Herauszoomen aus der Szene bei nur kurzen Kamerafahrten unerwünscht ist, bemisst sich die Stärke des Zooms an der Länge der Kamerafahrt (Abb. 6.4).

6.3.4 Geschwindigkeit der Kamerabewegung

Eine Kamerabewegung sollte auch bezüglich ihrer Geschwindigkeit ein physikalisch angemessenes und damit flüssiges Verhalten zeigen. Um abrupte Geschwindigkeitsänderungen zu vermeiden, wird die Kamera zu Beginn einer Kamerafahrt beschleunigt und zum Ende hin abgebremst. Die Geschwindigkeit korreliert dabei mit dem Zoomverhalten der Kamera – bei großen Zoomstufen bewegt sich die Kamera langsamer, bei größerer Entfernung zur Szene schneller. Ebenso wie beim Zoomen gibt es eine Maximalgeschwindigkeit, die erreicht wird, wenn die Kamera am weitesten aus der Szene herausgezoomt hat.

²⁶Dies geschieht erst nach einer Mindestzeit von 1 Sekunde nach dem Start, um ein sofortiges Wegschwenken der Kamera zu verhindern.

²⁷Die Kamera ist sehr nah an eine Struktur heran gezoomt.

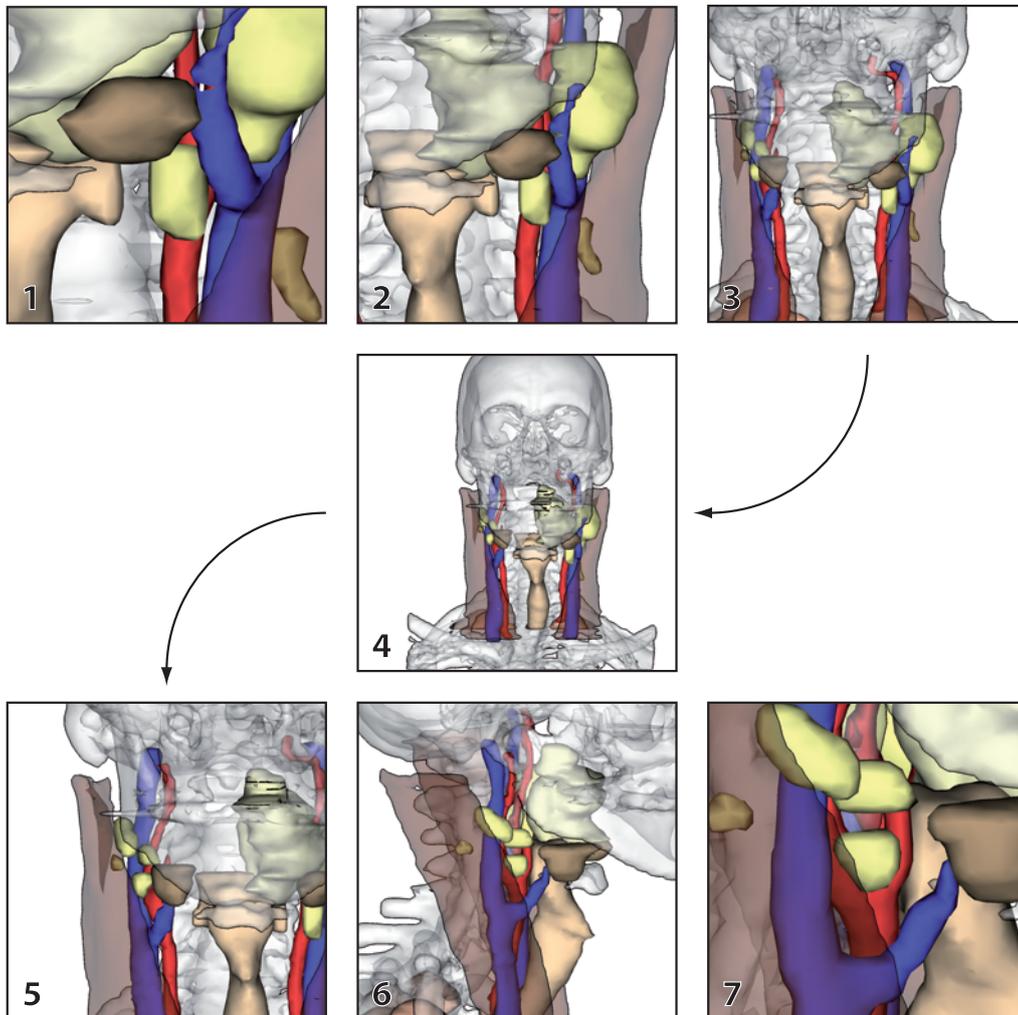


Abbildung 6.3: Bei langen Kamerapfaden wird aus der Szene heraus gezoomt, um für den Betrachter die Orientierung zu erhalten. Hier ist eine Kamerafahrt von einer Sicht auf einen Lymphknoten im rechten Halsbereich auf einen Lymphknoten im linken Halsbereich dargestellt. Zu Anfang und Beginn wird jeweils stark an die jeweilige Struktur heran gezoomt.

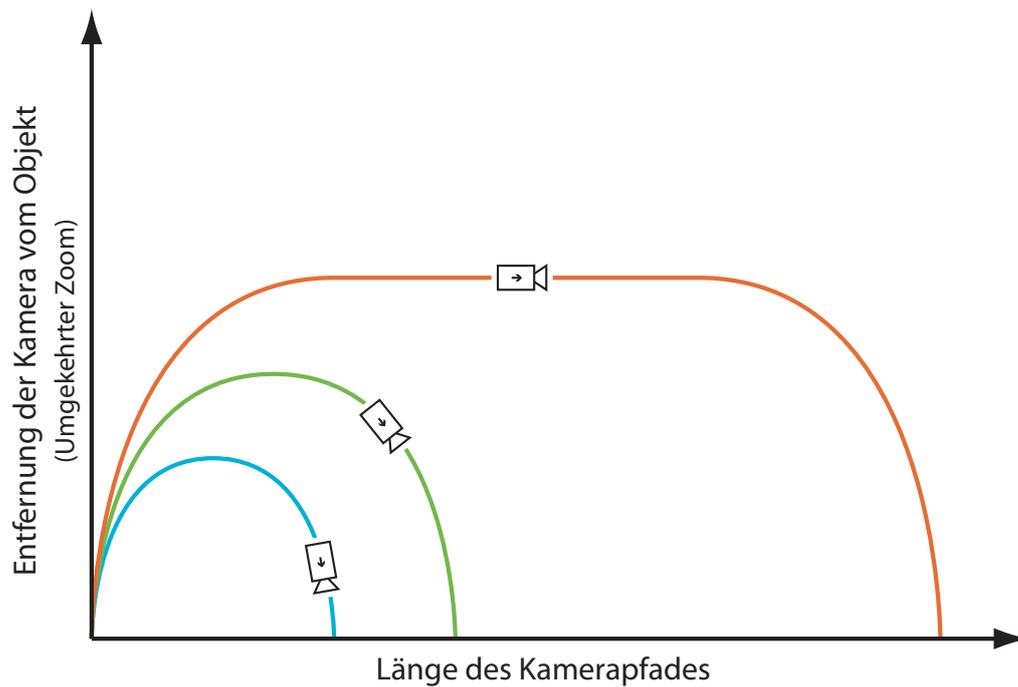


Abbildung 6.4: Die Stärke der Herauszoomens der Kamera aus der Szene hängt von der Länge des Kamerapfades ab. Bei kurzen Kamerapfaden wird nicht vollständig heraus gezoomt. Es gibt ein Maximum, auf das heraus gezoomt wird, wenn der Kamerapfad besonders lang ist. Das Maximum entspricht einer vollständigen Sicht auf die gesamte Szene.

6.4 Zusammenfassung

Es wurden zwei Verfahren zur Optimierung von Kamerapfaden in kompakten medizinischen Szenen vorgestellt. Dabei wurde auf einen möglichst hohen Informationsgehalt während einer Kamerafahrt sowie auf die Beibehaltung der Orientierung des Betrachters Wert gelegt.

Bewusst außen vor gelassen wurde das Feld der intra-strukturellen Kamerapfade, wie sie bspw. zur Endoskopieplanung genutzt werden. Solche Pfade erfordern einen völlig anderen Berechnungsansatz und stellen andere Anforderungen an neue Interaktionsformen. So spielt neben der deutlich größeren Länge der Pfade die Manipulation der Kamera *während* einer Kamerafahrt eine größere Rolle. Ein Pausieren der Kamerafahrt sowie ein Schwenken der Kamera zu den Seiten bei fortlaufender Vorwärtsbewegung sind nur zwei Aspekte, die es zu bedenken gilt.

Eine Evaluierung der Kamerapfade fand bisher nicht statt. Dies ist im Zusammenhang mit der Evaluierung des gesamten Animationsansatzes zu sehen, die durchaus kontrovers diskutiert werden kann (siehe Ausblick in Abschnitt [11.1](#)). Das bisherige Feedback zu den Pfaden als Teil komplexerer Animationen war jedoch ausnahmslos positiv.

7

Automatische Annotation medizinischer Darstellungen

Textuelle Annotationen sind in vielen Fällen eine zusätzliche Informationsquelle in Visualisierungen. Dem planenden Chirurgen können so zusätzliche Daten, wie Volumina, und dem Nutzer eines Lernsystems Strukturbezeichnungen direkt in der Visualisierung präsentiert werden. Dazu müssen die Annotationen automatisch in der interaktiven Darstellung platziert werden. Im Rahmen dieser Arbeit wurde ein Framework geschaffen, welches die automatische Annotation sowohl von medizinischen 3D-Darstellungen als auch 2D-Schichtbildern ermöglicht. Es wurde entwickelt, um in chirurgischen Trainingssystemen und Interventionsplanungsassistenten zum Einsatz zu kommen. Des Weiteren wurden neue Interaktionstechniken auf Basis von Annotationen entwickelt sowie eine Untersuchung hinsichtlich der genauen Darstellungsart und des Informationsinhaltes von Annotationen vorgenommen.

Gliederung des Kapitels

Nach einer kurzen Motivation (Abschnitt 7.1) werden zunächst grundlegende Eigenschaften, Anforderungen und Anwendungsfelder von Annotationen in medizinischen Visualisierungen eingeführt (Abschnitt 7.2). Anschließend werden verwandte Arbeiten aus der 2D- und 3D-Annotation sowie aus dem medizinischen Bereich präsentiert (Abschnitt 7.3). In den beiden folgenden Abschnitten werden zwei neue Verfahren zur Annotation von 2D-Schichtbildern (7.4) und 3D-Szenen (7.5) vorgestellt. Abschließend werden eine Evaluierung präsentiert (Abschnitt 7.6) und ein Ausblick auf zukünftige Entwicklungen gegeben (Abschnitt 7.7).

Eigene Publikationen

Dieses Kapitel basiert u.a. auf folgenden eigenen Publikationen:

MÜHLER & PREIM [2009a] Mühler, K. & Preim, B. *Automatic Textual Annotation for Surgical Planning*. In: **Proc. of Vision, Modeling, and Visualization (VMV)**, 2009

MÜHLER & PREIM [2009b] Mühler, K. & Preim, B. *Automatische Annotation medizinischer 2D- und 3D-Visualisierungen* **Proc. of Bildverarbeitung für die Medizin (BVM)**, 2009

7.1 Motivation

Textuelle Annotationen bereichern eine Visualisierung und bieten zusätzliche Informationen, z.B. Strukturbezeichnungen oder Abmessungen. Gerade in interaktiven Darstellungen ist eine automatische Platzierung von Annotationen an *guten* Positionen unabdingbar. Gute Positionen unterstützen die Lesbarkeit der Annotation, indem diese bspw. nicht verdeckt werden und auch selbst keine wesentlichen Informationen verdecken. In medizinischen Systemen besteht schon lange der Bedarf an adäquaten Annotationen, sowohl von 2D-Schichtbilder als auch von 3D-Darstellungen. Dennoch bieten die meisten Systeme keine oder nur sehr beschränkte automatische Annotationsmöglichkeiten. So hat die Firma *MeVis Medical Solutions* (**MEVIS MEDICAL SOLUTIONS - DISTANT SERVICES** [2010]) bisher über 4000 Fälle für die Planung von leberchirurgischen Eingriffen analysiert und die Ergebnisse grafisch aufbereitet (meist als Microsoft-Powerpoint-Präsentation oder Acrobat-PDF). In den meisten Fällen waren jeweils ca. 30 bis 40 Visualisierungen manuell zu annotieren. Es ist daher evident, welche Vorteile auch in zeitlicher Hinsicht ein System zur automatischen Annotation medizinischer Visualisierungen bringen würde.

Die Annotation sowohl von 2D- als auch 3D-Visualisierungen wurde in der Vergangenheit um eine Reihe von Verfahren bereichert, die eine automatische Annotation ermöglichen. Bei der 2D-Annotation ist das Kernanwendungsfeld in der Kartographie zu finden. Hier besteht das Ziel meist darin, möglichst viele einzelne Punkte und Flächen (z.B. Städte, Wälder oder Flüsse) überlappungsfrei zu beschriften. In der automatischen Annotation dreidimensionaler Darstellungen liegt der Fokus auf der Annotation einzelner Objekte einer Szene. Hier kann zwischen kompakten Szenen, die von außen betrachtet werden und Szenen, bei denen sich der Betrachter innerhalb der Darstellung befindet, unterschieden werden. Die Herausforderung in beiden Fällen liegt in der Identifizierung von freiem Raum zur Platzierung der Annotationen und deren überlappungsfreier Darstellung. In interaktiven Umgebungen muss dies in Echtzeit geschehen. Die spezielle Herausforderung bei medizinischen Darstellungen liegt in ihren speziellen Eigenschaften:

- Die im Rahmen dieser Arbeit genutzten 3D-Visualisierungen bestehen oft aus semi-transparenten Strukturen, die sich gegenseitig mehrfach verdecken. Aktuelle Verfahren können jedoch nur unverdeckte Objekte annotieren.
- Bei der Annotation von 2D-Schichtbildern stellt die Vermeidung des “Springens” der Annotationen beim Durchblättern der einzelnen Schichten eine besondere Herausforderung dar, um die Lesbarkeit und die Verfolgung einzelner Annotationen über mehrere Strukturen hinweg zu ermöglichen.
- Neben der rein passiven Rolle, Objekte zu beschriften, können gerade in medizinischen Visualisierungen Annotationen eine aktive Rolle als Mittel der Interaktion und Explorationsunterstützung sein. Durch Interaktion mit den Annotationen (bspw. durch Anklicken) können Veränderungen an der Visualisierung herbeigeführt werden – z.B. die Einblendung zusätzlicher Informationen oder die Hervorhebung nicht sichtbarer Strukturen. Dadurch kann der Explorationsprozess unterstützt und beschleunigt werden.

7.2 Anwendungsfelder und Anforderungsanalyse

Annotationen, gleich in welcher Anwendungsdomäne, bezeichnen meist implizit textuelle Annotationen. Auch in dieser Arbeit wird, so nicht explizit darauf hingewiesen wird, bei Annotationen von einem (zumeist kurzen) Text ausgegangen, der über eine Verbindungslinie mit der zu annotierenden Struktur verbunden ist. Der Endpunkt der Verbindungslinie auf oder in der Struktur wird als Ankerpunkt bezeichnet (Abb. 7.1). Interne Annotationen, wie sie z.B. von [GÖTZELMANN u. a. \[2005\]](#) und [ROPINSKI u. a. \[2007\]](#) präferiert werden, wurden bei ausführlichen Diskussionen mit den Anwendern und medizinischen Partnern generell als ungeeignet verworfen²⁸ und werden daher in dieser Arbeit nicht weiter betrachtet.

7.2.1 Anwendungsfelder von Annotationen in medizinischen Visualisierungen

Im Bereich der medizinischen Visualisierungen können Annotationen sowohl in 2D-Schichtbildern als auch in 3D-Visualisierungen zum Einsatz kommen. Im Kontext der in dieser Arbeit fokussierten Anwendungsfelder der Interventionsplanung und der medizinischen Ausbildung lassen sich drei maßgebliche Anwendungsfelder erkennen:

1. Benennung und Kenntlichmachung von Strukturen

Bei dieser ursprünglichsten Aufgabe von Annotationen werden einzelne

²⁸Interne Annotationen sind meist aufgrund der Verformung des Textes und des inhomogenen Hintergrundes nur sehr schwer lesbar.

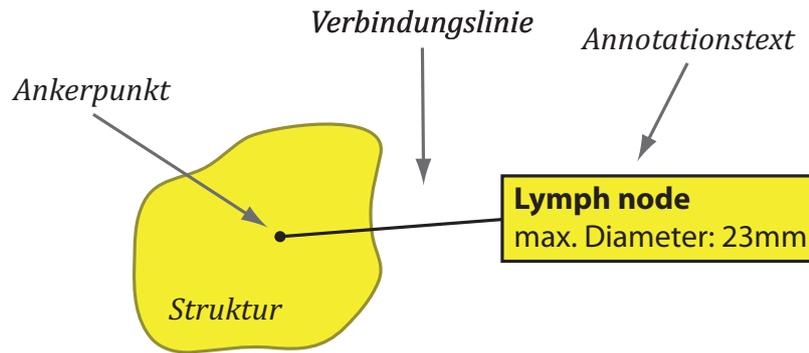


Abbildung 7.1: Schematische Darstellung der Annotation einer Struktur. Es sind die verschiedenen Teile, wie Ankerpunkt, Verbindungsline und Annotationstext, erkennbar.

Strukturen mit ihrer anatomischen Bezeichnung oder Zugehörigkeit annotiert. Das erleichtert ihre Identifizierung in einer Darstellung mit mehreren Strukturen. Diese Funktion ist vor allem für Lernsysteme von großer Bedeutung, da hier meist noch keine Vertrautheit mit der Darstellung von anatomischen Strukturen in 2D-Schichtbildern oder künstlichen 3D-Szenen vorhanden ist.

2. Vermittlung von Zusatzinformationen

Die Darstellung von Zusatzinformationen zu einzelnen Strukturen direkt in der Visualisierung bietet den Vorteil, dass der Betrachter diese Informationen direkt im Fokus präsentiert bekommt und sie nicht aufwendig zusätzlichen Texten oder Listen entnehmen muss. Die präsentierten Zusatzinformationen können vielfältiger Natur sein:

So können wichtige Maßzahlen, wie Volumina oder Ausdehnungen von Strukturen, in der Annotation gezeigt werden. In einem Lernsystem können einzelne Kommentare und Hinweise, die durch einen Experten hinterlegt wurden, direkt an den betreffenden Stellen der Visualisierung angezeigt werden. Auch im klinischen Alltag können über Annotationen Kurzbefunde und andere Kommentare zwischen den planenden und diagnostizierenden Ärzten ausgetauscht werden.

3. Einsatz von Annotationen als Explorationshilfe

Ein in der Forschung bisher wenig beachtetes Thema ist die Anwendung von Annotationen als Explorationsunterstützung. Dabei können im einfachsten Fall die Annotationstexte als zusätzliche Selektierungsfläche genutzt werden, was besonders bei sehr kleinen Strukturen hilfreich ist. Weiterhin kann mithilfe von Annotationen auf wichtige aber momentan nicht sichtbare Strukturen hingewiesen werden. Eine Selektion einer solchen Annotation kann wiederum zu einer interaktiven Anpassung der Visualisierung zur Sichtbarmachung der entsprechenden Struktur führen – beispielsweise in einer 3D-Darstellung durch einen Kameraflug zu einem guten Sichtpunkt auf die Struktur.

7.2.2 Grundlegende Anforderungen an Annotationen und stilistische Aspekte

Generell sind gewisse ästhetische Anforderungen an Annotationen zu stellen, um deren Lesbarkeit und die Zuordnung zu den entsprechenden Strukturen nicht zu erschweren. [IMHOF \[1975\]](#) und [GÖTZELMANN u. a. \[2005\]](#) haben entsprechende Untersuchungen vorgestellt. Weiterhin kann man sich an der Nutzung von Annotationen in Anatomieatlanten wie [VOLL & WESKER \[2009\]](#) orientieren. Dabei ist zu berücksichtigen, dass aufgrund der statischen Darstellungen sowie der hohen Produktionskosten pro Abbildung möglichst viele Annotationen in einer Darstellung untergebracht werden müssen. Bezogen auf das hier betrachtete Anwendungsfeld medizinischer Visualisierungen zur Operationsplanung lassen sich folgende Grundeigenschaften für Annotationen erkennen:

1. **Verbindungslinien** sollten sich nicht überschneiden.
2. Annotationen sollten **überlappungsfrei** platziert werden. Dabei sollten sowohl die Annotationstexte keine Strukturen verdecken, als auch die Annotationstexte untereinander nicht überlappen.
3. Es sollte ein **angemessener Abstand** zur jeweils annotierten Struktur gegeben sein. Der Annotationstext sollte möglichst nah, aber dennoch mit einem gewissen Mindestabstand zur Struktur platziert werden.
4. Es sollten nicht zu viele Annotationen in einer Darstellung platziert werden, um zum einen den Betrachter nicht zu verwirren und um zum anderen nicht zu viel Platz in der Visualisierung einzunehmen. Eine Möglichkeit, diesem zu begegnen, ist die **Gruppierung von Annotationen** der gleichen Struktur oder Strukturgruppe. Dabei werden ausgehend von einer Annotation mehrere Strukturen über Verbindungslinien annotiert.
5. Um eine übermäßige Verdeckung von sehr kleinen Strukturen durch Ankerpunkte und Annotationslinien zu vermeiden, sollten solche Strukturen beispielsweise durch einen **umgebenden Kreis anstelle eines Ankerpunktes** mit der Annotationslinie verbunden werden.
6. Wenn möglich, sollten **Annotationen aneinander ausgerichtet** sein, um ein klarer strukturiertes Gesamtbild zu erzeugen.

Bei der technischen Umsetzung der Annotationen muss grundsätzlich zwischen den Verfahren zur 2D- und 3D-Annotation unterschieden werden. Dennoch gibt es einige gemeinsame Eigenschaften. So sollten die Annotationen **vollständig automatisch** in die Visualisierungen integriert werden. Die Annotationen sollten sich weiterhin direkt an die Veränderungen der Visualisierung anpassen. Das sollte in Echtzeit geschehen. Bei 2D-Schichtbildern ist eine solche Veränderung beim Vorgang des Scrollens zwischen verschiedenen Schichtbildern gegeben. Es ist zu erwarten, dass eine stetige Neuberechnung aller Annotationspositionen zu einem

“Springen” der Annotationstexte führt, welches das Lesen und die Verfolgung einer annotierten Struktur über mehrere Schichtbilder hinweg erheblich erschweren würde. Daher ist bei einem Verfahren zur 2D-Annotation dies besonders zu berücksichtigen. Des Weiteren kann es passieren, dass durch große Strukturen kaum freier Platz zur überlappungsfreien Platzierung der Annotationstexte bleibt. Dies kann sowohl in 2D- als auch in 3D-Darstellungen auftreten und ist entsprechend zu beachten.

7.2.3 Annotationstexte

Die Annotationstexte sollten frei gestaltbar sein. So können die Strukturen mit ihrer jeweiligen Strukturbezeichnung annotiert werden (Abb. 7.2). Um neben der reinen Identifizierung von Strukturen einen zusätzlichen Mehrwert zu liefern, können Annotationen um Kommentare erweitert werden. In einem Lernsystem sind dies beispielsweise Anmerkungen eines Experten zu besonders interessanten oder beachtenswerten Aspekten, die es beim Finden einer Therapieentscheidung zu berücksichtigen gilt. In Planungssystemen können dies eigene Anmerkungen des Arztes oder Anmerkungen von Fachkollegen, wie eines Radiologen, sein.

In der HNO-Therapieplanung sind die Ausdehnungen vergrößerter Lymphknoten von besonderem Interesse. In Schichtbilddaten ist eine solche Ausdehnung kaum direkt wahrnehmbar, da nur die Ausdehnung in der jeweiligen Schicht erkennbar und keine Informationen über andere Ausdehnungsrichtungen sichtbar sind. Daher kann die Anzeige der maximalen Ausdehnung in der Annotation dem planenden Chirurgen hier direkt auf vergrößerte Lymphknoten hinweisen und den Prozess der Inspektion kritischer Lymphknoten beschleunigen (Abb. 7.3).

In anderen Bereichen, wie der Leberchirurgie, spielen die Volumina von Strukturen eine wesentliche Rolle. Zur Planung einer Leberresektion sind für den Chirurgen die Volumina der einzelnen Couinaud-Segmente und durch die verschiedenen Gefäßbäume versorgten Segmente von besonderem Interesse. Eine Annotation der Lebersegmente mit ihren jeweiligen Volumina direkt im Schichtbild stellt daher eine Verbesserung gegenüber der mental aufwendigeren Übertragung von Volumenangaben in einer gesonderten Liste o.ä. auf die Segmentierungsergebnisse in der 2D-Darstellung dar (Abb. 7.4).

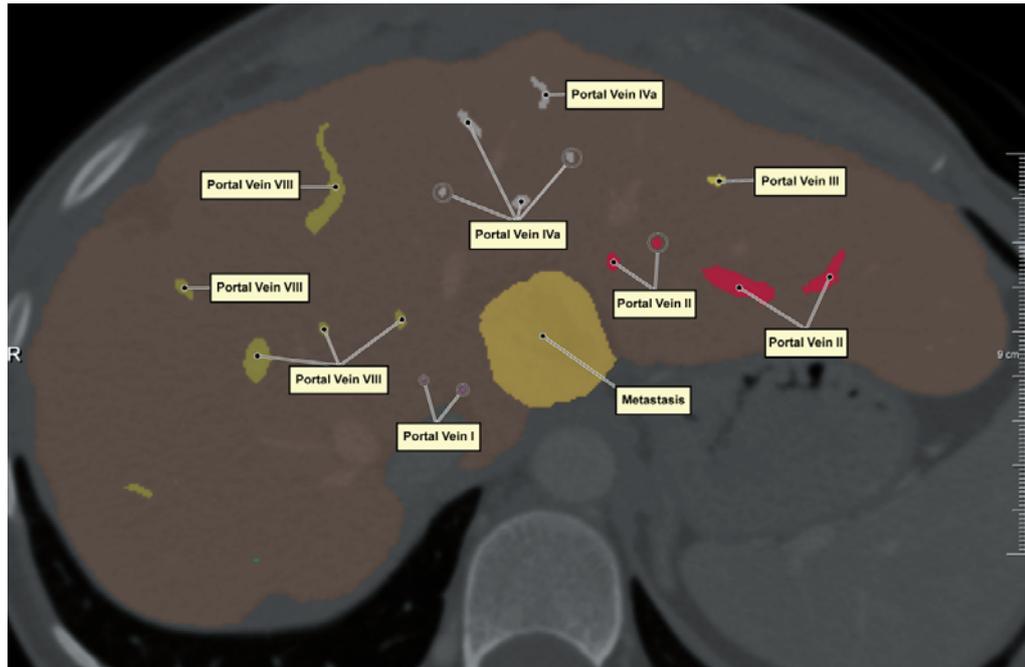


Abbildung 7.2: 2D-Annotation eines Schichtbildes der Leber. Annotation mehrerer Gefäße sowie des Tumors in einer Schichtbilddarstellung einer Leber.

7.3 Verwandte Arbeiten

Die Annotation von Visualisierungen ist schon lange Gegenstand der Forschung. So stellte schon METCALF [1920] Überlegungen an, auf welche Art und Weise biologische Illustrationen annotiert werden können. Neben der Annotation mit dem vollständigen Namen der Struktur schlug er vor, nur die Anfangsbuchstaben oder Zahlen zu nutzen, die dann in einer Legende am Bildrand referenziert würden. Der Einsatz von Legenden ist jedoch wenig brauchbar, da hier eine Teilung der Aufmerksamkeit des Betrachters provoziert wird (KHALIL u. a. [2005]).

Im Folgenden werden bereits existierende Verfahren zur automatischen 2D-Annotation (Abschnitt 7.3.1), zur automatischen 3D-Annotation (Abschnitt 7.3.2) sowie Systeme zur Annotation medizinischer Visualisierungen (Abschnitt 7.3.3) vorgestellt.

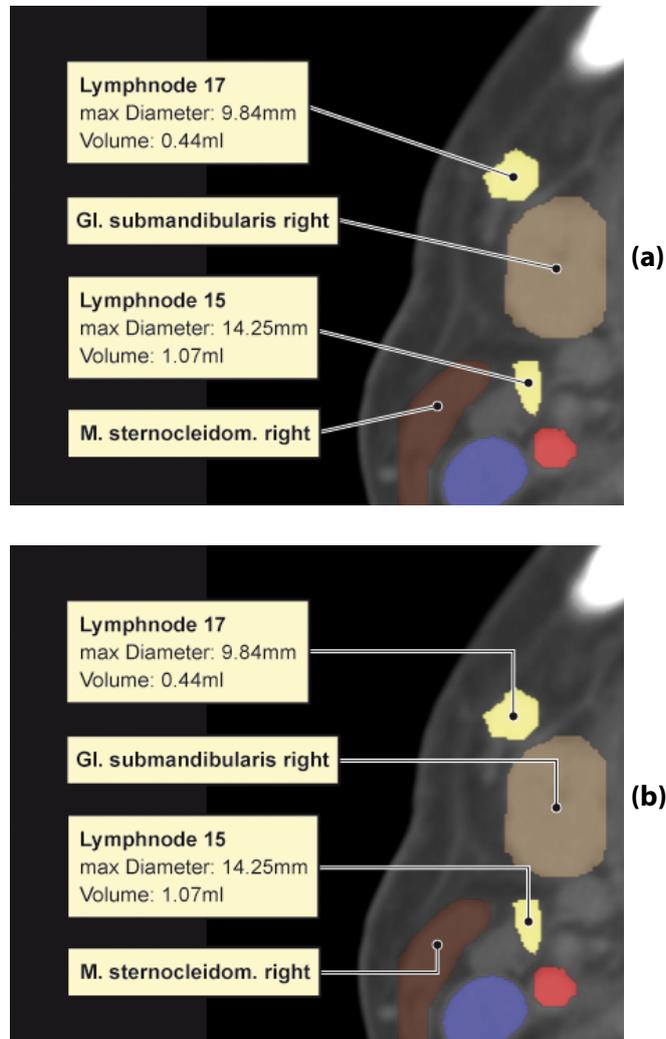


Abbildung 7.3: 2D-Annotation mehrerer Strukturen des Halses. Als Zusatzinformation werden die maximale Ausdehnung sowie das Volumen der Lymphknoten angezeigt. Es sind zwei mögliche Stile für die Verbindungslinien dargestellt – (a): direkte Linien und (b): orthogonale Linien.

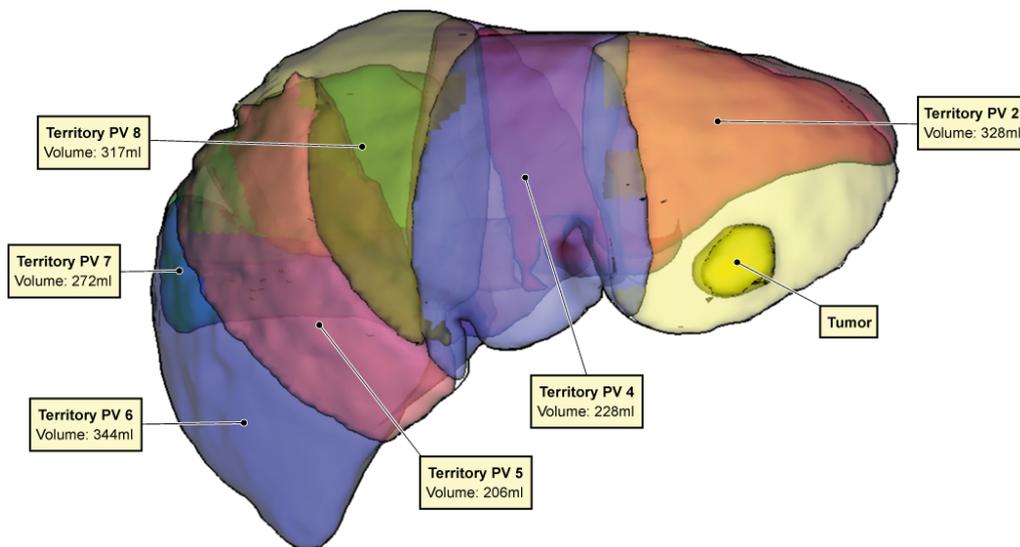


Abbildung 7.4: 3D-Annotation einer Leber. Es sind die Lebersegmente mit ihrem jeweiligen Volumen annotiert.

7.3.1 2D Annotationstechniken

Wesentliches Ziel bei der Annotation in zumeist statischen 2D-Visualisierungen ist die Platzierung von möglichst vielen Annotationen auf begrenztem Raum, wobei die zu Grunde liegende Visualisierung erkennbar bleiben muss, die Annotationen sich nicht gegenseitig verdecken und dennoch möglichst direkt an dem Ort platziert werden sollen, die sie annotieren. [CHRISTENSEN u. a. \[1995\]](#) haben gezeigt, dass dieses Problem der optimalen Platzierung von Annotationen NP-vollständig ist. Die Zahl möglicher Annotationsplatzierungen steigt exponentiell mit der Zahl der zu annotierenden Objekte. Da eine umfassende Untersuchung aller möglichen Platzierungen nicht in interaktiven Raten durchgeführt werden kann, werden zumeist Heuristiken angewandt und die Suche nach guten Positionen auf jeweils lokale Umgebungen einer Initialplatzierung beschränkt. [CHRISTENSEN u. a. \[1995\]](#) geben über die dabei eingesetzten Algorithmen und Verfahren einen guten Überblick.

In der Kartographie ist meist das sogenannte point-based-labeling, also die Annotation vieler Punkte mit Texten direkt an diesen, von Interesse. Die Annotation von Regionen über mit Annotationslinien verbundene Texte wurde dabei bisher wenig betrachtet. [BEKOS u. a. \[2006\]](#) haben einen Algorithmus vorgestellt, mit dem am Beispiel von Landkarten die jeweiligen Regionen mit ihrem Namen annotiert wurden. Dabei werden die Annotationstexte auf der linken bzw. rechten Seite der Visualisierung platziert und über Linien mit den Regionen verbunden. Die Reihenfolge der Platzierung von oben nach unten entscheidet sich über die y-Koordinate des jeweiligen Ankerpunktes. Je nach Anzahl der Annotationen können die Texte dabei in mehreren Spalten angeordnet werden. Der große Abstand zwischen Regionen und Text erschwert die Zuordnung enorm. Hinzu kommt,

dass sich bei vielen dicht nebeneinander verlaufenden, parallelen Linien einzelne Linien nur schwer verfolgen lassen.

Ein neueres Verfahren zur automatischen Annotation einzelner Illustrationen durch externe Annotationstexte wurde von [VOLLICK u. a. \[2007\]](#) vorgestellt. Dabei wird die Platzierung von Annotationen an einer Illustration beispielhaft durch einen Illustrator erstellt. Anschließend wird versucht, diese Art der Annotationsplatzierung auf andere Illustrationen automatisch zu übertragen. Dabei wird für jede Annotation mittels einer Energiefunktion über verschiedene Parameter (z.B. Ausrichtung und Abstand der Annotationen) versucht, die Differenz zur Vorlage zu minimieren. Dieser Prozess dauert selbst bei wenigen Annotationen (<20) mehrere Minuten und ist damit für interaktive Anwendungen nicht einsetzbar. Einen Beweis, dass das Verfahren schneller als ein Illustrator per Hand zu einem guten Ergebnis kommt, blieben [VOLLICK u. a. \[2007\]](#) schuldig. Das Verfahren ist als eine Art der 2D-Annotation einzuordnen, auch wenn die Illustrationen selbst teilweise 3D-Objekte zeigen. Die Dreidimensionalität wird jedoch weder im Algorithmus genutzt noch besonders unterstützt.

Grundsätzlich lässt sich festhalten, dass kein Verfahren zu automatischen Annotation von Regionen über mit Linien verbundene Annotationstexte direkt in einer zweidimensionalen Darstellung existiert.

7.3.2 3D Annotationstechniken

Ein erstes Verfahren zur automatischen Annotation von dreidimensionalen polygonalen Szenen wurde von [PREIM & RAAB \[1998\]](#) vorgestellt. Für ein anatomisches Ausbildungssystem (den ZOOMILLUSTRATOR ([PREIM & STROTHOTTE \[1996\]](#))) wurden einzelne Strukturen, wie Muskeln, Nerven oder Knochen, mit Texten automatisch annotiert. Die Texte wurden dabei in zwei Spalten links und rechts neben der Visualisierung platziert und über Linien mit der Struktur verbunden (Abb. 7.5). Die Ankerpunkte wurden über eine Skeletierung der Strukturen bestimmt. Dabei wurde besonderes Augenmerk auf elongierte und mehrfach verzweigende Strukturen gelegt. Diese wurden über mehrere Ankerpunkte mit dem Annotationstext verbunden.

Ein Verfahren zur automatischen Platzierung von Annotationen in Szenen, in denen sich der Betrachter frei bewegen kann, haben [BELL u. a. \[2001\]](#) vorgestellt. Beispielhaft werden die verschiedenen Gebäude des Campus der Columbia University annotiert. Dabei wird versucht, die Texte innerhalb des Objektes zu platzieren. Ist dieser Platz nicht ausreichend, so wird versucht, freien Platz außerhalb zu finden. Als freier Platz werden dabei Objekte betrachtet, die entsprechend zuvor bestimmt wurden, z.B. Rasen, Straßen oder der Hintergrund bzw. Himmel. Ist ein solcher Platz ebenfalls nicht zu finden, wird auf eine Annotation verzichtet.

Eine Übernahme der Ideen des point-based-labelings für dreidimensionale Visualisierungen präsentierte [OLBRICH \[2006\]](#). Dabei werden Verdeckungen von

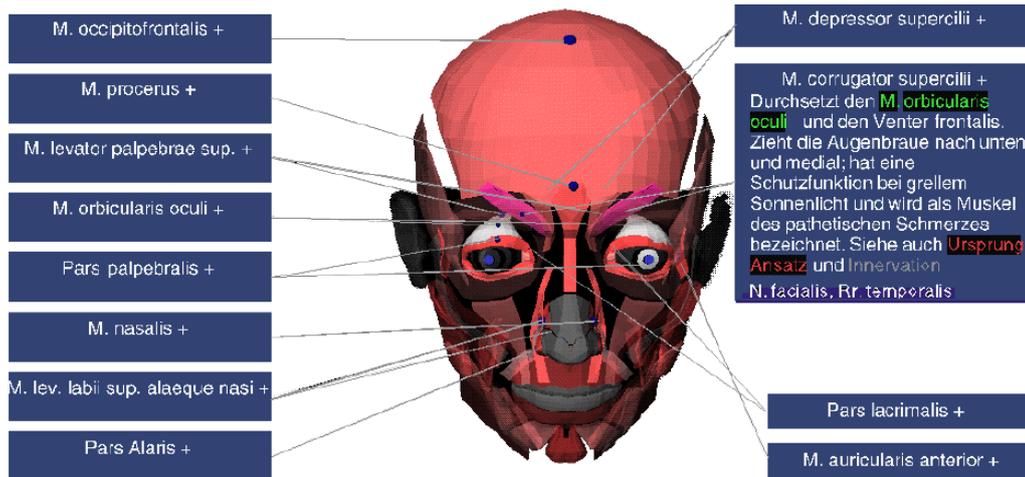


Abbildung 7.5: Annotation eines anatomischen Modells mit dem Verfahren von [Preim & Raab \[1998\]](#) (Abbildung mit freundlicher Genehmigung von Bernhard Preim).

Szenenbestandteilen durch Annotationstexte nicht berücksichtigt und die Annotationen direkt am Ankerpunkt des zu annotierenden Objektes platziert. Ein ausreichender Kontrast von Text und Hintergrund soll durch Boxen um den Text erreicht werden, was jedoch nur teilweise gelingt, da diese Boxen semi-transparent dargestellt werden, um wiederum verdeckte Objekte durchscheinen zu lassen.

[SONNET u. a. \[2004\]](#) platzierten ebenfalls Annotationstexte direkt an den betroffenen Objekten, ohne Überlappungen zu berücksichtigen. Dabei erscheint eine Annotationsbox, wenn man mit der Maus über ein Objekt fährt. Die Textboxen können frei in ihrer Form gestaltet werden und auch sehr lange Texte enthalten, die dann mittels eines Scrollbalkens erreicht werden können.

Für die Annotation medizinischer Objekte zu Ausbildungszwecken wurden bei [GÖTZELMANN u. a. \[2005\]](#) interne Annotationen den externen vorgezogen. Dabei wird der Annotationstext einer Skelettierung der projizierten Objektoberfläche angepasst. Die dadurch teilweise sehr starke Krümmung des Textes in Verbindung mit einem inhomogenen Hintergrund erschwert die Lesbarkeit enorm.

Arbeiten zur internen Annotation medizinischer Visualisierungen haben [ROPINSKI u. a. \[2007\]](#) sowie [BÜRGER u. a. \[2008\]](#) vorgestellt. Es wird dabei versucht, den Annotationstext möglichst direkt auf der Struktur darzustellen (Abb. 7.6b). Die Krümmung der Struktur wird analysiert und der Text entsprechend verformt. So soll eine bessere Formwahrnehmung der Struktur möglich sein. Dabei wird allerdings in Kauf genommen, dass die Texte teilweise bis zur kompletten Unlesbarkeit verformt werden. Hinzu kommt der oft schlechte Kontrast der Texte bezüglich der strukturieren Oberflächen.

Bei [GÖTZELMANN u. a. \[2006\]](#) wurde versucht, kontextuell zueinander gehörende Annotationen zu gruppieren. Hierbei blieb jede Annotation für sich erhalten,

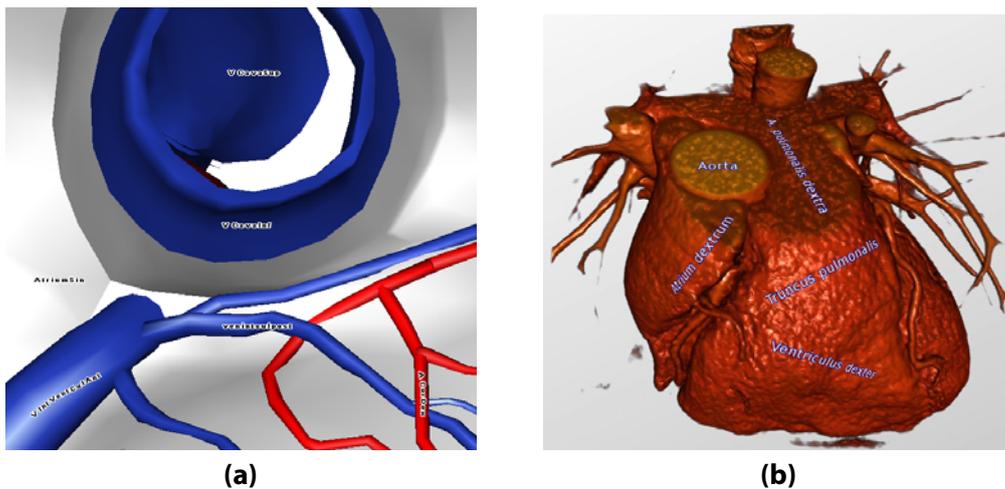


Abbildung 7.6: (a): Interne Annotation medizinischer Strukturen durch [Götzelmann u. a. \[2005\]](#). Die Annotationstexte wurden entlang einer Skelettierung der projizierten Oberfläche verformt und sind nur schwer lesbar (Bild mit freundlicher Genehmigung von Timo Götzelmann). (b): Interne Annotation medizinischer Strukturen durch [Ropinski u. a. \[2007\]](#). Die Annotationen wurden für eine bessere Tiefenwahrnehmung entsprechend der Krümmung der Oberfläche verformt (Bild mit freundlicher Genehmigung von Timo Ropinski).

konnte also dem Zielobjekt zugeordnet werden. Lediglich die Position der Texte wurde gruppenweise angepasst. Unterschiedliche Gruppen wurden dabei mit verschiedenen Farben hervorgehoben. Ausgehend von einem Schwerpunkt aller Ankerpunkte einer Gruppe wird die Box für die späteren Annotationstexte in mehrere Richtungen stückweise verschoben, bis eine überlappungsfreie Darstellung möglich ist. Es bleibt jedoch unklar, ob eine solche Form der Annotationsdarstellung von der avisierten Zielgruppe (Mediziner in der Ausbildung) überhaupt gewünscht wird oder vom Betrachter als Vorteil erachtet würde.

Ein umfängliches Verfahren zur 3D-Annotation haben [ALI u. a. \[2005\]](#) vorgestellt. Sie zielen auf die Platzierung von Annotationen sowohl extern auf einer die Darstellung umgebende Hülle als auch auf Annotationen innerhalb der Strukturen. Das Verfahren nutzt kompakte Darstellungen opaquer Strukturen (z.B. ein Motorblock mit Einzelteilen). Zur Bestimmung der sichtbaren Objekte, der Ankerpunkte sowie der umgebenden Hülle zur Platzierung der externen Annotationen wird die Darstellung in mehreren Zwischenstufen gerendert und analysiert:

1. Zunächst wird jede Struktur über eine eindeutige Farbe in einem IDBuffer gerendert (Abb. 7.7b). Jede homogene Farbfläche ist eindeutig einer Struktur zuzuordnen.
2. Auf dem zweidimensionalen Bild des IDBuffers wird eine Distanztransformation durchgeführt, so dass für jede Region ein Ankerpunkt bestimmt werden kann (der am weitesten im Inneren der Fläche liegende Punkt) (Abb. 7.7c).

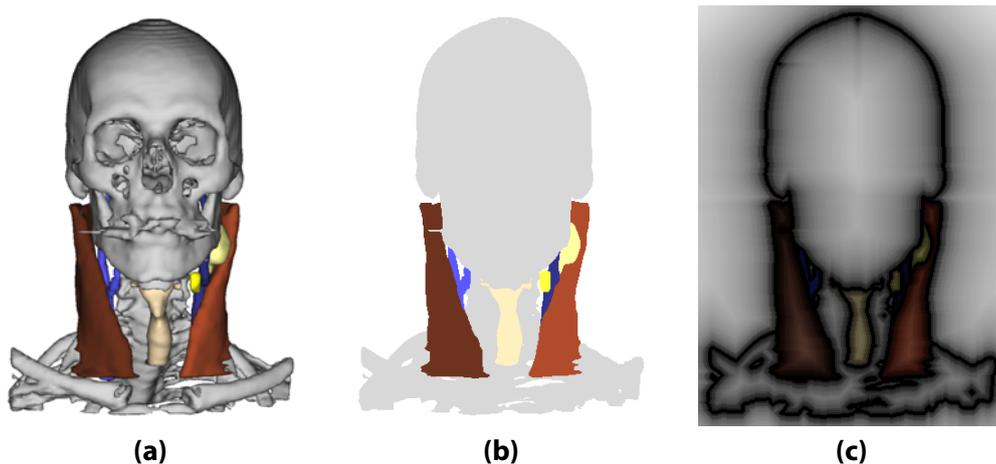


Abbildung 7.7: (a): 3D-Szene des Halses mit opaken Strukturen. (b): IDBuffer erstellt mit dem Verfahren von Ali u. a. [2005].^a Distanztransformation, ausgeführt auf dem IDBuffer von Ali u. a. [2005] (c).^b

^aDie Farben sind aus Gründen der verbesserten Darstellung den realen Farben der Strukturen angepasst.

^bHelle Farben entsprechen großen Distanzen.

3. Eine die Szene umgebende Hülle wird in einem gewissen Mindestabstand ebenfalls über die Distanztransformation bestimmt.
4. Die Annotationstexte werden als Massepunkte abstrahiert, die möglichst nah am Ankerpunkt auf der Hülle platziert werden. Dies bringt einige Vereinfachungen bei den algorithmischen Berechnungen mit sich, hat aber auch den Nachteil, dass Annotationstexte nur einzeilig mit einer fixen Höhe sein dürfen.
5. Sollten sich die Verbindungslinien zweier Annotationen kreuzen, so werden ihre Positionen getauscht. Mithilfe eines einfachen Feder-Masse-Modells werden die Annotationen gleichmäßig verteilt um durch größere Zwischenräume zwischen den Texten deren Lesbarkeit zu erhöhen.

Um ein Springen der Annotationen bei der interaktiven Exploration der Szene (z.B. Rotation) zu vermeiden (die Frame-Kohärenz soll erhalten werden), wird eine einmal bestimmte Position von Ankerpunkten und Annotationstexten versucht so lange zu halten, wie sich keine Konflikte ergeben, z.B. wenn ein Ankerpunkt sich außerhalb des Objektes befindet oder sich Linien überschneiden. Damit wird ein ruhigeres Erscheinungsbild der gesamten Darstellung erreicht und die Annotationen ziehen nicht unnötig viel Aufmerksamkeit auf sich. Das Verfahren von ALI u. a. [2005] ist echtzeitfähig und somit für den interaktiven Einsatz geeignet. Die Techniken von ALI u. a. [2005] wurden unter anderem von LI u. a. [2007] angewendet, um ausdrucksstarke Cut-Away Illustrationen automatisch zu annotieren.

Ein neueres Verfahren präsentierten [STEIN & DÉCORET \[2008\]](#). Dabei werden Annotationen initial möglichst nah am Objekt platziert und anschließend durch die Optimierung einer Energiefunktion die Annotation möglichst ohne Überlappungen und Überschneidungen platziert. Es wird auf eine umgebende Hülle wie bei [ALI u. a. \[2005\]](#) verzichtet und stattdessen eine freie Platzierung der Annotationen an jeder Stelle der Darstellung erlaubt. Somit lassen sich auch nicht kompakte Szenen annotieren. In die Energiefunktion fließen unter anderem:

- a) die Länge der Annotationslinie,
- b) die Skalierung der Annotation (bei geringem Platzangebot werden die Annotationen skaliert),
- c) der Winkel der Verbindungslinie (die Linie sollte entlang von Vorzugsrichtungen wie waagrecht oder senkrecht orientiert sein) und
- d) die Wichtigkeit möglicherweise verdeckte Polygone und Objekte ein.

Bei der Annotation von nicht kompakten Szenen ist meist wenig oder kein freier Hintergrund zur Annotationsplatzierung vorhanden. Daher wurde eine Wichtigkeit von Polygonen eingeführt, womit Annotationen vorzugsweise unwichtigere Polygone verdecken sollten. Um bei der sequentiellen Platzierung aller Annotationstexte den Suchraum möglichst klein zu halten und Konflikte mit schon platzierten Annotationen zu vermeiden, wird bei [STEIN & DÉCORET \[2008\]](#) von jedem neuen Ankerpunkt aus eine Schattenprojektion zu schon bestehenden Annotationen berechnet. Positionen im gedachten Schatten dieser Annotationen verbieten sich von vornherein für die neue Annotation, da es zwangsläufig zu Überschneidungen kommen würde. Das Verfahren von [STEIN & DÉCORET \[2008\]](#) ist in der Gestaltung der Annotationen frei. So können auch Bilder und längere Texte in den Annotationsboxen platziert werden.

Eine Analyse der Verhaltens von Annotationen in interaktiven Darstellungen wurde von [PREIM u. a. \[1995\]](#) im Rahmen des ZOOMILLUSTRATORS (siehe auch Seite 40) durchgeführt. Dabei ist zu beachten, dass ein anfänglich gutes initiales Layout der Annotationen durch die interaktive Manipulation der Szene nicht aufrecht erhalten werden kann. [PREIM u. a. \[1995\]](#) nahmen dies in Kauf und erzeugten ein neues Layout nur auf Initiative des Nutzers hin. Eine andere Möglichkeit ist die Reduzierung der dargestellten Annotationen bei einer Interaktion sowie eine automatische Layoutanpassung, wenn bestimmte Schwellenwerte, z.B. die Länge von Verbindungslinien, überschritten werden.

7.3.3 Annotationen in medizinischen Systemen

CLEYNENBREUGEL u. a. [1996] präsentierten einen Ansatz, bei dem der Radiologe Punkte im 2D-Schichtbild anklicken und mit einem textuellen oder gesprochenen Kommentar versehen kann. Dieser wird in einer extra Textdatei gespeichert und im Bild mit einer Nummer ohne Berücksichtigung möglicher Verdeckungen annotiert. Ein ähnlich beschränktes System präsentierten CHRONAKI u. a. [1997]. CAI u. a. [2001] zeigten ein System, bei dem Ärzte zur Unterstützung der Kommunikation einfache Primitive, wie Kreise und Rechtecke in Schichtbildern, einzeichnen konnten.

LOBER u. a. [2001] präsentieren die *Image Markup Language (IML)*, mit der 3D-Visualisierungen aus einzelnen, starren Sichten mit Annotationen versehen werden konnten. Die Annotationstexte werden dabei in einer separaten Spalte untereinander platziert, ohne auf Überschneidungen bei den Verbindungslinien zu achten. In Kombination mit dem großen Abstand der Annotationstexte zu den jeweiligen Objekten ist eine solche Visualisierung kaum lesbar. Die Darstellungen von LOBER u. a. [2001] sind nicht interaktiv und die Texte immer nur aus einer vorher definierten Blickrichtung sichtbar.

Ein weiteres System zu (manuellen) Annotation von Schichtbildern präsentierten GOEDE u. a. [2004]. Die eingezeichneten Linien und Polygone werden separat zum Bild in einer XML-Struktur gespeichert und können so effektiv weiter verteilt und verschickt werden. Die Software zielt auf Autoren ab, die einzelne Schichtbilder zu Ausbildungszwecken annotieren möchten.

Auch in aktuellen Systemen zur medizinischen Ausbildung HÖHNE u. a. [2008] kommen automatische Annotationen von 3D-Darstellungen zum Einsatz. Dabei werden Annotationen im Voxel-Man HÖHNE u. a. [2008] alle am Rand untereinander platziert, ohne dabei auf Überschneidungen von Linien oder die Verdeckung von Strukturen zu achten (Abb. 7.8). Die Lesbarkeit solcher Annotationen ist dadurch bis zur Unbrauchbarkeit eingeschränkt.

7.3.4 Zusammenfassung der verwandten Arbeiten

Der Blick auf bestehende Systeme zur Annotation von medizinischen Visualisierungen (insbesondere von 2D-Schichtbildern) zeigt die Notwendigkeit eines automatischen Annotationsframeworks sowohl für 3D- als auch für 2D-Darstellungen. Zwar sind manuelle Annotationen gerade zur Vermittlung von speziellen Inhalten, wie der Lokalisation von Läsionen oder Stenosen, unerlässlich. Eine automatische Platzierung dieser Annotationen nach ästhetischen Gesichtspunkten, wie der Verdeckung und der Lesbarkeit, ist dennoch wünschenswert. Gleiches gilt für die automatische Annotation bereits segmentierter Strukturen für eine bessere Orientierung, sowohl in 2D als auch in 3D.

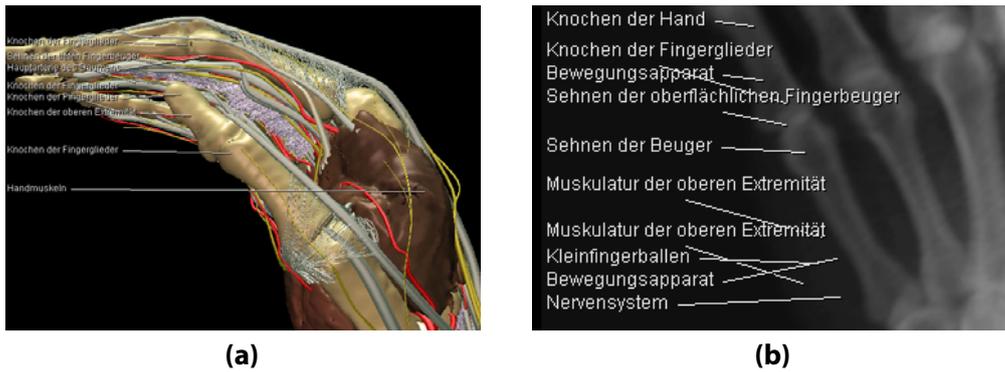


Abbildung 7.8: Screenshots von Annotationen einer 3D-Darstellung der Handanatomie (a) und eines Röntgenbildes (b) aus der Lernumgebung Voxel-Man (Höhne u. a. [2008]).

Für die Annotation dreidimensionaler medizinischer Visualisierungen in Echtzeit bietet das Verfahren von ALI u. a. [2005] einen sehr guten Ausgangspunkt, so z.B. die effiziente Ermittlung von Ankerpunkten und der konvexen Hülle über einen IDBuffer. Auf 2D-Schichtbilder sind die 3D-Verfahren nicht ohne weiteres übertragbar. Dies liegt im Wesentlichen in der Tatsache begründet, dass im Gegensatz zu den präsentierten 3D-Szenen bei Schichtbildern keine kompakte Darstellung vorherrscht, die von einer umgebenden Hülle aus annotiert werden kann. Es stellen sich ganz neue Probleme, vor allem, wenn es darum geht, mögliche Platzierungen für die jeweiligen Annotationstexte zu finden. Hinzu kommt, dass im Gegensatz zu ALI u. a. [2005] nicht von einer quasi Eindimensionalität der Annotationen mit nur einer festen Texthöhe (und Breite) ausgegangen werden kann.

7.4 Verfahren zur Beschriftung von 2D-Darstellungen

Um Strukturen in 2D-Schichtbilddaten angemessen automatisch annotieren zu können, wurde im Rahmen dieser Arbeit ein neues Verfahren entwickelt, welches den besonderen Anforderungen medizinischer Schichtbilddarstellungen, insbesondere der Schichtbildkohärenz, gerecht wird. Unter der Schichtbildkohärenz wird die Wahrung einer möglichst konstanten Position einer Annotation über mehrere Schichtbilder hinweg verstanden. Dies befördert die Lesbarkeit und erleichtert das Verfolgen von Strukturen über mehrere Schichtbilder hinweg.

7.4.1 Grundalgorithmus zur Annotationsplatzierung in 2D-Darstellungen

Das hier vorgestellte Verfahren annotiert Schichtbilder automatisch in Echtzeit (Abb. 7.2). Es werden dabei die folgenden Schritte durchlaufen:

1. Ermitteln der Zusammenhangskomponenten
2. Vorberechnung aller möglichen Positionen jeder Annotation
3. Bestimmung von Ankerpunkten
4. Initialplatzierung jeder Annotation
5. Auflösung möglicher Konflikte durch Überlagerungen und Überlappungen

Ermitteln der Zusammenhangskomponenten

Alle Berechnungsschritte fußen auf zweidimensionalen Feldern $p(x, y)$ der Größe des jeweiligen Schichtbildes. Die Zusammenhangskomponenten werden auf Basis der *Multi Coded Segmentation Masks* (MCSM)²⁹ unter Zuhilfenahme der Sichtbarkeitsinformationen der Strukturen ermittelt.³⁰ Als zu einer Komponente gehörig werden alle Voxel einer Struktur betrachtet, die über eine 4er Nachbarschaft miteinander verbunden sind. Gehört ein Voxel zu mehreren Strukturen gleichzeitig (z.B. einem Lebersegment und dem Leberparenchym), so wird er der Struktur mit der höheren Wichtigkeit zugeordnet. Für den seltenen Fall der Gleichheit der Wichtigkeiten wird er der kleineren der Strukturen zugeordnet. Aus der Analyse ergibt sich ein Feld der Zusammenhangskomponenten $p_{comp}(x, y)$ mit:

$$p_{comp}(x, y) = \begin{cases} 0 & \text{keine sichtbare Struktur} \\ i & \text{mit } i \text{ als Nummer der Zusammenhangskomponente} \end{cases} \quad (7.1)$$

²⁹Die Datenstruktur der *Multi Coded Segmentation Masks* wird in Kapitel 9 genauer vorgestellt.

³⁰Die Sichtbarkeitsinformationen werden mit dem Verfahren aus Kapitel 4 bestimmt.

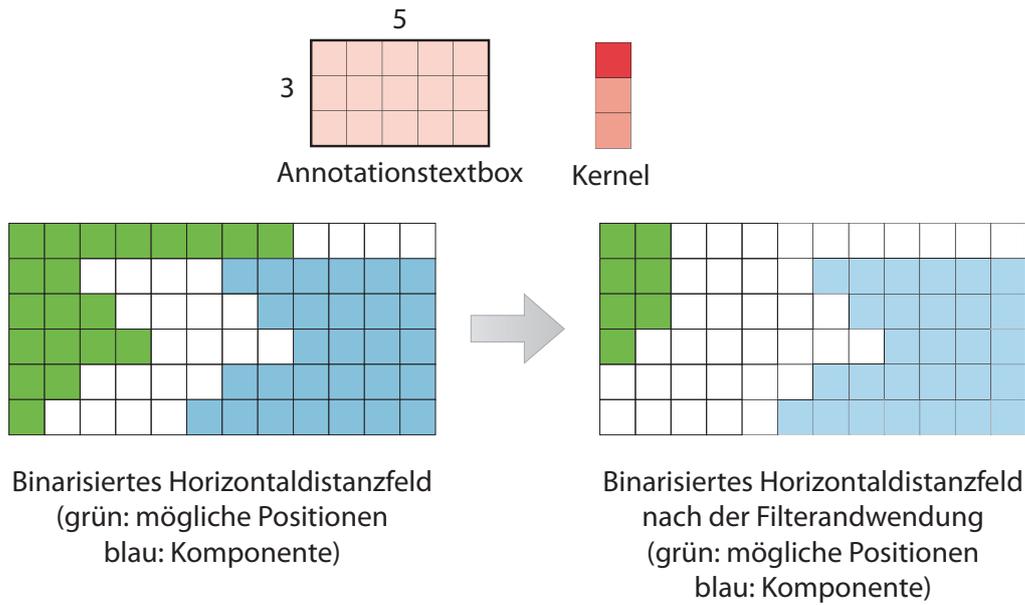


Abbildung 7.9: Ermittlung aller möglichen Positionen (grün) zur Platzierung einer Annotation in der Umgebung einer Struktur (blau).

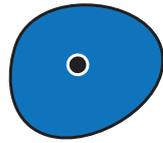
Vorbereitung möglicher Positionen

Um für jede Annotation alle möglichen Positionen berechnen zu können, an denen der Annotationstext platziert werden kann, wird eine Distanztransformation mit Manhattan-Distanz berechnet: für den horizontalen Platz, der an jedem Voxel maximal zur Verfügung steht, um eine Annotationstextbox zu platzieren. Dabei wird das Feld der Komponenten p_{comp} zeilenweise von rechts abgetastet und für die Distanztransformation $p_{horz}(x, y)$ werden folgende Werte gesetzt:

$$p_{horz}(x, y) = \begin{cases} 0, & \text{für Randvoxel oder Komponentenvoxel} \\ p_{horz}(x + 1, y) + 1, & \text{für Hintergrundvoxel (sonst)} \end{cases} \quad (7.2)$$

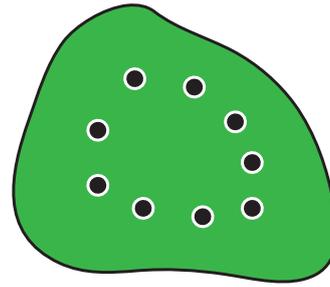
Im Folgenden werden daraus für jeden Annotationstext alle möglichen Positionen einer Platzierung ermittelt. Dazu wird p_{horz} jeweils schwellenwertbasiert binarisiert. Als Schwellenwert dient dabei die jeweilig horizontale Ausdehnung der Annotationstextbox. Auf das binarisierte Feld p_{horz} wird zusätzlich ein Filter in Form einer vertikalen Linie der Höhe der Annotationstextbox angewendet (Abb. 7.9). So wird garantiert, dass im Ergebnisfeld p_{free} nur Voxel mit 1 bewertet sind, bei denen der Text über ausreichend Voxelzeilen hinweg in der nötigen Breite platziert werden kann und somit Überlappungen ausgeschlossen werden können.

Im Ergebnis erhält man für jede Annotation eine binäre Maske p_{free} mit:



Kleine Struktur

Nur ein Ankerpunkt an der Stelle der größten Entfernung zum Rand wird ermittelt



Große Struktur

Mehrere Ankerpunktkandidaten mit einer Maximalentfernung zum Rand und einem Mindestabstand werden berechnet

Abbildung 7.10: Bei großen Strukturen werden im Gegensatz mehrere Ankerpunktkandidaten ermittelt, die alle eine Mindestentfernung zum Rand haben und aus denen dann der ausgewählt wird, der zum Annotationstext die geringste Entfernung hat.

$$p_{free}(x, y) = \begin{cases} 1 & \text{Position mit ausreichend Platz für den Text} \\ 0 & \text{sonst} \end{cases} \quad (7.3)$$

die alle möglichen Positionen für die Platzierung eines Annotationstextes enthält. Dabei gilt die linke obere Ecke der Annotationstextbox als Position des Textes.

Bestimmung von Ankerpunkten

Im Folgenden wird für jede Annotation der Ankerpunkt im Inneren der jeweiligen Komponente bestimmt. Dazu wird eine Distanztransformation auf den Komponenten durchgeführt. Als Ankerpunkt wird derjenige Punkt mit der größten Distanz zum Rand der Komponente gewählt. Bei sehr großflächigen Strukturen würde jedoch ein Ankerpunkt im Zentrum der Fläche zu einer zu langen Verbindungslinie führen. Aus diesem Grund gilt für die Distanz eines Ankerpunktes zum Rand der Struktur ein Maximalwert. Daraus resultierend werden für großflächige Strukturen mehrere gleichwertige Ankerpunkte ermittelt (Abb. 7.10). Aus diesen wird jeweils der Kandidat gewählt, der am nächsten zum letztendlich platzierten Annotationstext liegt und so eine möglichst kurze Verbindungslinie garantiert.

Initialplatzierung jeder Annotation

Für jeden Ankerpunkt wird ein einfaches Distanzfeld p_{anchor} bestimmt, welches im Folgenden genutzt wird, um den Annotationstext möglichst dicht an der jeweiligen Komponente zu platzieren. Für jede Annotation i wird aus dem Feld aller in Frage kommenden Positionen p_{free_i} und dem Distanzfeld des Ankerpunktes p_{anchor_i} ein Ergebnisfeld p_{sum_i} mit

$$p_{sum_i}(x, y) = p_{free_i} \cdot p_{anchor_i} \quad (7.4)$$

berechnet. Das Minimum von p_{sum_i} stellt die Position für die Initialplatzierung der Annotationstextbox i dar.

Das Verfahren hat bis hierhin dafür gesorgt, dass kein Annotationstext eine Komponente überlappt. Ein Mindestabstand zu den Komponenten wird zu Beginn in die Größe der Annotationstextbox eingerechnet und auch schon bei der Initialplatzierung berücksichtigt.

Auflösung möglicher Konflikte

Bei der Initialplatzierung der einzelnen Annotationstexte kann es zu Überlappungen und Überschneidungen der Annotationstexte und Verbindungslinien untereinander kommen. Mögliche Konflikte werden zum Abschluss des Prozesses in mehreren Iterationsschritten gelöst. Dabei werden pro Iterationsschritt alle vorhandenen Konflikte durch Neuplatzierung der beteiligten Annotationstexte gelöst, was nicht ausschließt, dass dadurch neue Konflikte auftreten, die in weiteren Schritten gelöst werden müssen. Es ist nicht garantiert, dass das Verfahren konvergiert, da es Konstellationen geben kann, in denen nicht alle Annotationen konfliktfrei platziert werden können. Daher wird dieser Prozess nach 10 Iterationsschritten gestoppt und eine möglicherweise aus Sicht des Betrachters nicht vollständig optimale Lösung angezeigt.

Es wird zwischen zwei Arten von Konflikten unterschieden:

1. Überschneidungen von Verbindungslinien
2. Überlappungen von Annotationstextboxen

Um Überschneidungen von Verbindungslinien zu lösen, werden die Positionen der Textboxen getauscht, wobei aufgrund der verschiedenen Textgrößen nicht garantiert ist, dass jeweils genau an der Position der anderen Textbox die getauschte Textbox auch platziert werden kann, ohne andere Komponenten dabei zu überlappen (Abb. 7.11). Daher sucht jede Textbox, von der neuen Position ausgehend, nach der am nächsten gelegenen freien Position in ihrem Feld p_{free} . Dies geschieht immer in entgegengesetzter Richtung zur alten Position, damit beide Textboxen nicht die Möglichkeit haben, wieder in ihre alten Positionen zurückzufallen.

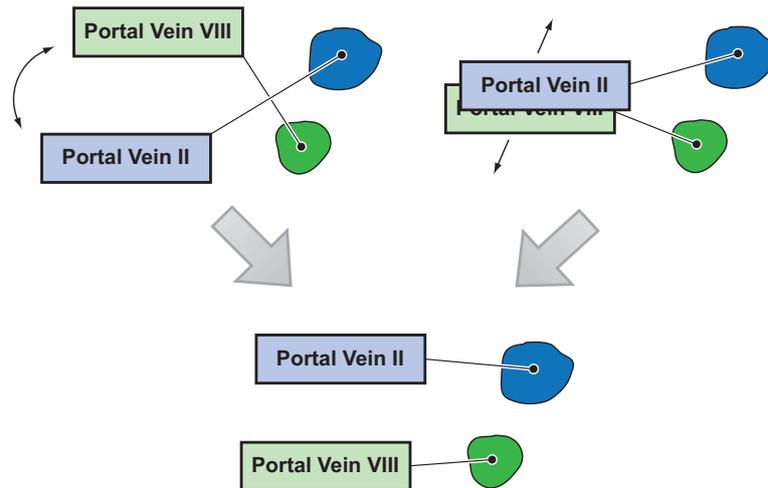


Abbildung 7.11: Lösen von Konflikten bei der Platzierung von 2D-Annotationen. Überlappen sich die Verbindungslinien zweier Annotationen (links), so werden ihre Positionen getauscht. Überlappen sich zwei Annotationstexte (rechts), so wird für jeden Text eine neue Position gesucht. Die Suche erfolgt jeweils von der Überlappungsstelle weg. Im Umkreis der neuen Position wird in beiden Fällen für jeden Annotationstext die nächste freie Position gesucht.

Um Überlappungen von Textboxen zu lösen, werden sich überlappende Textboxen zu gleichen Teilen voneinander weg bewegt. Sollten an den jeweils neuen Positionen keine Platzierungen möglich sein, wird ähnlich wie bei dem Positionstausch bei Linienüberschneidungen nach der nächstliegenden freien Position in p_{free} gesucht, wobei auch hier die Suchrichtung vom Ort der Überlappung zeigt (Abb. 7.11).

Nicht berücksichtigt werden dabei Überschneidungen von Verbindungslinien mit Textboxen. Ebenfalls nicht berücksichtigt werden Überschneidungen von Verbindungslinien einer Gruppe mit anderen Verbindungslinien. Diese sind ohne eine Auflösung oder Aufteilung der Gruppe(n) nicht lösbar.

7.4.2 Besondere Aspekte der Visualisierung von Annotationen in 2D-Schichtbildern

Neben der reinen Platzierung und Gruppierung von Annotationen sind bei der Annotation von 2D-Schichtbildern weitere Besonderheiten, wie die Schichtbildkohärenz und stilistische Aspekte, zu beachten.

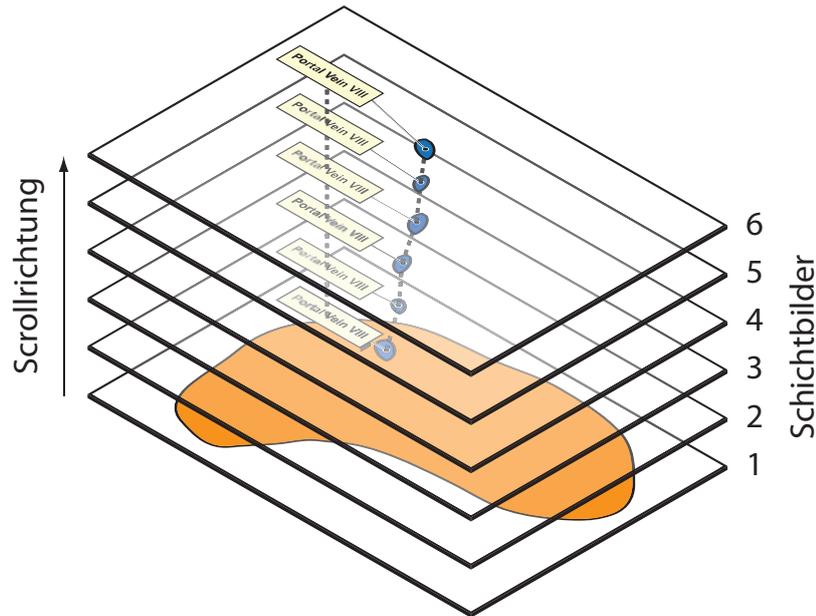


Abbildung 7.12: Schichtbildkohärenz bei 2D-Annotationen. Ein Annotationstext hält über mehrere Schichten seine Position, während die Struktur selbst ihre Position ändern kann.

Schichtbildkohärenz

Wie schon in Abschnitt 7.2 angesprochen, kommt es zu sprunghaften Positionswechseln von Annotationen, wenn der Nutzer durch die Schichten scrollt. Dies erschwert die Lesbarkeit und die Verfolgung von Annotationen einer Struktur über mehrere Schichtbilder hinweg. Daher versucht der hier präsentierte Algorithmus, die Position eines Annotationstextes über mehrere Schichten hinweg konstant zu halten. Es werden nur die jeweiligen Ankerpunkte und die Verbindungslinien verschoben (Abb. 7.12). Ein Annotationstext hält so lange seine Position, wie die Struktur im Schichtbild sichtbar ist und keine Konflikte, wie Überlappungen, auftreten. Dies resultiert in nicht optimalen Platzierungen der Annotation im jeweiligen Schichtbild, führt jedoch zu einer verbesserten Lesbarkeit und Verfolgbarkeit von Annotationen.

Verhalten bei großflächigen Strukturen

Da Annotationen möglichst nah an den jeweiligen Strukturen platziert werden sollen, dabei jedoch keine anderen Strukturen verdeckt werden sollen, stellen großflächige, kompakte Strukturen, wie bspw. die Leber, ein Problem dar. Sie nehmen oft einen Großteil der Fläche des Schichtbildes ein und erzwingen die Platzierung von Annotationen für kleine Strukturen (bspw. intrahepatische Gefäße) weit von diesen entfernt (Abb. 7.13(a)). Gerade solche großen Strukturen wie die Leber dienen meist nur als Kontextinformation. Eine Überlappung mit Annotationstexten ist daher akzeptabel und würde die Darstellung nicht negativ

beeinflussen. Aus diesem Grund nutzt das vorgestellte Verfahren verschiedene Wichtigkeiten von Strukturen, um Annotationen wichtiger Strukturen innerhalb weniger wichtiger Strukturen zu platzieren (Abb. 7.13(b)). Dadurch können Annotationen bspw. wichtige intrahepatische Gefäße direkt an diesen platziert werden, obwohl sie dabei die Leber überdecken.

Weitere ästhetische Aspekte

Um eine starke Konzentration von Annotationstexten zu vermeiden, können Annotationen, die verschiedene Teile ein und derselben Struktur annotieren, automatisch zusammengefasst werden (Abb. 7.14). Eine solche **Gruppierung** erfolgt, wenn der Abstand zwischen den Ankerpunkten der betreffenden Teile der Struktur einen Schwellenwert unterschreitet. Dieser Abstand kann applikationsbedingt variieren. Es gibt Fälle, in denen Strukturen über einen größeren Raum gruppiert werden sollen (bspw. alle linksseitigen Lymphknoten in einem Schichtbild des Halses) oder nur Strukturteile in einem sehr kleinen Gebiet. Ein Beispiel dafür sind Teile von Gefäßbäumen der Leber, die in einem Schichtbild innerhalb eines gewissen Gebietes in Form vieler kleiner getrennter Regionen auftreten, z.B. wenn ein Gefäß die Ebene des Schichtbildes mehrfach durchstößt. Für solche Fälle ist eine Annotation jeder einzelnen Region mit dem gleichen Annotationstext verwirrend und eine Gruppierung daher sinnvoll.

Bei einer Gruppierung von mehreren Komponenten werden alle Distanzfelder der Ankerpunkte p_{anchor_j} der an der Gruppe beteiligten Komponenten j addiert und wie schon bei der einfachen Annotation mit dem Feld aller möglichen Positionen für den Annotationstext p_{free} ³¹ multipliziert:

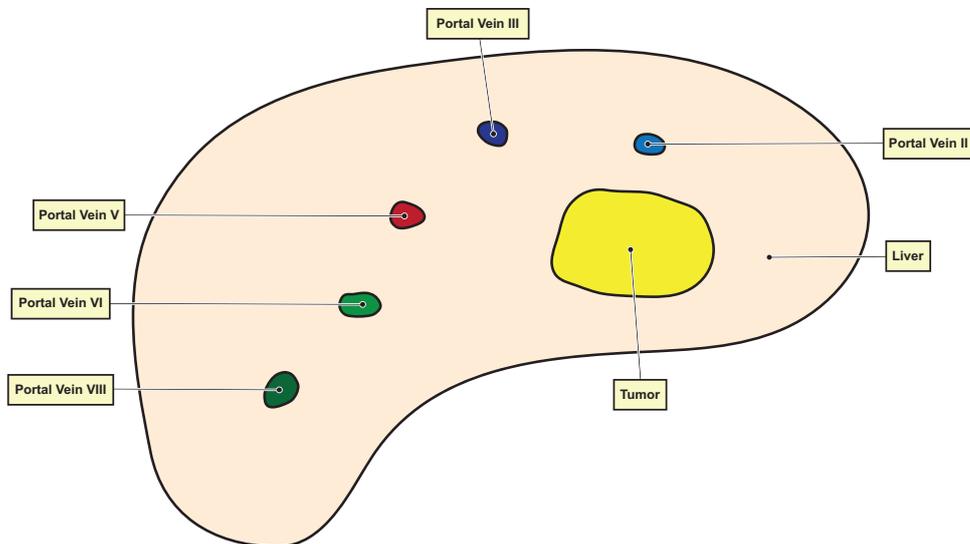
$$p_{sum}(x, y) = \left(\sum_{j=1}^n p_{anchor_j} \right) \cdot p_{free} \quad (7.5)$$

Das Minimum dieses Feldes ist wiederum die Initialplatzierung der Gruppenannotation, mit dem Unterschied, dass dies eine Position ist, die möglichst nah an allen Komponenten der Gruppe liegt.

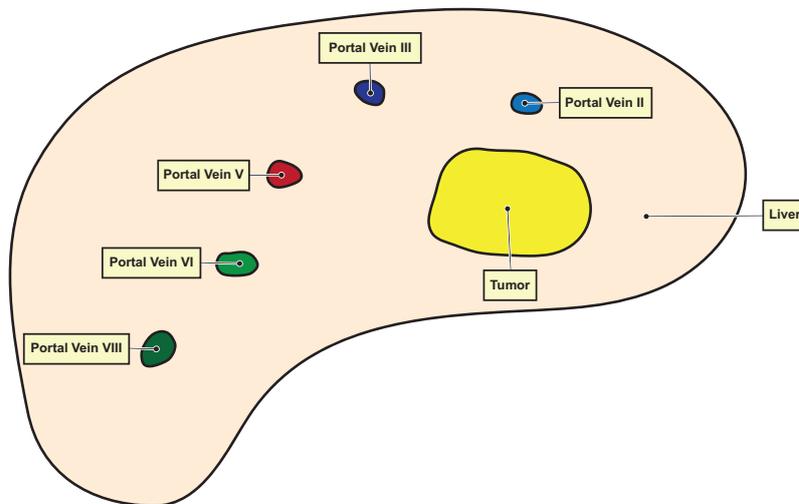
Sehr kleine Strukturen von nur wenigen Voxeln Größe würden durch einen Ankerpunkt unnötig verdeckt. Aus diesem Grund werden kleine Strukturen „eingekreist“ und über eine Verbindungslinie mit der Annotation verknüpft (Abb. 7.14).

Zur Stärkung des Kontrastes kann der Annotationstext hinterlegt werden. Die **Hinterlegung** kann jedoch mit einer gewissen Transparenz versehen werden, um bei 2D-Schichtbildern darunter liegende Grauwertinformationen nicht vollständig zu verdecken. Die Farbe der Hinterlegung kann sowohl einheitlich als auch der Farbe der jeweiligen Struktur angepasst sein (Abb. 7.18 für ein Beispiel bei

³¹Dieses Feld ist für alle beteiligten Komponenten gleich, da sie mit dem selben Text annotiert werden. Es ist daher egal, welches der Felder herangezogen wird.



(a)



(b)

Abbildung 7.13: Berücksichtigung von Wichtigkeiten bei der Platzierung von Annotationen. Werden keine Unterschiede zwischen den Strukturen gemacht, müssen Annotationen kleiner Strukturen oft weit von diesen entfernt platziert werden (a). Durch Nutzung verschiedener Wichtigkeiten können Annotationen wichtiger Strukturen weniger wichtige (und meist größere) Strukturen überlappen (b).

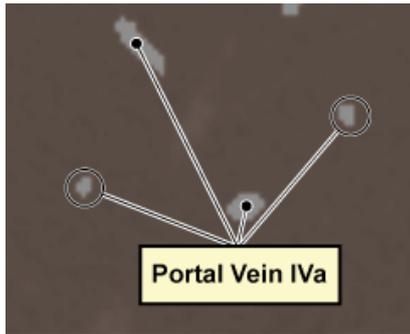


Abbildung 7.14: Automatische Gruppierung von Annotationen. Anstatt mehrere Teile einer Struktur mit den jeweils gleichen Annotationstexten zu versehen, werden die Annotationen gruppiert und nur noch ein Text für alle Teile wird angezeigt. Ebenfalls zu erkennen ist die Annotation sehr kleiner Strukturen mit einer Umkreisung, um deren Sichtbarkeit zu garantieren.

3D-Annotationen). Solange ein ausreichender Kontrast die Lesbarkeit unterstützt, ist dies vor allem eine Frage der persönlichen Präferenz des Nutzers (siehe Abschnitt 7.6).

Jede Annotation ist möglichst nah zur jeweiligen Struktur platziert, hält aus ästhetischen Gründen jedoch einen **Mindestabstand** zur Struktur selbst und zu allen anderen Strukturen und Annotationen.

7.5 Verfahren zur Beschriftung von 3D-Szenen

Das in dieser Arbeit entwickelte Verfahren zur automatischen Annotation von 3D-Szenen in Echtzeit baut auf dem Verfahren von [ALI u. a. \[2005\]](#) auf, das sich als gute Grundlage erwiesen hat. Jedoch haben sich bezüglich der in Abschnitt 7.2 definierten Anforderungen an die Annotation medizinischer Visualisierungen besondere Defizite gezeigt. Das Verfahren von [ALI u. a. \[2005\]](#) nimmt alle Objekte der Szene als opaque an und kann nur solche annotieren. Die Nutzung von Transparenzen bei der Darstellung sich überlappender medizinischer Strukturen steht dazu im Gegensatz (Abb. 7.7). Ebenso ist eine Annotation von nicht sichtbaren bzw. komplett verdeckten Strukturen bei [ALI u. a. \[2005\]](#) nicht möglich. Bei dem in dieser Arbeit entwickelten Verfahren werden folgende Schritte durchlaufen:

1. Zusammenfassen von sich nicht überlagernden Strukturen aus der aktuellen Sichtrichtung in jeweils einem IDBuffer
2. Erstellen eines IDBuffers mit allen Strukturen für die Ermittlung des umgebenden Freiraums zur Annotationsplatzierung
3. Distanztransformation auf jedem IDBuffer zur Ermittlung der Ankerpunktkandidaten
4. Sichtbarkeitstest für jede Struktur an den Ankerpunktkandidaten
5. Auswahl des Annotationsstils abhängig vom Ergebnis der Sichtbarkeitstests
6. Platzierung der Annotationen auf einer umgebenden Hülle

7.5.1 Berechnung der Multi-IDBuffer

Um das Problem zu lösen, werden in dieser Arbeit die *Multi-IDBuffers* eingeführt. Wo bei [ALI u. a. \[2005\]](#) ein einzelner IDBuffer genutzt wurde, um per Distanztransformation für die sichtbaren Strukturen die Ankerpunkte sowie den freien, für die Annotationstexte nutzbaren Bereich, zu ermitteln, kommen bei dem neuen Verfahren mehrere IDBuffer zum Einsatz (Abb. 7.15). Im einfachsten Fall wird jede Struktur der Szene, die annotiert werden soll, in einem eigenen IDBuffer gerendert, um so (ebenfalls per Distanztransformation) die Ankerpunkte zu bestimmen. Aus Gründen der Performance wird die Zahl der IDBuffer jedoch enorm reduziert, indem Sichtbarkeitsdaten, wie sie im Verfahren zur Sichtpunktbestimmung berechnet wurden (siehe Kapitel 4), genutzt werden. Es werden all jene Strukturen zusammen in einem IDBuffer gerendert, die sich nicht gegenseitig überlappen oder verdecken. Genutzt werden dazu die Daten der Kameraposition in den Sichtbarkeitsdaten, die der aktuellen Kameraposition am nächsten liegt.

Für die Darstellung in Abbildung 7.15 sind zehn Strukturen sichtbar. Von diesen sollen neun Strukturen annotiert werden. Dafür sind aus der gewählten Sichtrichtung fünf IDBuffer nötig.

7.5.2 Bestimmung der Ankerpunkte

Aus den entstandenen IDBuffers werden mithilfe einer Distanztransformation für jede Struktur mehrere mögliche Ankerpunkte bestimmt. Um zu ermitteln, ob eine Struktur als sichtbar gelten kann und somit mit einer Annotationslinie samt Annotationstext annotiert wird, wird an den Stellen der Ankerpunkte jeweils ein Strahl in die Szene geschickt. Dieser wird von jeder Struktur, die vor der zu annotierenden Struktur liegt, entsprechend deren Opazität abgeschwächt. Übersteigt die Abschwächung einen Schwellenwert, gilt die Struktur an dieser Stelle als nicht sichtbar. Wird ein Ankerpunkt gefunden, an dem die Struktur sichtbar ist, so wird sie normal mit einer Annotationslinie annotiert (Abb. 7.16). Bei einem nicht zu hoch gewählten Schwellenwert führt eine solche Annotation einer eigentlich verdeckten Struktur nicht zur Verwirrung des Betrachters, da dieser die sehr transparenten verdeckenden Strukturen ignorieren, faktisch durch sie hindurch sehen wird. In einigen Fällen wäre eine Annotation von eigentlich sichtbaren Strukturen sonst aus keiner Sichtrichtung möglich – beispielsweise im Falle eines parenchymatösen Organs, wo das oft nur schemenhaft oder sehr transparent dargestellte Parenchym innere Strukturen wie Gefäße und Tumore immer komplett verdeckt (Abb. 7.16).

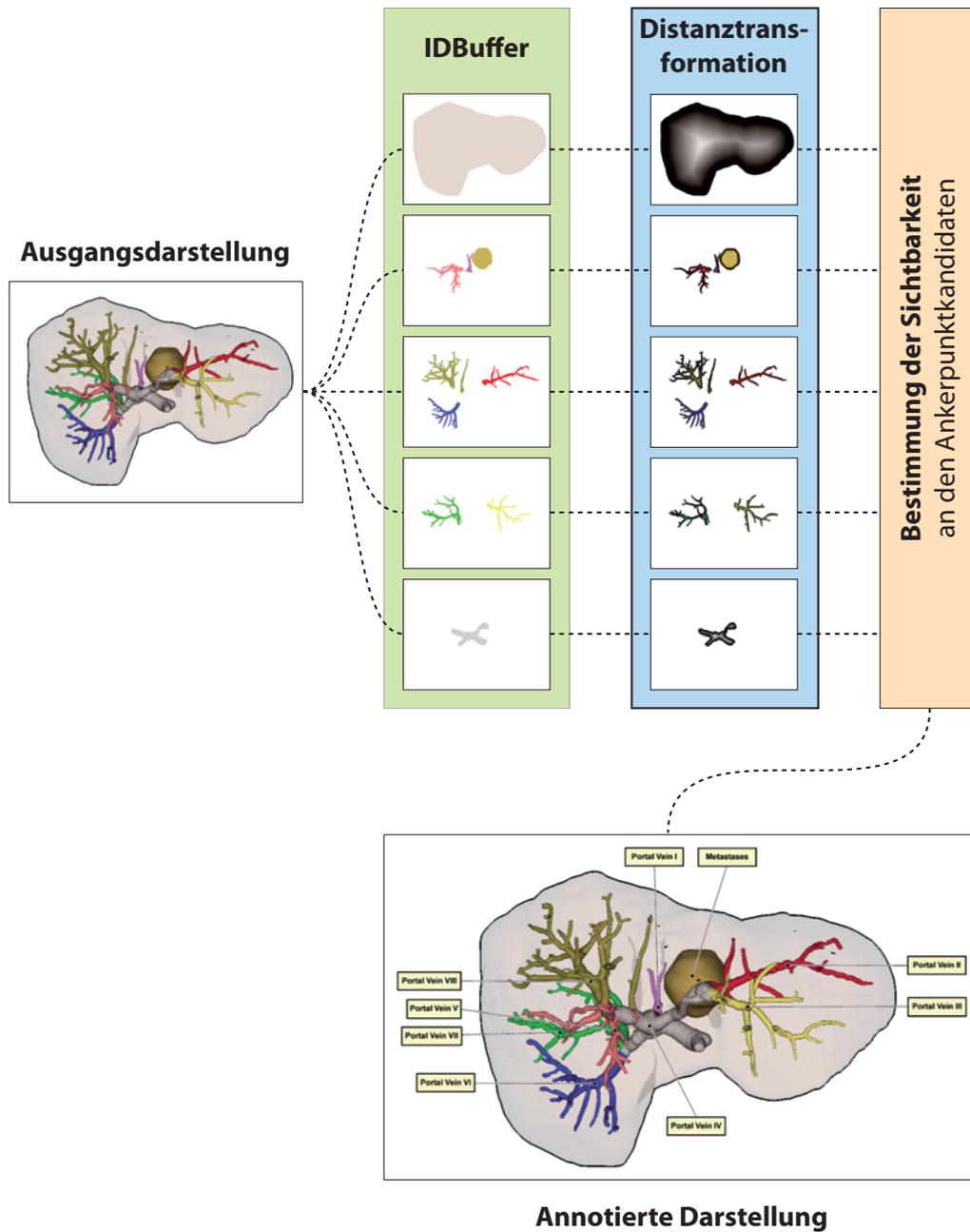


Abbildung 7.15: Einsatz mehrerer IDBuffer zur 3D-Annotation. Für die Ermittlung aller Ankerpunktkandidaten werden für die Darstellung fünf IDBuffer genutzt. Die Strukturen in einem IDBuffer überlappen sich jeweils nicht. Auf den IDBuffern wird für die dargestellten Regionen eine Distanztransformation durchgeführt.

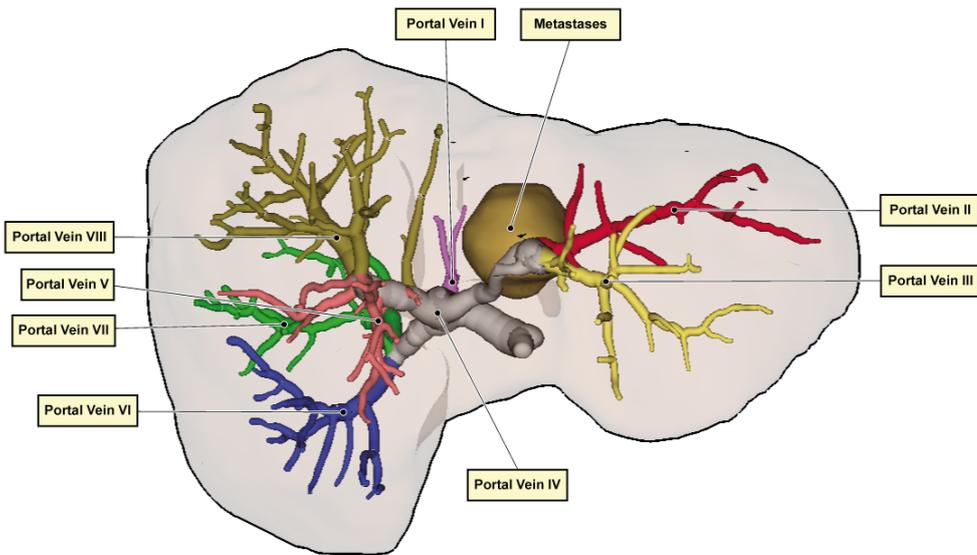


Abbildung 7.16: Annotation mehrerer Strukturen innerhalb der alle anderen Strukturen komplett verdeckenden Leber.

7.5.3 Annotation verdeckter Strukturen

Soll eine nicht sichtbare Struktur annotiert werden (z.B. weil es sich um eine kritische Pathologie handelt), so wird zwar ihr Annotationstext eingblendet, jedoch weist ein Pfeil anstelle der Annotationslinie in Richtung der Struktur (Abb. 7.17). Damit weiß der Nutzer von der Existenz dieser Struktur in der Szene. Durch Anklicken der Annotation wird ein Kameraflug zu einem guten Sichtpunkt³² auf die betreffende Struktur ausgeführt. Durch die Annotation verdeckter Strukturen kann der Nutzer jederzeit auf einen Blick die Zahl und Lage wichtiger Strukturen erfassen. Bei der Planung von Lymphknotenausräumungen des Halses wären dies beispielsweise alle vergrößerten Lymphknoten (Abb. 7.17). Die Gefahr, solche Strukturen zu übersehen, wird somit vermindert.

7.5.4 Verhalten bei starker Vergrößerung

Bei der Exploration von 3D-Darstellungen kann es vorkommen, dass der Nutzer so stark in die Szene hineinzoomt (z.B. um eine Struktur oder einen Abstand besser beurteilen zu können), dass nicht mehr ausreichend Platz vorhanden ist, um die Annotation auf einer umgebenden Hülle um die Darstellung zu platzieren (Abb. 7.18). In solchen Fällen kommt das schon für die 2D-Annotation eingeführte Konzept der Wichtigkeit von Strukturen (siehe Abschnitt 7.4.2) zum Einsatz.

³²Dieser Sichtpunkt wird mit dem Verfahren aus Kapitel 4 berechnet.

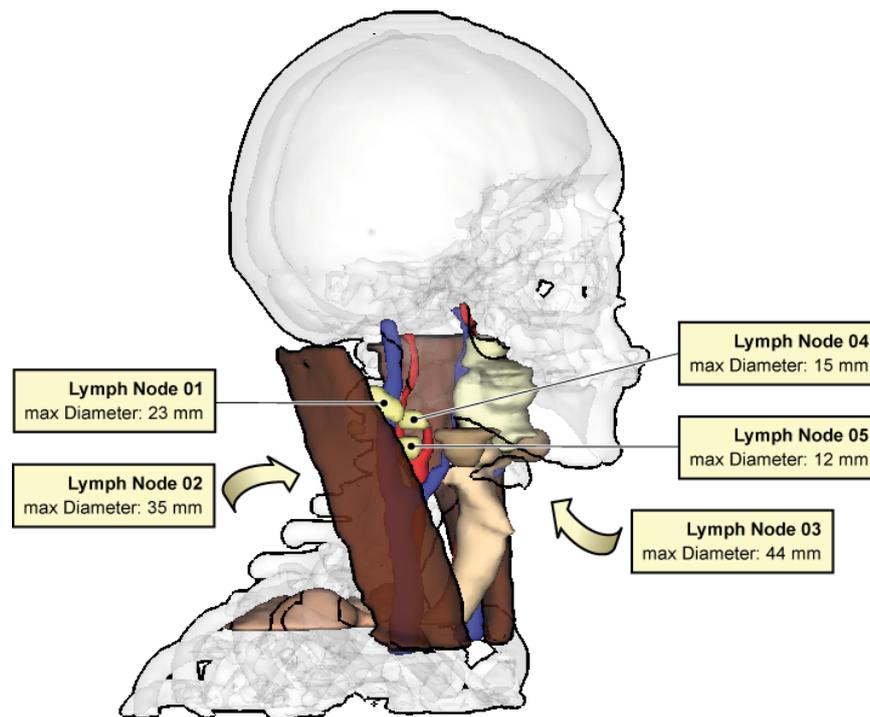


Abbildung 7.17: Annotation mehrerer vollständig verdeckter Strukturen in einer anatomischen Darstellung des Halses. Aufgrund ihrer potentiellen Wichtigkeit sind die Strukturen dennoch annotiert, um auf ihre Existenz und Lage hinzuweisen.

Die Annotationen wichtiger Strukturen werden über weniger wichtigen Strukturen platziert. Dabei werden weniger wichtige Strukturen als Freiraum zur Platzierung der wichtigeren Strukturen betrachtet. Die möglicherweise ebenfalls vorhandenen Annotation der weniger wichtigen Strukturen (im Fall von Abbildung 7.18 die Leber) können in solchen Fällen nicht angezeigt werden.

7.5.5 Gruppierung von 3D-Annotationen

Die Gruppierung von Annotationen einzelner Strukturen ist in 3D-Darstellungen seltener als in 2D-Darstellungen nötig, da hier pro Struktur meist nur eine Annotation platziert wird. Zwar ist die Annotation ein und derselben Struktur mit mehreren Annotationen denkbar, wenn die Struktur in sehr vielen kleinen Teilen verteilt dargestellt wird. Jedoch trat dieser Fall in der Analyse der genutzten Darstellungen sehr selten auf. Eine Gruppierung von Annotationen in 3D-Darstellungen ist sehr wohl gegeben, wenn die Strukturen beispielsweise mit einer übergeordneten Gruppenbezeichnung (z.B. *Lymphknoten* oder *Muskel*) annotiert werden. Dann kann bei mehreren nah beieinander liegenden Strukturen mit der gleichen Annotation eine Gruppenannotation erstellt werden (Abb. 7.19(b)). Die Stile der Gruppierung sind vielfältig (Abb. 7.19(c) und 7.19(d)) und wie schon andere Darstellungsparameter von der persönlichen Präferenz des Nutzers abhängig (siehe Abschnitt 7.6).

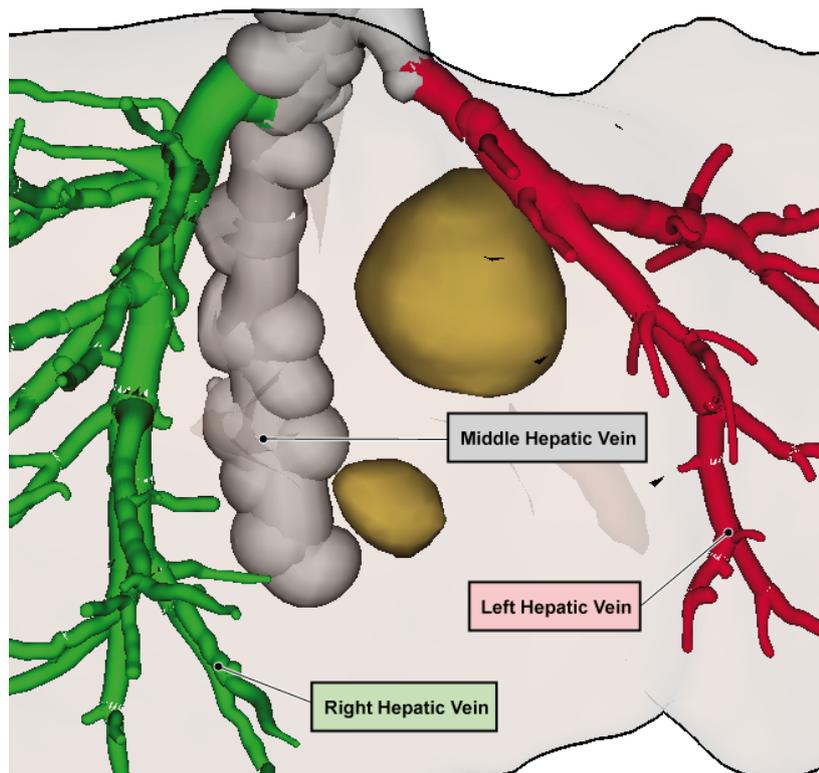


Abbildung 7.18: Annotation bei starken Vergrößerungen. Die Annotation wichtiger Strukturen (hier intrahepatische Gefäße) können über weniger wichtigeren Strukturen (hier der Leber) platziert werden, wenn nicht ausreichend Platz für eine umgebende Hülle vorhanden ist.

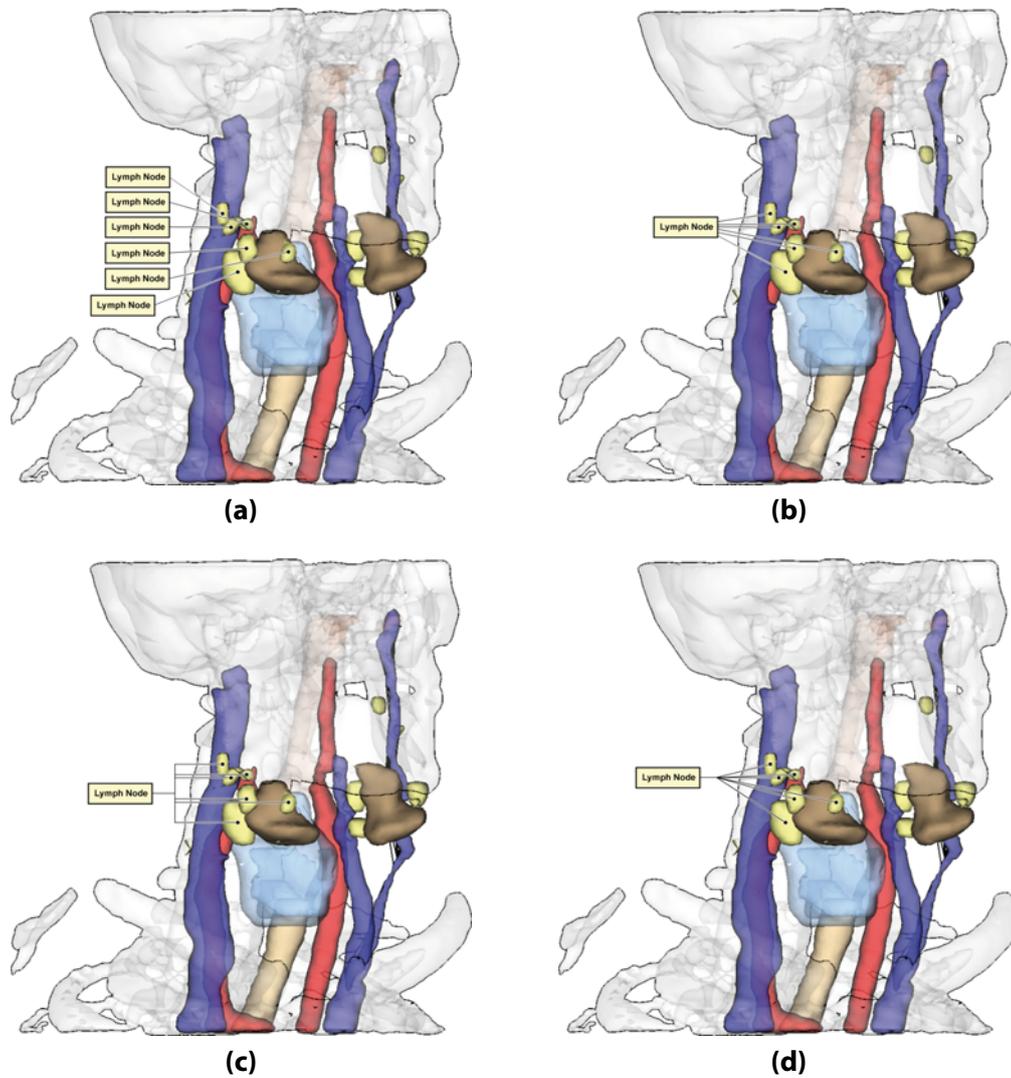


Abbildung 7.19: Gruppierung von 3D-Annotationen. Sind mehrere nah beieinander liegende Strukturen mit der selben Gruppenbezeichnung annotiert **(a)**, können die Annotationen zu einer Gruppenannotation zusammengefasst werden **((b), (c) und (d))**.

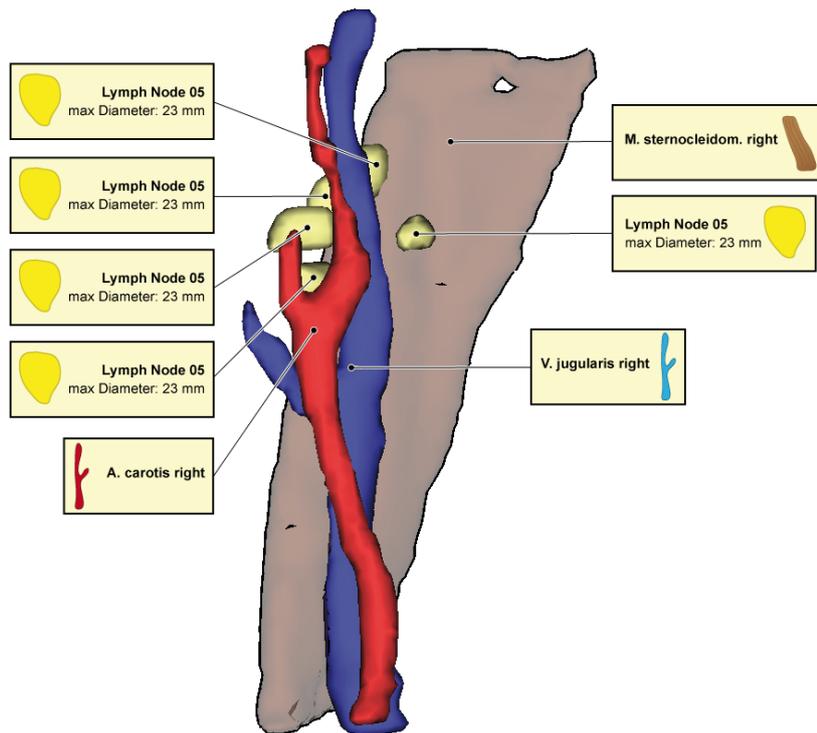


Abbildung 7.20: Annotation mit symbolhaften Darstellungen. Zur Unterstützung der Erfassung einzelner Strukturgruppen ist eine zusätzliche Annotation mit Symbolen der annotierten Strukturen möglich.

7.5.6 Symbole in Annotationen

Da die visuelle Darstellung der Annotationstexte in den vorgestellten Verfahren frei ist, ist auch eine Platzierung von zusätzlichen Symbolen möglich. Symbolhafte Darstellungen der annotierten Struktur können die Erfassung der anatomischen Zugehörigkeiten einer Struktur zusätzlich beschleunigen, indem der Nutzer in einer Darstellung mit mehreren Annotationen schneller die für ihn wichtigen Annotationen ausfindig machen kann (Abb. 7.20).

7.6 Evaluierung

Um die Annotationen, vor allem deren verschiedenen Darstellungsformen, auf ihre Akzeptanz hin zu untersuchen, wurde ein Erkundungsexperiment durchgeführt. Dafür wurden Screenshots von 11 Annotationsdarstellungen in Form eines Fragebogens im Internet präsentiert.³³ Die Nutzer wurden gebeten, zu den einzelnen Bildern eine Bewertung der *Darstellung* und der *Zuordnung* auf einer Skala von 1 (*sehr gut*) bis 5 (*sehr schlecht*) abzugeben. Zusätzlich stand es den Nutzern frei, Kommentare zu jedem Bild abzugeben. Es nahmen 39 Nutzer (9 weiblich,

³³Die vollständige Auflistung aller gezeigten Bilder ist Anhang A zu entnehmen.

30 männlich) an der Umfrage teil. Von ihnen bescheinigten sich 22 ein gutes oder sehr gutes medizinisches Wissen in Verbindung mit einem zumeist medizinischen Beruf, wie Arzt oder MTRA. Die zusammengefassten Durchschnittswerte für jede Darstellungsart sind Tabelle 7.1 zu entnehmen. Die genauen Werte für jedes Bild sowie einige abgegebene Kommentare sind in Anhang A zu finden.

Art der Annotation	Darstellung	Zuordnung
Annotationen ohne Boxen	2,6 ($\sigma = 0,84$)	2,5 ($\sigma = 0,84$)
Annotationen mit einfarbigen Boxen	2,0 ($\sigma = 0,63$)	2,0 ($\sigma = 0,63$)
Annotationen in der Farbe der Struktur	2,1 ($\sigma = 1,12$)	1,9 ($\sigma = 0,89$)
Annotationen mit Symbolen	2,4 ($\sigma = 1,16$)	1,9 ($\sigma = 0,98$)
Ungruppierte Annotationen	3,6 ($\sigma = 0,81$)	2,8 ($\sigma = 0,96$)
Gruppierte Annotationen ^a	2,1 ($\sigma = 0,59$)	2,0 ($\sigma = 0,63$)

Tabelle 7.1: Ergebnisse der Evaluierung der Annotationen mit 39 Nutzern. 1=*sehr gut*; 5=*sehr schlecht*. Die Ergebnisse für die 2D-Darstellungen sind vergleichbar.

^aim am meisten bevorzugten Stil

Es zeigte sich eine generell positive Bewertung aller Annotationsstile, wobei die ungruppierten Annotationen bei der Annotation von Strukturgruppen am schlechtesten abschnitten und die Annotationen mit einfarbig hinterlegten Boxen am besten. Hervorzuheben ist, dass im Gegensatz zu den numerischen Bewertungen die textuellen Kommentare sehr inhomogen waren. So wurden Annotationsstile wie die Symboldarstellung von einigen Befragten als äußerst sinnvoll, von anderen dagegen sehr ablehnend beurteilt. Dass dennoch eine zumeist sehr gute Bewertung der Annotationen abgegeben wurde, ist eventuell damit zu erklären, dass keinerlei Vergleichsdarstellungen in der alltäglichen Routine existieren und so jede Art von Annotation als eine Verbesserung empfunden und dem entsprechend positiv bewertet wird. Aus der Inhomogenität der Kommentare lässt sich die Nutzung von verschiedenen Voreinstellungen (Presets) ableiten. Wenige (zwei bis vier) gut ausgewählte Presets können dem Nutzer in späteren Applikationen zur Auswahl des von ihm präferierten Annotationsstils angeboten werden.

Das Erkundungsexperiment lässt keine Rückschlüsse auf die Wertigkeit der einzelnen Annotationsstile hinsichtlich ihrer Unterstützung der Wahrnehmung, d.h. der objektiven Verbesserung der Zuordnungsgeschwindigkeit, zu. Dennoch zeigt es durch die Abfrage persönlicher Präferenzen und Einschätzungen, dass ungeachtet objektiver Maße die Akzeptanz eines Annotationsstils entscheidend für seinen späteren Einsatz sein kann.

7.7 Zusammenfassung und Ausblick

In diesem Kapitel wurden neue Verfahren zur Annotation von 2D-Schichtbildern und 3D-Darstellungen segmentierter Strukturen vorgestellt. Die Verfahren berücksichtigen dabei die besonderen Eigenschaften medizinischer Darstellungen, beispielsweise die Schichtbildkohärenz. Für die 3D-Darstellung wurde eine neue Technik eingeführt, die die Annotation von verdeckten Strukturen ermöglicht. Im Zusammenspiel mit Kamerafahrten zu guten Sichtpunkten auf solche Strukturen erhält der planende Arzt einen tieferen Ein- und Überblick und kann alle wichtigen Strukturen in ihrer Lage auf einen Blick erfassen.

Generell ermöglicht das Annotationsframework eine individuelle Beschriftung jeder einzelnen Struktur – auch hinsichtlich des Stils. Für den alltäglichen Einsatz, sei es in Lernsystemen oder chirurgischen Planungssystemen, haben sich allerdings wohl definierte Voreinstellungen als sehr hilfreich erwiesen. Im Lernsystem LIVERSURGERYTRAINER werden beispielsweise alle sichtbaren Gefäße mit ihrem Namen annotiert und sichtbare Lebersegmente zusätzlich mit ihrem Volumen, während das Lebergewebe selbst nicht annotiert wird (Abb. 7.4 und 7.16). Zur Inspektion von Lymphknoten im NECKSURGERYPLANNER werden in der Standardeinstellung alle Lymphknoten unabhängig von ihrer Sichtbarkeit mit ihrer maximalen Ausdehnung annotiert (Abb. 7.17).

Die durchgeführte Evaluierung kann nur ein erster Schritt hin zu einer umfassenderen Evaluierung sein, die auch objektive Parameter wie die Effizienz der Erfassung der Annotationen in verschiedenen Stilen berücksichtigt. GÖTZELMANN u. a. [2007a] geben dafür ein gutes Beispiel, indem sie den Nutzer in animierten Annotationsdarstellungen mit abstrakten Texten bestimmte Annotationen suchen lassen und die dafür benötigte Zeit messen. Die Annotationstexte bestehen aus vier Buchstaben und unterscheiden sich jeweils in nur einem Buchstaben, das heißt, sie haben eine Levenshtein-Distanz von 1 (LEVENSHEIN [1966]).

Für die medizinische Visualisierung ist die Annotation von direkten Volumendarstellungen eine weitere Herausforderung. Dabei ist sowohl die Ermittlung von Ankerpunkten wie auch die möglichst überlappungsfreie Platzierung der Annotationstexte wichtig. Die Zuordnung jedes Voxels zu einer oder mehreren Strukturen oder Pathologien ist zwingend, um später Annotationen vornehmen zu können. Wenn keine Vorsegmentierung vorliegt, muss auf automatische Verfahren zurückgegriffen werden. GLASSER u. a. [2010] haben beispielsweise ein Verfahren entwickelt, um Plaque-Ablagerungen in Herzkranzgefäßen automatisch mittels lokaler Transferfunktionen hervorzuheben. Das Wissen um die Lage und Ausdehnung solcher Plaques kann bei einem Verfahren zur Annotation als Ausgangspunkt für die Platzierung von Hinweisen auf diese Plaques dienen. Die manuell eingefügten Annotationen in Abbildung 7.21 geben einen ersten Eindruck eines möglichen Ergebnisses.

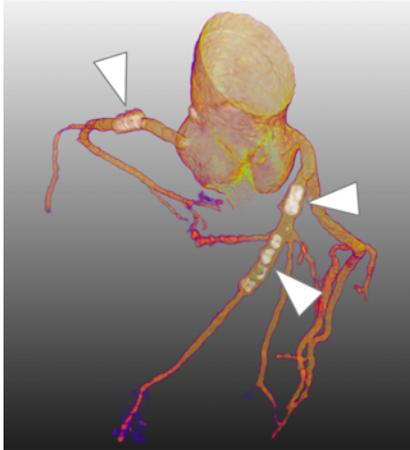


Abbildung 7.21: Annotationen in einer Darstellung der Herzkranzgefäße von [Glaßer u. a. \[2010\]](#). Die hell dargestellten Plaques könnten mit einem zukünftigen Verfahren automatisch annotiert werden (Bild mit freundlicher Genehmigung von Sylvia Glaßer).

[WOHLFART & HAUSER \[2007\]](#) geben ebenfalls einen Eindruck, wie Annotationen in einer direkten Volumendarstellung aussehen könnten. Allerdings müssen hier die Ankerpunkte in einem Authoring-System manuell gesetzt und im dreidimensionalen Raum verankert werden. Dies stellt im Kontext der Erstellung weniger Animationen zu Ausbildungszwecken, wie [WOHLFART & HAUSER \[2007\]](#) es vorschlagen, einen akzeptablen Aufwand dar, ist aber in der klinischen Routine nur vereinzelt denkbar – beispielsweise, wenn ein Radiologe wichtige Punkte in einer 3D-Darstellung für den planenden Chirurgen markiert.

In der Interaktion mit den Annotationen sind Ergänzungen denkbar. So könnte der Nutzer die Position von Annotationen selbst beeinflussen, indem er z.B. mehrere Annotationen zusammenfasst und zur Seite schiebt. Ebenfalls sind Techniken zum individuellen Ein- und Ausschalten einzelner Annotationen vorstellbar (derzeit lassen sich nur Annotationen ganzer Strukturgruppen ein- und ausschalten). Auch sind noch nicht alle Möglichkeiten der Anpassung des Erscheinungsbildes der Annotationen ausgeschöpft – ausgefeilte Ausrichtungen der Annotationen zueinander können zur besseren visuellen Ausgewogenheit einer Darstellung beitragen. Mit solchen Ansätzen kann zwar dem individuellen ästhetischen Anspruch eines Nutzers Rechnung getragen werden. Jedoch sollten zusätzliche Interaktionsformen immer gut durchdacht und begründet sein, um eine Explorationssituation nicht mit zu vielen Möglichkeiten für den Nutzer zu überfrachten.

8

Generierung wiederverwendbarer Visualisierungen und Animationen

Zu einer effizienten Nutzung von komplexen Visualisierungstechniken in der alltäglichen Routine der Operationsplanung ist ein Konzept zur Wiederverwendung einmal erstellter Visualisierungen und Animationen unerlässlich. Da sich der Ansatz von [MÜHLER u. a. \[2006a\]](#) (Kapitel 3), dem Arzt bzw. Chirurgen eine skriptbasierte Schnittstelle als wenig praktikabel herausgestellt hat, sind wesentlich einfachere Konzepte nötig. Ein solches Konzept ist das der *Keystates*, welches versucht, in existierenden Visualisierungen Aspekte und Komponenten ausfindig zu machen, die sich auch auf andere, noch unbekannte Daten anwenden lassen. Das Konzept der Keystates erlaubt so eine intuitive Wiederverwendung vieler Visualisierungsparameter vom Erscheinungsbild einzelner Strukturen bis hin zu komplexen Sichten auf ganze Szenen.

Gliederung des Kapitels

In Abschnitt 8.1 wird der Bedarf eines neuen Verfahrens kurz motiviert. Verwandte Arbeiten werden in Abschnitt 8.2 diskutiert. In Abschnitt 8.3 wird das Konzept der Keystates ausführlich vorgestellt, bevor in Abschnitt 8.4 Anwendungsszenarien für Keystates präsentiert werden. Wie die Erstellung von Animationen mithilfe von Keystates erleichtert werden kann, wird in Abschnitt 8.5 erläutert. Das Kapitel wird mit zwei Beispielanwendungen (Abschnitt 8.6) sowie einer Diskussion und Zusammenfassung (Abschnitt 8.7) abgeschlossen.

Eigene Publikationen

Dieses Kapitel basiert u.a. auf folgender eigenen Publikation:

MÜHLER & PREIM [2010b] Mühler, K. & Preim, B. *Reusable Visualizations and Animations for Surgery Planning*. In: **Computer Graphics Forum**, 2010

8.1 Motivation

Die Erstellung einer aussagekräftigen Visualisierung durch den planenden Arzt ist immer mit Aufwand verbunden. Strukturen müssen ein- bzw. ausgeblendet, Farben und Transparenzen zugewiesen und ein geeigneter Sichtpunkt auf die Szene gefunden werden. Selbst wenn der Arzt dabei durch Techniken, wie die automatische Sichtpunktbestimmung aus Kapitel 4, unterstützt wird, ist dieser Prozess mitunter energieverbrauchend – gerade wenn es um immer wiederkehrende Visualisierungen geht. Die Planung einer onkologischen Tumorresektion der Leber basiert zwar jedes Mal auf den *individuellen* Bilddaten und Befundes des Patienten. Dennoch haben *alle* Fälle gemein, dass eine oder mehrere Metastasen oder Tumore mit einem Sicherheitsrand aus der Leber reseziert werden müssen. Die dafür generierten Visualisierungen weisen daher ein großes Maß an Gemeinsamkeiten auf (Abb. 8.1). Das Ermitteln solcher Gemeinsamkeiten und deren Anwendung bei zukünftigen Fällen können den Explorationsprozess des Arztes enorm beschleunigen. Ein ähnliches und damit vergleichbares Erscheinungsbild möchte ein Chirurg auch bei der Nutzung mehrerer Datensätze desselben Patienten erreichen. Zum (visuellen) Vergleich von prä- und post-operativen Visualisierungen, z.B. zur Follow-Up-Kontrolle, sind eine möglichst identische Visualisierung sowie ähnliche Sichtpunkte sehr förderlich. Da sich die Datensätze jedoch zumeist in Form, Lage und Anzahl der vorhandenen Strukturen unterscheiden (z.B. wenn Metastasen reseziert oder nicht alle Strukturen in allen Datensätzen gleichermaßen segmentiert wurden), muss eine Visualisierung adaptiv angepasst werden.

Neben der Unterstützung der Exploration kann auch die Erstellung von Animationen und deren Präsentation von einer Wiederverwendung einmal getroffener Visualisierungsentscheidungen profitieren. Allerdings muss der Definitionsprozess solcher Animationen intuitiv sein und sich lückenlos in den Explorations- und Planungsprozess integrieren.

Für beide Anwendungsgebiete – die Wiederverwendung von einzelnen Visualisierungen und die Erstellung von Animationen – erscheint somit ein Ansatz angemessen, der rein visuell auf der Basis der individuell durchgeführten Exploration basiert und kaum zusätzlichen Aufwand (wie z.B. das Lernen einer Skriptsprache) benötigt. Es müsste eine Lösung gefunden werden, die dem Arzt „von den Augen abliest“ (respektive: aus seinen Explorationen liest), welche Strukturen und Gebiete er warum wie visualisiert und welche Kameraeinstellungen er dazu wünscht.

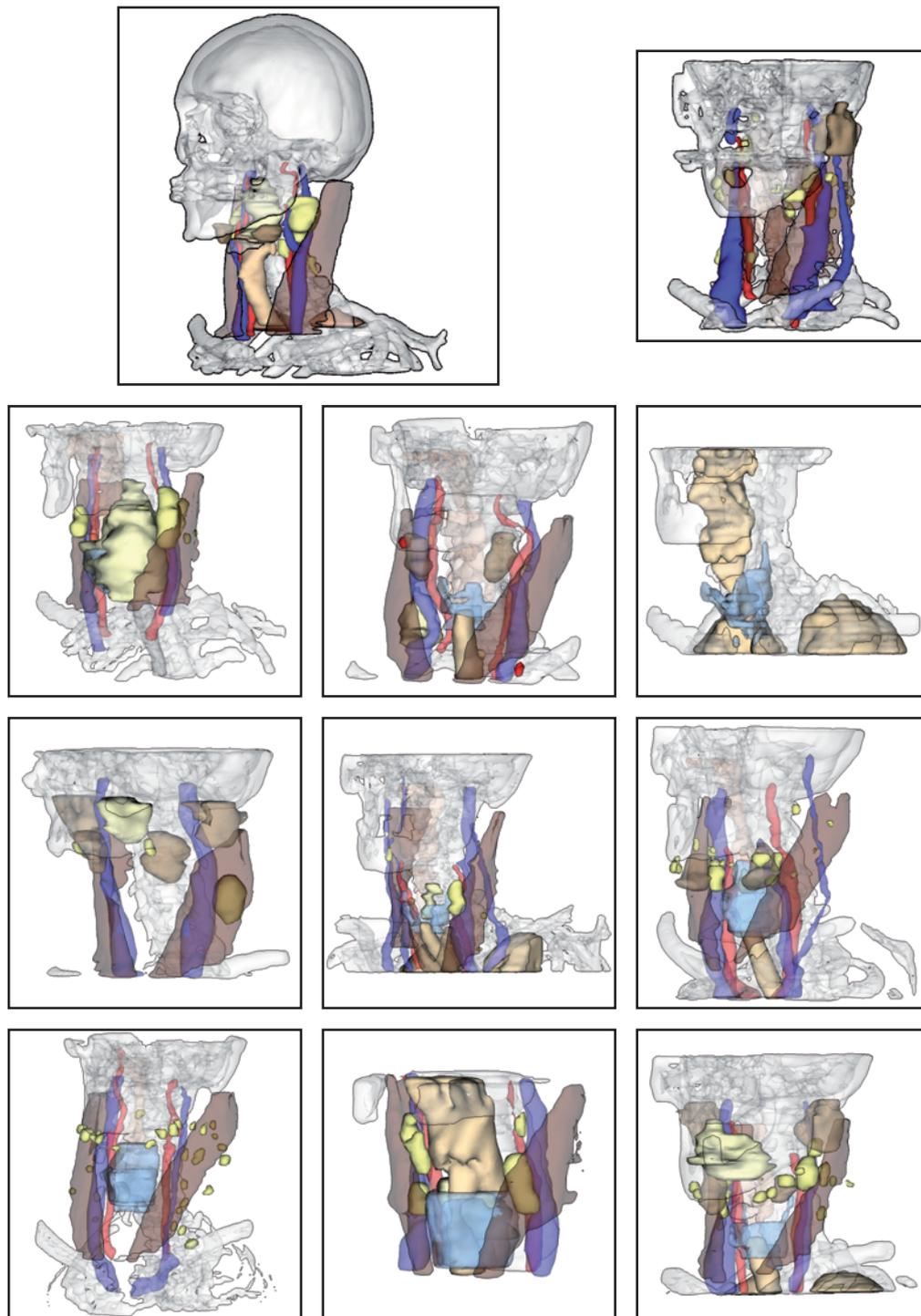


Abbildung 8.1: Visualisierungen verschiedener Patientendatensätze zur HNO-Operationsplanung. Die Visualisierungen sind in ihrem Stil und den sichtbaren Strukturen gesamtheitlich der oberen linken Visualisierung ähnlich. Die automatische Erzeugung solcher Visualisierungen kann den Operationsplanungsprozess enorm beschleunigen.

8.2 Verwandte Arbeiten

Im Folgenden werden Arbeiten analysiert, die sich mit der effizienten Erstellung von Visualisierungen und deren Wiederverwendung sowie dem Animations-Authoring beschäftigen.

8.2.1 Effiziente Erstellung von Visualisierungen

In der Vergangenheit haben sich verschiedene Gruppen mit der effizienten Erstellung von Visualisierungen zur Unterstützung von Explorationen beschäftigt. [MA \[1999\]](#) präsentierte ein Framework, die *Image Graphs*, welches die Veränderungen visualisierte, die in einer Darstellung während des Explorationsprozesses vorgenommen wurden. Die Knoten des Graphen stellen Screenshots der Visualisierungen dar, welche durch Kanten verbunden sind. Jede Kante repräsentiert dabei die Veränderung eines einzelnen Parameters, wie Rotation oder Farbe. Benachbarte Knoten unterscheiden sich in einem Parameter. *Image Graphs* können gemeinschaftlich genutzt werden. Animationen können durch die Selektion einzelner Screenshots als Keyframes generiert werden.

[JANKUN-KELLY u. a. \[2007\]](#) definierten ein Modell zur formellen Beschreibung der Exploration einer Visualisierung. In diesem Modell werden die Veränderungen an Parametern gesondert gespeichert, um Informationen zwischen verschiedenen Visualisierungsapplikationen auszutauschen oder Explorationsergebnisse mit anderen Anwendern gemeinsam zu nutzen. Die einzelnen Schritte einer Exploration werden ähnlich den *Image Graphs* von [MA \[1999\]](#) dargestellt und erlauben die Rekonstruktion eines früheren Visualisierungsstatus.

[MARKS u. a. \[1997\]](#) führten die *Design Galleries* ein, bei denen eine Menge an zufällig generierten Visualisierungsparametern, wie die Beleuchtung, genutzt wird, um eine Menge von Visualisierungen und Animationsübergängen zu generieren. Diese Menge wird dem Nutzer in der Hoffnung präsentiert, dass die gewünschte Darstellung darunter ist. Zur Erzeugung einer ausreichend großen Menge an Visualisierungen ist eine erhebliche Rechenzeit erforderlich.

[VAN WIJK & NUJ \[2003\]](#) stellten einen neuen Ansatz vor, mit dem die Nutzer aus einer Reihe von Presets in hochdimensionalen Parameterräumen neue Einstellungen generieren können. Dazu werden verschiedene Presets gewichtet summiert. Zur Einstellung der Gewichte kommt ein neues grafisches Steuerelement, der *Preset Controller*, zum Einsatz. Da Schieberegler für die verschiedenen Gewichte als wenig intuitiv und einladend zum Experimentieren empfunden wurden, wurde dem Nutzer eine zweidimensionale Darstellung präsentiert, in der er die vorhandenen Presets sowie seine neue Parameterwahl als Punkte frei platzieren konnte. Die Gewichte der einzelnen Presets waren dabei entsprechend der Metapher „*Nähe bedeutet starken Einfluss*“ indirekt proportional zur Entfernung der neuen Parameterwahl zu den einzelnen Presets. Als Anwendungen wurden das Morphing

von Illustrationen sowie die Musiksynthese aus verschiedenen Instrumenten präsentiert.

[KOOP u. a. \[2008\]](#) haben für VISTRAILS (siehe Abschnitt 9.2.1) einen Algorithmus mit dem Namen *VisComplete* entwickelt, der dem Nutzer beim Aufbau einer Visualisierungspipeline in der netzwerkbasierter Programmierung verschiedene Vorschläge unterbreitet, wie der Nutzer die Pipeline vervollständigen könnte. Die Vorschläge basieren auf der Analyse bisheriger Netzwerke mit vergleichbaren Elementen und Parametrisierungen. *VisComplete* zielt auf Forscher ab, die über umfangreiche Programmierkenntnisse verfügen und einzelne, visuell ansprechende Visualisierungen erzeugen möchten. Um die Erzeugung einer Visualisierung reproduzierbar zu machen, schlugen [SCHEIDEGGER u. a. \[2007\]](#) einen Verlauf der Netzwerkentwicklung in VISTRAILS vor, der in einem Versions-Baum gespeichert würde.

Zur Erstellung von Fokus+Kontext-Darstellungen in medizinischen Volumenerendings haben [RAUTEK u. a. \[2008\]](#) einen Ansatz vorgestellt, bei dem je nach Entfernung einer Region zum Mauszeiger zu einer Clipping-Ebene oder in Abhängigkeit von der Sichtrichtung andere Transferfunktionen genutzt werden, um das Volumen zu rendern. Die Farb- und Transparenzeigenschaften einer jeden Transferfunktionen für jeden dieser Bereiche werden als Kugeln dargestellt, was eine intuitive Auswahl aus einer Reihe von voreingestellten Transferfunktionen ermöglicht.

[REZK-SALAMA u. a. \[2006\]](#) haben die aufwendige Parametrisierung von Transferfunktionen für Volumenerendringdarstellung in der Art vereinfacht, dass nur noch ein Parameter in Form eines Schieberegler zur Anpassung der aktuellen Transferfunktion an ein Template verändert werden muss. Eine genau Kenntnis oder gar ein Verständnis der zugrunde liegenden Transferfunktion ist nicht mehr erforderlich. Sie haben dazu am Beispiel einer CTA 20 Datensätze analysiert und Templates für verschiedene Transferfunktionen erstellt.

8.2.2 Wiederverwendung von Visualisierungen

Die Wiederverwendung von Visualisierungen wird nur von wenigen Arbeiten thematisiert. So analysierten [HAMEL & STROTHOTTE \[1999\]](#) nicht-photorealistic Illustrationen auf die verwendeten Stile hin, um diese in neuen Visualisierungen wieder zu verwenden. Parameter, wie die Liniendicke oder -art, konnten so auf andere Visualisierungen angewendet werden, um ein vergleichbares visuelles Erscheinungsbild zu erhalten.

[SVAKHINE u. a. \[2005\]](#) boten verschiedene vordefinierte Illustrationsstile an, um so effizient medizinische Volumenillustrationen zu erzeugen. Die Stile enthielten verschiedene Effekte für Bereiche von Interesse, z.B. um verdeckendes Material zu entfernen oder um Kanten in einem besonderen Stil darzustellen. Es wurden

verschiedene Interfaces für Autoren und Nutzer angeboten, um die Stile zu definieren.

[GROTH & STREEFKERK \[2006\]](#) stellten ein System vor, um die Herkunft (Provenienz) einer Visualisierung speichern zu können. Sie wollten die Schritte auf dem Weg zu einer fertigen Visualisierung nachvollziehbar machen, um so die Erkenntnistiefe im Bereich der molekularen Visualisierungen zu erhöhen. Jede Aktion (z.B. Zoomen oder Rotationen) werden in einem Verlaufs-Graphen gespeichert und können durch den Nutzer manuell annotiert und um weitere Informationen angereichert werden. Der Ansatz erlaubt keinerlei Wiederverwendung oder Adaption für weitere Datensätze, gestattet aber die Reproduktion der Exploration eines Datensatzes.

Eine Methode zur Spezifikation eines Layouts von textuellen Annotationen durch die Vorgabe eines Beispiellayouts wurde von [VOLLICK u. a. \[2007\]](#) vorgestellt. Sie nutzten dafür Annotationslayouts, die manuell durch einen Illustrator erstellt wurden und versuchten diese Layouts auf neue Bilder anzuwenden, um so eine vergleichbare Anordnung der Annotationen zu erreichen. Ein Nachteil des Systems ist seine Langsamkeit – es erforderte vier Minuten, um ein Layout aus 20 Annotationen auf eine neue Darstellung anzuwenden.

Eine frühe Methode der Wiederverwendung aus dem Bereich der Makroerstellung haben [KURLANDER & FEINER \[1992\]](#) vorgestellt. Bei der makrobasierten Erstellung von einfachen Schemazeichnungen boten sie dem Nutzer eine automatisch generierte Historie an. Durch die Auswahl eines vergangenen Zustandes konnten die Parameter dieser Objekte auf neu erzeugte Objekte übernommen und so wiederverwendet werden.

8.2.3 Animations-Authoring

Es sind einige Ansätze bekannt, die den Prozess der Animations-Erstellung (des „Authoring“) beschleunigen und effizienter machen wollen. Dabei ist offensichtlich, dass klassische Animations-Tools, wie Maya [®][AUTODESK \[2009\]](#) oder Blender [THE BLENDER FOUNDATION \[2009\]](#), auf professionelle Nutzer abzielen, die einzelne, komplexe Animationen mit einem großen Aufwand erstellen. Solche Tools sind grundlegend ungeeignet für eine (automatische) Massenproduktion von Animationen oder für unerfahrene Nutzer, wie Chirurgen.

Ein Ansatz, bei dem existierende Animationen kombiniert und auf neue Szenen angewendet werden, wurde von [WANG & LEE \[2008\]](#) vorgestellt. Sie extrahierten einzelne Teile verschiedener Oberflächenanimationen und versuchten, dafür kompatible Mengen von Dreiecken in einer neuen Szene zu finden, um sie auf gleiche Art und Weise zu animieren.

Weitere Systeme zum Authoring medizinischer Animationen wurden von [WOHLFART & HAUSER \[2007\]](#) sowie [ISERHARDT-BAUER u. a. \[2002\]](#) präsentiert. Diese Arbeiten wurden ausführlich in Abschnitt [3.2](#) vorgestellt.

Nahezu alle präsentierten Ansätze zielen auf eine sorgfältige Exploration einzelner Datensätze oder die Erstellung einzelner, besonders hervorstechender Animationen und Visualisierungen ab. Die Zielgruppen sind zumeist Wissenschaftler oder erfahrene Computerexperten. In Kontrast dazu zielt der im Folgenden vorgestellte Ansatz der Keystates auf Nutzer ohne Programmierkenntnisse oder Visualisierungs-Hintergrund, vornehmlich Chirurgen, ab.

8.3 Konzept der Keystates

Um die Wiederverwendung von Visualisierungen sowie das einfache Authoring von Animationen durch den Arzt selbst zu ermöglichen, wurden im Rahmen dieser Arbeit die **Keystates** entwickelt. Ein Keystate ist eine abstrakte Beschreibung einer Visualisierung (egal ob 2D oder 3D) und enthält alle Informationen, die für eine Wiederverwendung bzw. Anwendung auf einen anderen Datensatz nötig sind. Mehrere Keystates können als Abfolge von Visualisierungen gleichsam den aus der Animationstechnik bekannten Keyframes zur Beschreibung kompletter Animationen genutzt werden. Das Ziel ist, einen Keystate auf einen neuen, unbekanntem Datensatz anzuwenden, um im Ergebnis eine Visualisierung zu erhalten, die den gleichen Informationsgehalt wie die ursprüngliche Visualisierung bietet. In den folgenden Abschnitten wird beschrieben, welche Informationen in einem Keystate gespeichert werden (Abschnitt 8.3.1), wie diese Informationen erfasst werden (Abschnitt 8.3.2) und wie ein Keystate auf neue Datensätze angewendet wird (Abschnitt 8.3.3).

8.3.1 Informationen, die in einem Keystate gespeichert werden

Ein Keystate muss alle Informationen enthalten, die notwendig sind, um ähnliche Visualisierungen zu einer gegebenen Ausgangsvisualisierung zu erzeugen. Diese Informationen umfassen unter anderem segmentierte und künstliche Strukturen, wie Nadeln:

1. Informationen über die Sichtbarkeit jeder Struktur.
Dies schließt Strukturen ein, die im Datensatz enthalten sind, aber momentan nicht dargestellt werden.
2. Informationen über die grafischen Stile jeder Struktur. Dies sind: Farbe, Transparenz, Silhouettenbreite und -farbe.
3. Informationen über die momentane Struktur, die für den Arzt von Interesse ist.
Welche Struktur liegt im Fokus des Arztes?
4. Informationen über den Sichtpunkt bzw. das momentane Schichtbild.
Wo befindet sich die Kamera in der 3D-Szene und worauf ist sie ausgerichtet? Welches Schichtbild wird gerade angezeigt?

Besonders der dritte Aspekt erfordert Zusatzwissen darüber, was der Arzt in der aktuellen Visualisierung zu sehen wünscht. Der Fokus kann dabei auf der Beurteilung der Morphologie einer einzelnen Struktur, einer Strukturgruppe oder auf der räumlichen Beziehung mehrerer Strukturen (z.B. einem Abstand) liegen. Das Wissen über den aktuellen Fokus des Arztes ist eine wichtige Voraussetzung zur Interpretation des momentanen Sichtpunktes in einer 3D-Szene in der Art, die eine Generierung ähnlicher Sichtpunkte auf neue 3D-Szenen ermöglicht. Für 2D-Schichtbilddaten ist die Struktur von Interesse wichtig, um ähnliche Strukturen in neuen Datensätzen in vergleichbaren Schichten präsentieren zu können.

Daher wird ein Keystate für eine 3D-Szene als Tripel K_{3D} mit

$$K_{3D} = (S, V, I) \tag{8.1}$$

definiert, wobei S eine Menge an Stilparametern für jede Struktur oder Strukturgruppe ist, V ist Information über den Sichtpunkt und I die Struktur von Interesse in Kombination mit einem Interessentyp. Für 2D-Schichtbilddarstellungen ist ein Keystate ein 3-Tupel K_{2D} mit

$$K_{2D} = (S, C, I) \tag{8.2}$$

wobei S und I die gleichen Informationen wie bei K_{3D} repräsentieren und C die Kontextinformation beinhaltet, die im aktuellen Schichtbild sichtbar ist. Der genaue Aufbau von S , V , I und C wird im folgenden Abschnitt beschrieben.

8.3.2 Erfassung der Informationen

Alle Informationen, die ein Keystate kapselt, sollten automatisch aus der gegebenen aktuellen Visualisierung gewonnen werden, um dem Anspruch der minimalen und intuitiven Interaktion gerecht zu werden. Es wird aufgezeigt werden, dass dies nicht in jedem Fall möglich sein wird.

Die **Informationen für die Sichtbarkeit, Farbe, Transparenz** usw. jeder Struktur kann einfach aus der Visualisierung entnommen werden. Dennoch müssen die Informationen generalisiert werden, um eine Wiederverwendung bei ähnlichen, aber nicht völlig gleichen Datensätzen zu ermöglichen.³⁴ Es ist offensichtlich, dass „*Lymphknoten 27 ist sichtbar und opak*“ nicht zu verwenden ist, wenn ein Datensatz bspw. nur 15 Lymphknoten enthält. Daher werden Aussagen wie „*Alle Lymphknoten sind sichtbar und opak*“ angestrebt.

Das hier vorgestellte Verfahren analysiert deshalb die Visualisierung auf verschiedene Strukturgruppen hin. Einige Gruppen sind implizit in den Falldaten gegeben, so z.B. anatomische Zugehörigkeiten wie *Knochen*, *Gefäße* oder *Muskeln*.

³⁴Unterschiede können allein schon durch den Umfang der durchgeführten Segmentierungen entstehen.

Andere Gruppen werden automatisch aus dem Wissen um die Applikation und deren Anforderungen ermittelt: Dies sind zum Beispiel Gruppen wie „*Strukturen auf der linken/rechten Seite des Halses.*“ oder „*Lymphknoten größer als 2cm.*“. Im Ergebnis erhält man für jede ermittelte Gruppe Informationen über gemeinsam genutzte Stilparameter. Kann kein gemeinsamer Stil für alle Strukturen einer Gruppe festgestellt werden, z.B. sind einige Lymphknoten gelb, während andere rot sind, so wird diese Gruppe nicht im Keystate erfasst. Die Informationen zu den Visualisierungsstilen wird in S als eine Menge von Einzelstilen s mit

$$s = (\text{groupname}, \text{visibility}, \text{color}, \text{transparency}, \text{silhouette}) \quad (8.3)$$

gespeichert.

Zur Ermittlung der **Informationen über den Sichtpunkt** in 3D-Szenen genauso wie für die **Informationen über die aktuelle Schicht** in der 2D-Darstellung ist es essenziell, das Visualisierungsziel des Arztes zu kennen. Dies kann z.B. die Begutachtung einer Infiltration einer anatomischen Struktur durch einen Tumor sein. Da möglichst keine Nutzerinteraktion gewünscht bzw. in einigen Fällen gar nicht möglich ist (siehe Abschnitt 8.4.1), muss diese Information automatisch ermittelt werden. In Fällen, in denen der Arzt die Struktur von Interesse aus einer Liste oder in der Szene direkt ausgewählt hat, wird diese Struktur als Fokus angenommen. Für 3D-Szenen, in denen die Kamera manuell positioniert wurde, wird nach der Struktur gesucht, die

- a) den geringsten Abstand zum Bildmittelpunkt hat,
- b) die möglichst unverdeckt von anderen Strukturen ist und
- c) die aufgrund der zugrunde liegenden Daten als wichtig eingeschätzt wird.³⁵

Für jede Struktur i wird ein Interessenwert SOI (Structure of Interest) berechnet:

$$SOI_i = (IC_i + A_i) * IMP_i \quad (8.4)$$

Dabei ist A die sichtbare Fläche der Struktur im Verhältnis zur Fläche des Bildraumes und IMP ist die Wichtigkeit der Struktur. IC (für *image center*) ist der Kehrwert der mittleren Distanz aller Pixel der Struktur zur Mitte des Bildraumes. Der Wert von IC ist nahe 1, wenn die Struktur nur durch wenige Pixel in der Mitte des Bildraumes dargestellt wird und nahe 0, wenn viele Pixel der Struktur am Rand des Bildraumes liegen.³⁶ IC wird wie folgt berechnet:

³⁵Bei HNO-Operationen werden z.B. Lymphknoten als wichtig bewertet, während Knochen als unwichtig gelten.

³⁶Aufgrund der Größenverhältnisse von IC und A (A wird meist viel kleiner als IC sein), kommt IC praktisch eine größere Bedeutung zu, was den realen Beobachtungen entspricht: Nutzer „platzieren“ Strukturen von Interesse zumeist im Zentrum des Bildraumes.

$$IC = 1 - \frac{\sum_{j=1}^n x_j + y_j}{n * (is_w + is_h)} \tag{8.5}$$

Dabei sind x und y die Koordinaten jedes Pixels j im Bildraum, n die Anzahl der sichtbaren Pixel einer Struktur und is_w bzw. is_h sind die Breite und Höhe des Bildraumes. Die Daten zur Lage einer Struktur im Bildraum sowie zu möglichen verdeckten Flächen werden aus den Parametern gewonnen, wie sie für die Sichtpunktbestimmung ermittelt wurden (Kapitel 4). Am Ende der Berechnung steht die Struktur mit dem höchsten *SOI*-Wert als die wahrscheinlichste Struktur von Interesse im Fokus des Arztes. Diese Struktur wird in I gespeichert.

Der automatische Ansatz hat sich in einigen Fällen als nicht genau genug herausgestellt und sollte daher nur zum Einsatz kommen, wenn keine Nutzerinteraktion möglich oder gewünscht ist. Weiterhin kann dieser Ansatz nicht für 2D-Darstellungen genutzt werden, wenn in einem Schichtbild die Overlays von mehreren Strukturen dargestellt werden. Die Lage im Bildraum spielt im 2D-Schichtbild keine Rolle für die mögliche Wichtigkeit einer Struktur. Daher kommt, wo immer es möglich ist, eine Interaktion mit dem Arzt in Betracht. Dabei wird dem Arzt bei der Erstellung eines Keystates eine Liste möglicher Strukturen präsentiert, aus denen er die Strukturen von Interesse in der aktuellen Sicht auswählt. Diese Liste ist dahingehend optimiert, dass die im automatischen Verfahren ermittelten Kandidaten mit den höchsten *SOI*-Werten prominent angezeigt werden.

Mit dem Wissen über die Struktur von Interesse können die Werte ermittelt werden, die nötig sind, um den Sichtpunkt auf diese Struktur in anderen Darstellungen rekonstruieren zu können. Für jeden Parameter, wie er zur Sichtpunktbestimmung genutzt wird (Kapitel 4), wird der skalare Wert an der aktuellen Kameraposition ermittelt.³⁷ Der Parameterwert p_i wird im Vektor P mit $P = (p_1, p_2, \dots, p_n)$ gespeichert, wobei n die Anzahl der Parameter ist. Das Gewicht eines jeden Parameters w_i wird im Vektor W mit $W = (w_1, w_2, \dots, w_n)$ gespeichert. Die Werte der Gewichte basieren auf den ermittelten Werten im Zuge der automatischen Sichtpunktbestimmung, sind applikationsabhängig und für alle Keystates einer Applikation gleich. Um dennoch mögliche persönliche Abweichungen der Gewichte in späteren Entwicklungen zu erlauben (bspw. durch ein System, welches die persönlichen Präferenzen des Nutzers erfasst), werden die Gewichte in jedem Keystate erfasst. Die Werte zur Rekonstruktion eines Sichtpunktes werden im 2-Tupel V gespeichert, mit:

$$V=(W, P) \tag{8.6}$$

³⁷Da die aktuelle Kameraposition zumeist nicht identisch mit einer vorberechneten Position auf der umgebenden Kugel ist, wird die nächstgelegene Kameraposition aus den vorberechneten Daten genutzt.

Für 2D-Schichtbilddarstellungen ist es nicht ausreichend, die absolute Nummer des aktuellen Schichtbildes zu speichern, so wie es in der 3D-Darstellung nicht reicht, die absolute Kamerakoordinate zu speichern. Für kleine Strukturen wie Lymphknoten, die nur in wenigen Schichten sichtbar sind, ist es ausreichend, zur Rekonstruktion eine Schicht zu finden, in der diese Struktur ebenfalls auftaucht. Für größere, elongierte Strukturen wie Gefäße ist dieser Ansatz nicht Erfolg versprechend. Daher wird zusätzlich die im aktuellen Schichtbild sichtbare Kontextinformation in Form der sichtbaren Strukturen ermittelt und im Vektor cs mit $cs = (cs_1, cs_2, \dots, cs_m)$ gespeichert, wobei m die Anzahl der sichtbaren Kontextstrukturen ist.

Es ist offensichtlich, dass die Kontextstrukturen unterschiedliche Bedeutung und damit Wichtigkeiten besitzen. Zum Beispiel hat ein Muskel im Bereich eines Tumors der Halsregion eine größere Bedeutung für die Planung einer Lymphknotenausräumung als der Schädelknochen. Daher werden neben den Kontextstrukturen auch deren Wichtigkeiten cs_{imp} in einem Keystate gespeichert:

$$cs_{imp} = (cs_{imp_1}, cs_{imp_2}, \dots, cs_{imp_m}).$$

Damit stellt C ein 2-Tupel dar, mit:

$$C = (cs, cs_{imp}) \quad (8.7)$$

Da nun bekannt ist, wie die benötigten Informationen für einen Keystate ermittelt werden, wird im folgenden Abschnitt die Anwendung eines Keystates auf einen neuen Datensatz beschrieben.

8.3.3 Wiederverwendung von Keystates

Das Hauptziel der Keystates ist die Wiederverwendung einer großen Zahl an Visualisierungsparametern für andere, ähnliche Datensätze. Wendet man einen Keystate auf einen neuen Datensatz an, so sollte eine Visualisierung erzeugt werden, die vergleichbar der Visualisierung ist, auf deren Basis der Keystate erstellt wurde. Um dieses Ziel zu erreichen, müssen drei Schritte absolviert werden:

1. Die visuellen Stile, wie Farbe oder Transparenz, die in einem Keystate gespeichert sind, müssen auf die Strukturen im neuen Datensatz angewendet werden.
2. Für 3D-Visualisierungen muss ein Sichtpunkt gefunden werden, der dem ursprünglichen Sichtpunkt ähnlich ist und auf den gespeicherten Sichtpunktinformationen im Keystate basiert. Für 2D-Darstellungen muss eine Schicht gefunden werden, die die Struktur von Interesse sowie die passenden Kontextstrukturen enthält.
3. Die Struktur von Interesse des Keystates muss auf ihr Vorkommen im neuen Datensatz hin analysiert werden. Davon abhängig kann ein Keystate unter Umständen nicht angewendet werden (wenn die Struktur von Interesse im neuen Datensatz nicht existiert) oder sie muss vervielfältigt werden

(wenn z.B. mehrere Tumore anstelle eines Tumors im Datensatz identifiziert werden können).

Anwendung von Stilen. Die Anwendung von im Keystate gespeicherten Stilen wird durch die Gruppierung von Strukturen bei der Erstellung des Keystates begünstigt. Dadurch können Stile für ganze Gruppen von Strukturen angewendet werden (z.B. „Lymphknoten auf der linken Seite“ oder „Muskeln“), unabhängig davon, wie viele Strukturen im aktuellen Datensatz zur Gruppe gehören oder ob überhaupt Strukturen zur Gruppe gerechnet werden können. Das heißt, es ist egal, ob im ursprünglichen Datensatz 10 oder 25 Lymphknoten enthalten waren. Sind im aktuellen Datensatz neue Strukturen enthalten, die nicht im Ursprungsdatensatz enthalten waren, so werden diese mit den für ihre anatomische Zugehörigkeit festgelegten Standardstilen dargestellt.

Wahl des Sichtpunktes. Um einen angemessenen Sichtpunkt zu ermitteln, der der Sicht im ursprünglichen Datensatz entspricht werden die im Sichtpunktvektor P und im Vektor für die Parametergewichte W gespeicherten Werte genutzt. Für jeden möglichen Sichtpunkt i in der Szene wird ein Parametervektor p_{new_i} auf der Basis der Sichtpunktinformationen der neuen Szene erzeugt:

$$P_{new_i} = (p_{new_1}, p_{new_2}, \dots, p_{new_n})$$

Die Parametervektoren, P vom Keystate und p_{new_i} für jeden Sichtpunkt in der neuen Szene, werden mit den im Keystate gespeicherten Gewichten für die verschiedenen Parameter gewichtet:

$$p_w = (w_1 * p_1, w_2 * p_2, \dots, w_n * p_n) \tag{8.8}$$

$$P_{new-w} = (w_1 * p_{new_1}, w_2 * p_{new_2}, \dots, w_n * p_{new_n}) \tag{8.9}$$

Der Sichtpunkt mit der kürzesten euklidischen Distanz im n -dimensionalen Parameterraum (wobei n die Anzahl der Parameter ist) zum gewichteten Parametervektor p_w des Keystates wird als derjenige Sichtpunkt gewählt, der dem Sichtpunkt im Ursprungsdatensatz am nächsten kommt und somit eine vergleichbare Sicht auf die Szene gewährt.

Ermittlung der Schicht für die 2D-Darstellung. Um die darzustellende Schicht im 2D-Schichtbilddatensatz zu ermitteln, wird das Auftreten der Struktur von Interesse im neuen Datensatz untersucht. Dazu werden für jede Schicht die Kontextstrukturen analysiert und die Summe ihrer Wichtigkeitswerte wird ermittelt. Das Schichtbild, welches die Struktur von Interesse enthält und den höchsten Wert für die Wichtigkeiten der Kontextstrukturen erhält, wird als das wahrscheinlichste angenommen und angezeigt. Dabei gibt es zwei Spezialfälle zu unterscheiden:

1. Existieren mehrere Schichten mit den gleichen Kontextstrukturen und damit mit dem gleichen Wichtigkeitswert, wird die Schicht gewählt, die aus der momentanen Sichtrichtung (z.B. axial) am zentralsten liegt.

2. Existieren mehrere Schichten mit dem gleichen Wichtigkeitswert für die Kontextstrukturen, jedoch unterschiedlichen Kontextstrukturen, so werden mehrere Instanzen des aktuellen Keystates dargestellt – für jede in Frage kommende Schicht eine.

Mehrfache Instanzen von Keystates. Neben dem eben besprochenen Spezialfall für die 2D-Schichtbilddarstellung gibt es weitere Fälle, die eine Adaption der Anzahl der Keystates für einen neuen Datensatz notwendig machen. Selbst wenn die Datensätze äquivalent bezüglich der medizinischen Fragestellung und anatomischen Region sind, so treten dennoch oft Unterschiede in der Anzahl und Existenz einzelner segmentierter Strukturen auf. Beispiele dafür sind:

- Der ursprüngliche Datensatz enthielt einen Tumor, der exploriert und in einem Keystate gespeichert wurde. Ein neuer Datensatz kann dagegen drei Tumore enthalten.
- Im ursprünglichen Datensatz wurden drei vergrößerte Lymphknoten untersucht und in drei Keystates gespeichert. Ein neuer Datensatz kann dagegen nur einen vergrößerten Lymphknoten enthalten.

Daher werden die in einem Keystate gespeicherten Strukturen auf ihre Existenz im neuen Datensatz hin untersucht:

- Existiert die Struktur, auf die sich der Keystate bezieht, nicht, so wird dieser Keystate verworfen.
- Befinden sich weniger Strukturen im neuen Datensatz, als Visualisierungen durch entsprechende Keystates erzeugt werden sollen, so werden die überflüssigen Keystates verworfen.
- Befinden sich mehr Strukturen vom Typ der Struktur von Interesse im neuen Datensatz, so werden zusätzliche Instanzen des Keystates erzeugt.

8.3.4 Vergleichbare Konzepte

Zum besseren Verständnis der Keystates kann ein Blick auf vergleichbare Konzepte aus anderen Anwendungsdomänen hilfreich sein. Schon früh wurden INI-Dateien genutzt, um bestimmte Voreinstellungen zu definieren. Durch eine INI-Datei konnten viele Parameter einer Anwendung wiederholbar automatisch gesetzt werden, ohne die Parameter erneut eingeben zu müssen. Im Umfeld von Textverarbeitungen kommen sogenannte Formatvorlagen zum Einsatz. Über eine Formatvorlage kann einem Textteil oder einem ganzen Dokument ein einmal eingestellter Stil (eine Kombination aus Schriftart und -größe, Zeilenabstand, Schriftschnitt u.v.m.) zugewiesen werden (Abb. 8.2). Der Nutzer muss sich dabei nur einmalig mit dem Aussehen seines Textes befassen und kann einmal definierte Stile für beliebige weitere Dokumente verwenden. In der Domäne der Webentwicklung gibt es Templates als Definition des Erscheinungsbildes von Webseiten.

Abbildung 8.2: Auswahl von Formatvorlagen in Microsoft®Word 2007 für unterschiedliche Textdarstellungen.



Basierend auf Techniken wie den Cascading Style Sheets (CSS) kann ein Webentwickler einmal definierte Stile und Farben schnell auf alle Seiten einer umfangreichen Website anwenden. Im Kontext dieser Konzepte sind auch die Keystates zu sehen. Sie erlauben die einfache Wiederverwendung einmal getroffener Visualisierungsentscheidungen.

8.4 Anwendung von Keystates in der Interaktion

Damit die Keystates ihr volles Potenzial entfalten können, müssen sie gut in den Applikations-Workflow integriert sein. Keystates können dabei automatisch oder manuell erstellt werden. Für die automatische Lösung sind verschiedene Aspekte zu berücksichtigen, die im nächsten Abschnitt diskutiert werden. Verschiedene Anwendungsgebiete der Keystates werden in Abschnitt 8.4.2 vorgestellt.

8.4.1 Automatische Generierung von Keystates

Automatisch erzeugte Keystates können für die Falldokumentation nützlich sein oder einem Undo-Verlauf einer Applikation als Grundlage dienen. Im Unterschied zu manuell erzeugten Keystates ist es bei automatisch erzeugten schwieriger und fehleranfälliger, die Struktur von Interesse zu bestimmen (siehe Abschnitt 8.3.2). Desweiteren müssen Bedingungen definiert werden, **wann** ein Keystate erzeugt wird. Eine Möglichkeit ist die Erzeugung eines Keystates nach jeder erfolgten Interaktion, wie die Veränderung einer Farbe oder das Verschieben der Kamera. Diese Granularität hat sich als zu fein erwiesen, da zu viele Keystates erzeugt werden. So führt das Ausblenden von fünf Strukturen kurz nacheinander zu fünf Keystates mit geringen Unterschieden, obwohl am Ende dieser Aktion nur eine neue Visualisierung steht, die dem Betrachter interessant erscheint. Daher sieht das Verfahren einen Zeitbereich vor, in dem alle Änderungen zu

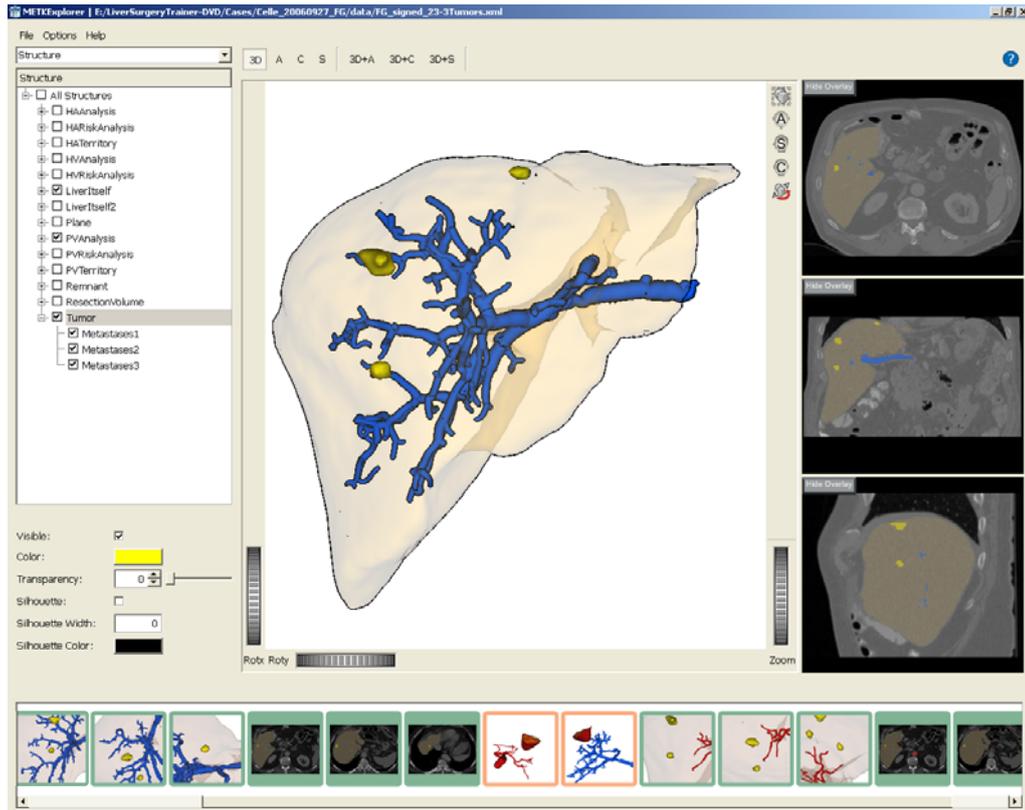


Abbildung 8.3: Verwendung von Keystates in einer Applikation. Die horizontale Liste am unteren Fensterrand enthält alle momentan verfügbaren Keystates. Keystates können von einem älteren Datensatz geladen oder automatisch (roter Rahmen) oder manuell (grüner Rahmen) neu erzeugt werden.

einem einzigen Keystate zusammengefasst werden. Dieser Zeitbereich liegt standardmäßig bei 3 Sekunden, kann aber angepasst werden.³⁸ Zur Diskretisierung der manuellen Kamerabewegung wird ein Keystate erst nach einer Mindestzeit nach dem Ende einer Bewegung erzeugt. Wird innerhalb dieser Zeit die Bewegung der Kamera wieder aufgenommen, so wird kein Keystate erzeugt. Dadurch werden mehrere Kamerabewegungen zur Exploration in einem Keystate zusammengefasst. Um in der Darstellung der erzeugten Keystates in der Applikation manuell von automatisch generierten unterscheiden zu können, werden sie mit unterschiedlichen Farben prominent separiert (Abb. 8.3).

³⁸Dabei wird im Sinne einer einfachen Nutzbarkeit keine Feineinstellung der Zeit zugelassen, sondern ein Schieberegler mit einer Beschriftung wie „Hohe Empfindlichkeit“ und „Schwache Empfindlichkeit“ angeboten.

8.4.2 Anwendungsgebiete für Keystates

Im Folgenden werden drei ausgesuchte Anwendungsgebiete für automatisch generierte Keystates während des Explorationsprozesses vorgestellt: a) Undo-Verläufe, b) die Weitergabe von Workflows und c) die Dokumentation.

Undo-Verläufe: Jeder automatisch erzeugte Keystate kann als ein Undo-Status in einer Visualisierungs-Historie dienen. Durch die Auswahl eines zuvor erzeugten Keystates wird instantan diese Visualisierung wieder hergestellt. Wird die Exploration von einem früheren Keystate aus fortgeführt, so wird der Undo-Verlauf weder verzweigt (wie es z.B. GROTH & STREEFKERK [2006] vorschlagen) noch werden alle danach gespeicherten Keystates gelöscht und überschrieben (wie es z.B. in Adobe Photoshop® von ADOBE SYSTEMS [2009] gehandhabt wird). Diskussionen mit Nutzern ergaben, dass ein Hinzufügen aller neuen Keystates am Ende des Verlaufs gewünscht wurde. Dadurch wird das lineare Erscheinungsbild des Undo-Verlaufs beibehalten.

Weitergabe von Workflows: Automatisch im Hintergrund erstellte Keystates erlauben die einfache Weitergabe von Workflows zwischen verschiedenen Chirurgen oder zwischen Chirurgen und Radiologen. So kann der Radiologe eine erste grobe Planung aus seiner Sicht vornehmen und dem Chirurgen anbieten. Dieser kann diese Planung und die erfolgten Explorationen nachvollziehen, verändern und erweitern. Dies ist im Besonderen nützlich, wenn Chirurg und Radiologe räumlich weit voneinander entfernt arbeiten oder wenn die Aufbereitung und Segmentierung eines Datensatzes durch externe Dienstleister weltweit erfolgt.

Dokumentation: Die Dokumentation des Planungsprozesses wird derzeit nur unzureichend von den Softwarelösungen vor Ort unterstützt. Keystates können dabei als Repräsentation der durchgeführten Planung direkt gespeichert und in Form jeglicher anderer Medien (Screenshots, Videos, interaktive Darstellungen) in einer digitalen Patientenakte gespeichert werden. Gerade aus Keystates generierte Animationen können dabei einen detaillierten Einblick in die Vorgehensweise und Planungsschritte des Chirurgen geben.

8.5 Animations-Authoring mit Keystates

Neben ihrer unterstützenden Funktion im individuellen Explorationsprozess der Operationsplanung dienen Keystates als Kernelemente eines Systems zur automatischen Generierung von Animationen. Ein Keystate kann mit einem in der klassischen Animationserstellung verwendeten Keyframe verglichen werden. Ein Keystate beschreibt dabei den Zustand einer kompletten Szene zu einem bestimmten Zeitpunkt, wo ein Keyframe nur ein spezielles, herausragendes und wichtiges Bild in einer Folge von Bildern ist. Wo bei einer klassischen Animation zwischen zwei Keyframes die einzelnen Bilder interpoliert werden, werden bei der Nutzung von

Keystates in Animationen die verschiedenen Parameter eines Keystates, wie Stile und Kamerapositionen, interpoliert.

Da Keystates automatisch auf einen neuen Datensatz zur Generierung einer Visualisierung angewendet werden können, ist mit der Nutzung von Animationen auch die automatische Erstellung von einmal über Keystates definierte Animationen auf der Basis neuer Datensätze möglich. Wie ein Chirurg im Detail auf der Basis von Keystates einfach und schnell Animationen erstellen kann, wird im folgenden Abschnitt beschrieben. Anschließend werden einige Aspekte zu Animationsübergängen vorgestellt und diskutiert (siehe Abschnitt 8.5.2).

8.5.1 Interaktionstechniken zum Animations-Authoring

Die Anwendung der Keystates zur Erstellung von Animationen ist zu großen Teilen ein MCI-Problem.³⁹ So können Keystates automatisch oder manuell erstellt werden. Alle Keystates eines Datensatzes werden in Form von Thumbnails in einer Liste angezeigt. Um eine Animation zu beschreiben, zieht der Nutzer die Keystates, die er für die Animation verwenden möchte, auf eine Arbeitsfläche – das Storyboard (Abb. 8.4). Dort können die Keystates in einer beliebigen Reihenfolge angeordnet werden. Zusätzlich muss der Chirurg nur zwei weitere Parameter angeben: Die Zeit für den Übergang zwischen zwei Keystates und die Zeit, die die Kamera, respektive der Betrachter, an jedem Keystate verweilen soll. Die Eingabe der Übergangszeiten erfolgt auf abstrakte Art und Weise: Es wird ein Schieberegler mit Werten wie „langsam“ und „schnell“ angeboten. Außerdem werden keine numerischen Werte verwendet, da die Übergangszeit zwischen zwei Keystates zusätzlich durch die Distanz zwischen den Sichtpunkten (die Länge des Kamerapfades) beeinflusst wird. Auch wenn die Angabe einer individuellen Wartezeit für jeden Keystate möglich ist, so wird in der klinischen Routine zumeist ein Standardwert für alle Keystates verwendet.

Ein spezieller Übergang ist eine Rotation. Da hier nur ein Keystate anstelle der sonst üblichen zwei Keystates zum Einsatz kommt, markiert der Chirurg einen solchen Keystate per Doppelklick oder Kontextmenu und gibt anschließend die Rotationsachse (axial, sagittal, coronar) an. Solche Keystates werden in der Darstellung aller Keystates einer Animation mit einem besonderen Icon markiert und hervorgehoben.

Zusammenstellungen von Keystates zur Beschreibung von Animationen können für die spätere Wiederverwendung in anderen Datensätzen gespeichert werden. Dadurch kann eine einmal definierte Animation ohne zusätzliche Interaktionen für andere Datensätze wiederverwendet werden. Die Animationen können sich dabei in ihrer Gesamtlänge und der Anzahl der Keystates unterscheiden.

³⁹Mit MCI wird allgemein die „Mensch-Computer-Interaktion“ abgekürzt. Andere geläufige Synonyme sind die GUI (Graphical User Interface) oder einfach nur UI (User Interface).

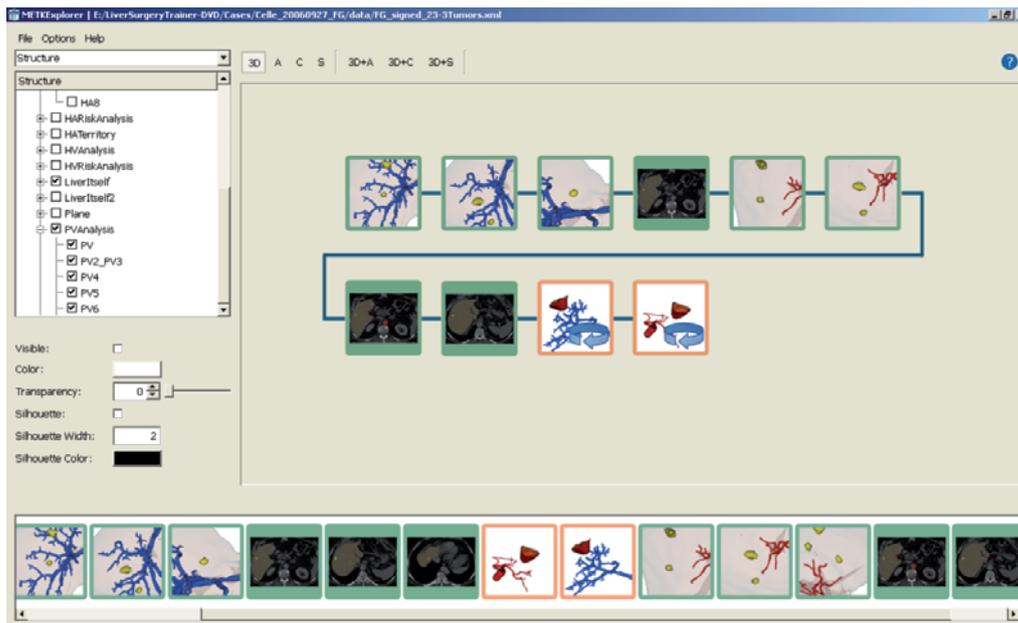


Abbildung 8.4: Storyboard zur Anordnung von Keystates für eine Animation. Keystates können per Drag&Drop in einer beliebigen Reihenfolge zu einer Animation komponiert werden. Wenn eine Rotation um eine Darstellung erfolgen soll, so kann dies durch einen Doppelklick aktiviert werden. Ein solcher Keystate wird durch ein spezielles Icon hervorgehoben.

8.5.2 Animations-Übergänge

Neben den Keystates selbst, die einzelne wichtige Sichten definieren, sind die Übergänge zwischen den Keystates eine Quelle zusätzlicher Informationen für den Betrachter einer Animation. Generell ist ein Übergang die Interpolation zwischen zwei Keystates. Dabei muss jeder Parameter, wie die Farbe einer Struktur oder die Kameraposition, gesondert interpoliert werden. Da je nach Art des Parameters andere Anforderungen an eine Interpolation gestellt werden müssen, werden drei Arten von Übergängen unterschieden:

- die Interpolation von Stilparametern,
- die Bewegung der Kamera,
- der Übergang von 3D-Szenen auf 2D-Darstellungen und zwischen verschiedenen Schichten in einer 2D-Schichtbilddarstellung.

Die Bewegung der Kamera wurde bereits in Kapitel 6 einer ausführlichen Analyse unterzogen. Im Folgenden wird daher auf die Interpolation von Stilparametern sowie auf Übergänge im Zusammenhang mit 2D-Schichtbilddarstellungen näher eingegangen.

Interpolation von Stilparametern

Zur Interpolation von Stilparametern, wie Farbe oder Transparenz einer Struktur, ist die lineare Interpolation der jeweiligen Werte über die Zeit des Übergangs naheliegend. Da sich bei einem Übergang aber auch zumeist die Kamera bewegt, sind nicht alle sich verändernden Strukturen die gesamte Zeit über sichtbar. Daher werden alle Interpolationen von Stilparametern in den ersten 50% eines Übergangs durchgeführt. Somit kann der Betrachter sich im zweiten Teil eines Übergangs auf die Zielstrukturen konzentrieren und muss nur noch die Kamerabewegung mental verarbeiten, um die Orientierung im Raum zu behalten. Zur Interpolation von booleschen Werten, wie Sichtbarkeit, werden diese Parameter in numerische Werte überführt. So können auch diese Werte kontinuierlich über die Zeit interpoliert werden und abrupte Veränderungen, wie das plötzliche Auftauchen einer Struktur, werden vermieden.

Übergänge bei 2D-Schichtbilddarstellungen

Der Übergang von einer 3D- in eine 2D-Darstellung und umgekehrt sollte nicht abrupt geschehen. Kontinuierliche Übergänge werden als angenehmer wahrgenommen und können durch den Betrachter besser interpretiert werden. Die Abbildung der 3D- auf die 2D-Informationen, z.B. die Lage des gewählten Schichtbildes in der 3D-Szene, sind wichtige Informationen, die dargestellt und erhalten werden müssen. Daher wird beim Übergang zwischen den verschiedenen Dimensionen eine spezielle Form der Überblendung verwendet (Abb. 8.5). Zunächst wird das Schichtbild in der 3D-Szene an seiner korrekten Position eingeblendet. Anschließend wird die Kamera in eine Position senkrecht über dem Schichtbild bewegt, wobei die 3D-Strukturen weiterhin sichtbar bleiben. Abschließend wird von den 3D-Strukturen auf die 2D-Overlays des Schichtbildes übergeblendet. Da die Stilparameter der Strukturen zwischen 2D- und 3D-Darstellungen synchron gehalten werden (siehe dazu auch die softwaretechnischen Ausführungen in Abschnitt 9.3.1), werden die Strukturen mit den gleichen Farben in 3D und 2D dargestellt, was den Abbildungsprozess zusätzlich unterstützt. Der Übergang von einer 2D-Schichtbilddarstellung auf eine 3D-Szene erfolgt in gleicher Weise, nur in umgekehrter Reihenfolge.

Zur Animation des Übergangs zwischen zwei Schichtbildern ist ein einfaches Überblenden der Schichten ungenügend. Daher wird vom Start- zum Zielschichtbild gescrollt, wobei jedes Schichtbild zwischen den beiden Schichten angezeigt wird. So behält der Betrachter die Orientierung und erhält als zusätzliche Information die Lage der Schichten zueinander. Außerdem entspricht diese Darstellungsform dem normalen Scrollen eines Arztes in den Schichtbildern und bildet diese 1:1 ab.

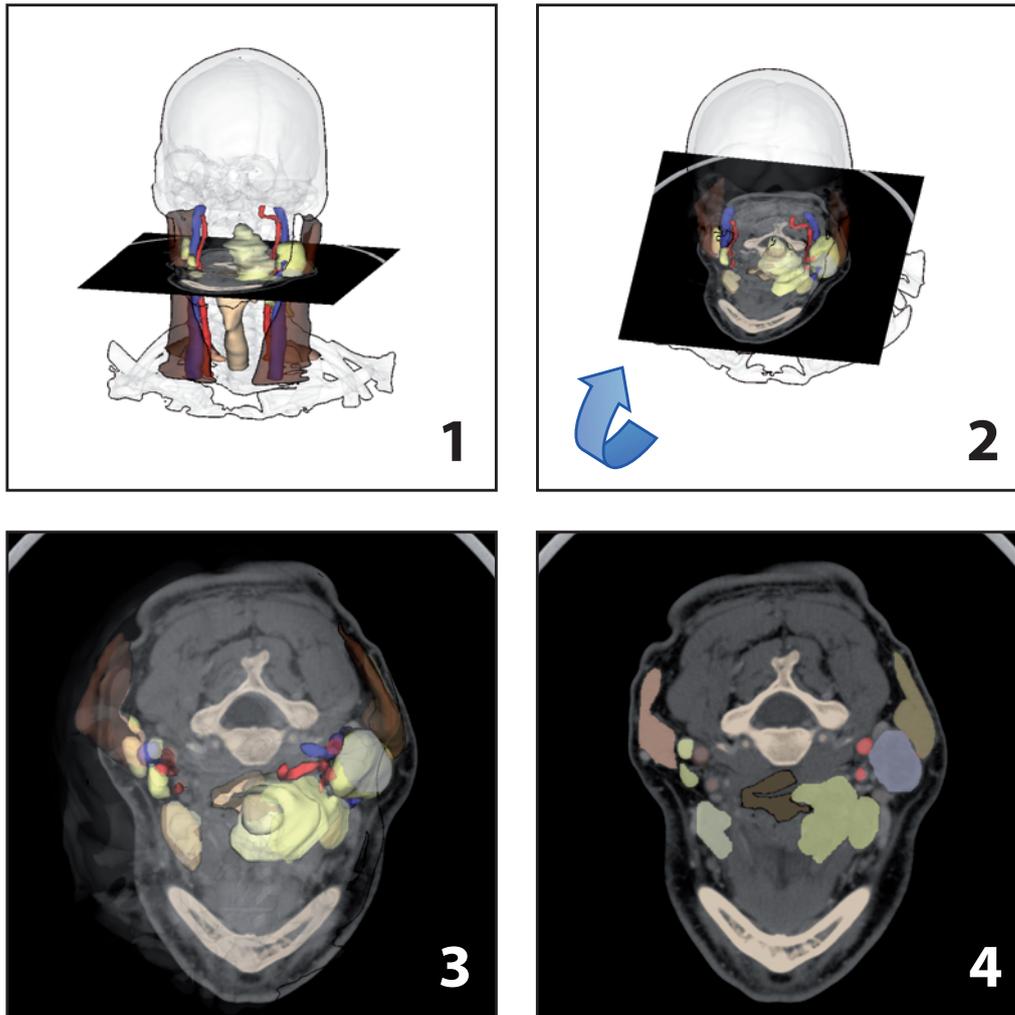


Abbildung 8.5: Ein animierter Übergang von einer 3D-Szene in eine 2D-Schichtbilddarstellung. Die Schicht wird in die 3D-Darstellung eingeblendet, die Kamera bewegt sich senkrecht über die Schicht und die 3D-Strukturen werden in ihre 2D-Overlays übergeblendet.

8.6 Beispielanwendungen für Keystates

Die Keystates wurden exemplarisch für die Planung von 21 Tumorresektionen der Leber, 7 Leberlebenspenden sowie 14 Fällen von Halslymphknotenausräumungen eingesetzt. In diesem Abschnitt werden einige Keystates der Tumorresektion sowie der HNO-Operationsplanung in ihrem medizinischen Kontext dargestellt.

Alle im Folgenden beschriebenen Keystates sind in Auszügen in den Abbildungen 8.6 (für die Leberoperationsplanung) und 8.7 (für die HNO-Planung) dargestellt. Dabei werden neben dem ursprünglichen Fall, auf dessen Basis die Keystates erzeugt wurden, zusätzlich jeweils zwei weitere Fälle gezeigt.

Keystates in der Leberoperationsplanung

Zuerst betrachtet der Chirurg die verschiedenen Gefäßanomalien, wie atypische Trifurkationen an kritischen Punkten beider Gefäßsysteme: der Portalvene (blau) und Leberarterie (rot). Dies spiegelt Keystate 1 in Abb. 8.6 wider. Der Aspekt ist allein auf der Basis von 2D-Schichtbildern schwierig zu beurteilen. Daher wurde die 3D-Ansicht gewählt. Als nächstes inspiziert der Chirurg die Leber hinsichtlich vorhandener Metastasen unter Berücksichtigung von Infiltrationen verschiedener Gefäße. Dazu blendet er die Portalvene und die Metastase sowie das Leberparenchym als Kontextinformation ein. Er wählt einen Sichtpunkt, von dem aus die minimale Distanz der Metastase zur Portalvene bzw. deren Infiltration am besten zu sehen ist (Keystate 2). Er überprüft seine Einschätzungen in den Schichtbildern (Keystate 3). Die Keystates wurden auf Basis von „Patient 1“ erstellt, der nur eine Metastase enthält. In der Leber von „Patient 3“ wurden drei Metastasen gefunden. Daher werden automatisch zusätzliche Instanzen von Keystate 2 und 3 erzeugt (in Abb. 8.6 sind aus Platzgründen nur zwei der drei Keystates dargestellt). Anschließend wiederholt der Chirurg die Inspektion von Infiltrationen für den zweiten wichtigen Gefäßbaum, die Leberarterie (Keystate 4 – der Keystate für die 2D-Darstellung ist in Abb. 8.6 nicht abgebildet, ist aber vergleichbar dem Keystate 3). Danach inspiziert der Chirurg die vorgesehene Resektions-ebene unter besonderer Berücksichtigung der Schnittpunkte mit den großen Gefäßen. Dabei wird zunächst das Remnant (grün) im Zusammenhang mit der Leberarterie (Keystate 5) und der Portalvene (Keystate nicht in der Abbildung) untersucht. Anschließend wird das Resektionsvolumen (orange) untersucht – ebenfalls im Zusammenhang mit der Leberarterie (Keystate nicht in der Abbildung) und der Portalvene (Keystate 6). Die Keystates wurden auf Basis eines Datensatzes erzeugt, bei dem nur eine Resektionsebene geplant wurde. Für „Patient 2“ wurden zwei Resektionsvorschläge erstellt. Daher wurden von den Keystates 5 und 6 automatisch zusätzliche Instanzen erzeugt. Für „Patient 3“ wurde kein Resektionsvorschlag erstellt. Daher wurden die entsprechenden Keystates verworfen. Es wurden weitere Explorationen zur Inspektion des Resektionsvolumens mit verschiedenen vaskulären Systemen in 2D und 3D durchgeführt, um abschätzen zu können, ob jeweils ein ausreichender Sicherheitsrand um die Metastase eingehalten werden kann. Diese Keystates sind hier nicht dargestellt.

Keystates in der HNO-Operationsplanung

Der HNO-Chirurg inspiziert als erstes den Tumor im Kontext aller segmen-

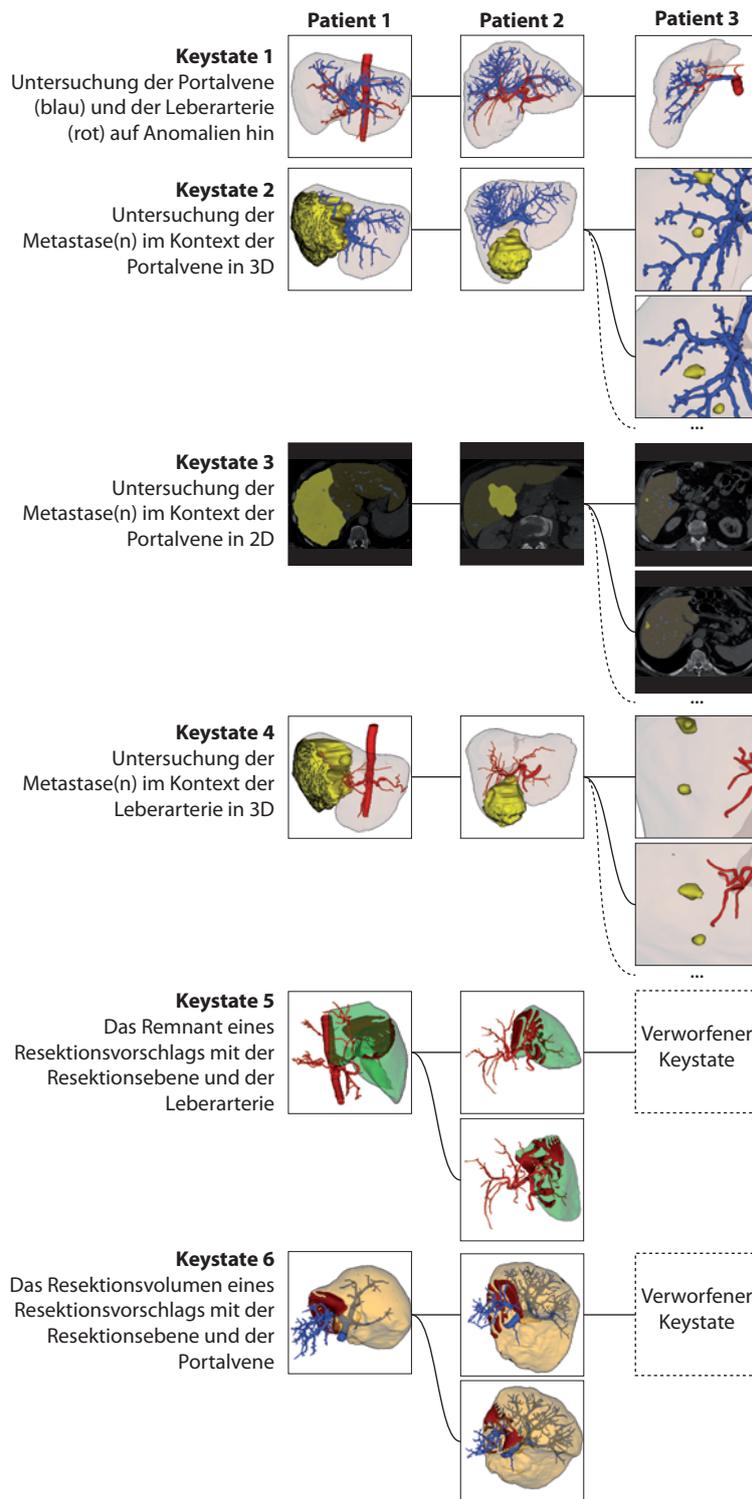


Abbildung 8.6: Keystates für die Planung von Leberoperationen. Jede Spalte repräsentiert einen Datensatz. Von einigen Keystates wurden mehrere Instanzen erzeugt. Ebenso wurden Keystates verworfen, weil der Resektionsvorschlag im 3. Datensatz nicht vorhanden ist.

tierten Strukturen im Halsbereich (Keystate 1 in Abb. 8.7). Er wiederholt dies in der 2D-Schichtbilddarstellung (Keystate 2). Anschließend bewertet er die Abstände aller Lymphknoten zu verschiedenen Risikostrukturen auf der linken und rechten Halsseite: Muskeln, Arterien und Venen. Er untersucht den *Musculus sternocleidomastoideus* auf der linken Halsseite (Keystate 3) und fährt mit der *Arteria carotis* (Keystate 4) und der *Vena jugularis* (Keystate 5) fort. Der Chirurg schließt die Betrachtung mit den gleichen Strukturen auf der rechten Halsseite ab (Keystates nicht dargestellt). Für das Tumorboard erstellt er eine Animation, bei der eine Rotation um jede Risikostruktur ausgeführt wird.

8.7 Diskussion und Zusammenfassung

Das vorgestellte Konzept der Keystates ermöglicht die Wiederverwendung von Visualisierungen und Animationen sowie das effiziente Authoring von Animationen. Durch die Keystates kann der planende Chirurg auf einer visuellen Ebene ohne komplizierte Skripte oder Definitionen beschreiben, wie zukünftige Visualisierungen zur Planung von Operationen aussehen sollen. Zusätzlich wird durch die einfache Generation von Animationen der (interdisziplinäre) Planungsprozess unterstützt. Der Ansatz ist unabhängig von der zugrunde liegenden Bildmodalität und setzt lediglich segmentierte Strukturen voraus.

Ein noch offener Aspekt ist die nachträgliche Bearbeitung von einzelnen Keystates. Die Möglichkeit der späteren Anpassung von Keystates durch den Chirurgen erscheint interessant und nützlich und könnte zum Einsatz kommen, wenn der Chirurg bei neuen und zukünftigen Fällen Aspekte visualisiert sehen möchte, die so bei der Erstellung der Keystates mit dem ersten Fall nicht bedacht wurden oder nicht möglich waren. Anstelle der Erstellung eines neuen Keystates (was immer noch eine sehr effiziente Möglichkeit wäre) könnten Explorationen und Modifikationen in neuen Datensätzen die bestehenden Keystates verändern und anpassen.

Eine andere offene Frage ist der Transfer von Keystates zwischen verschiedenen chirurgischen Disziplinen. Visualisierungen, die für eine Leberoperationsplanung generiert wurden, könnten z.B. für einen Orthopäden interessant sein – in der Art, dass er eine ähnliche Visualisierung auf der Basis seiner Daten (mit anderen segmentierten Strukturen) generiert haben möchte. Es scheint ein interessantes Anwendungsfeld von Keystates zu sein, wenn ein Chirurg das visuelle Erscheinungsbild einer Visualisierung oder Animation eines Kollegen für seine eigenen Visualisierungen übernehmen könnte. Auch netzbasierte Galerien auf

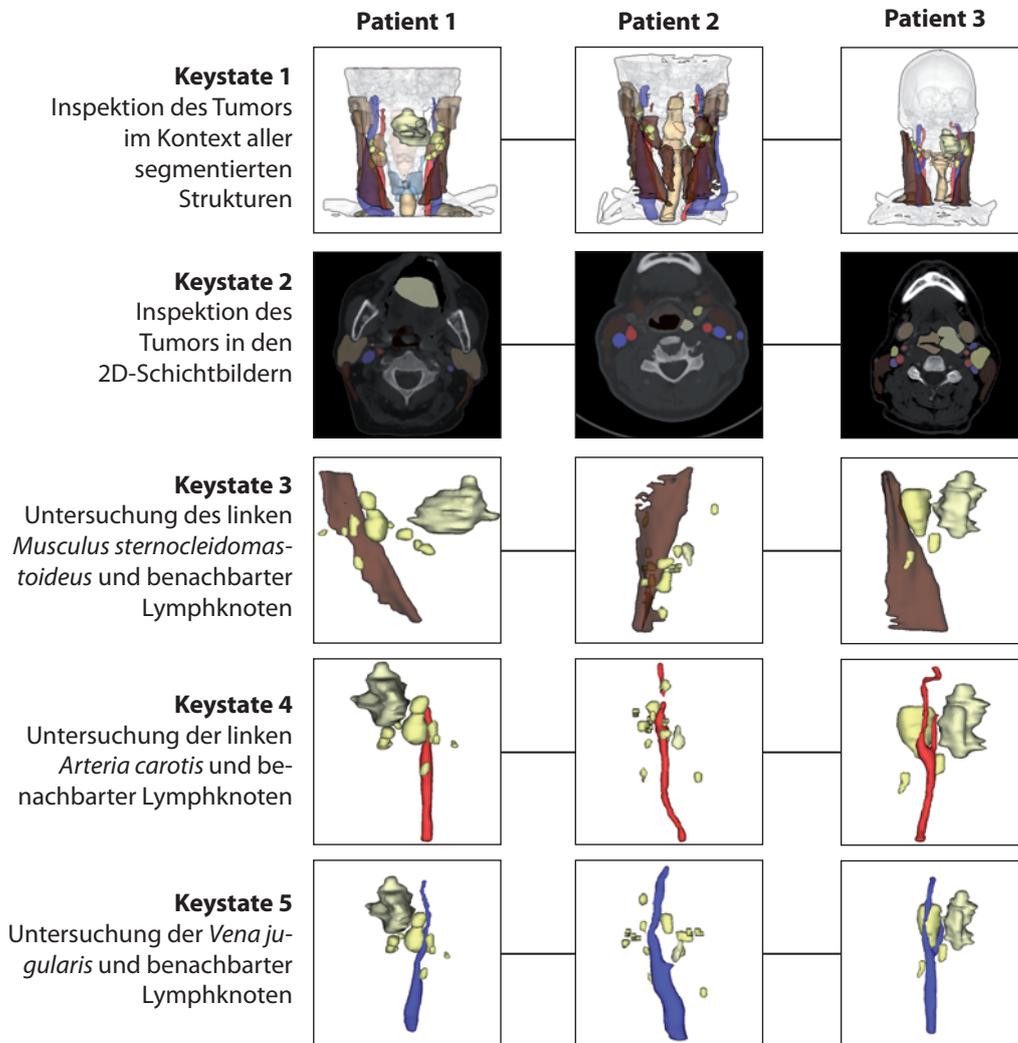


Abbildung 8.7: Beispiel für die Wiederverwendung von Keystates für die Planung von Halslymphknotenausräumungen. Jede Spalte repräsentiert einen Patientendatensatz. Der Chirurg inspiziert den Tumor und die benachbarte Risikostrukturen und Lymphknoten, um deren Infiltration und Abstände zueinander zu beurteilen.

der Basis nutzergenerierter Visualisierungen zur Auswahl von Erscheinungsbildern wären auf dieser Grundlage möglich.

Ein weiterer Aspekt ist die Frage der partiellen Definition von Keystates: Bisher umfassen Keystates die Beschreibung einer kompletten Visualisierung. Jedoch sind Anwendungsfälle denkbar, in denen der Chirurg nur das Erscheinungsbild einzelner Strukturen oder Strukturgruppen übernehmen möchte, bspw. den visuellen Stil der Lymphknoten oder die Art einer Gefäßeinfärbung der Leber. Es erscheint möglich, dass Konzept der Keystates unter diesen Gesichtspunkten zu erweitern.

9

Das Medical Exploration Toolkit

Die im Rahmen dieser Arbeit entwickelten Techniken wurden mehrheitlich in einem Framework implementiert, welches als quelloffenes Toolkit, dem Medical Exploration Toolkit (METK), für die Entwicklungsplattform MEVisLAB vorliegt. Im ersten Teil dieses Kapitels wird die Motivation zur Entwicklung des Toolkits (Abschnitt 9.1) im Kontext existierender Lösungen (Abschnitt 9.2) im Besonderen MEVisLAB (Abschnitt 9.2.2) vorgestellt. Der strukturelle Aufbau (Abschnitt 9.3), die Applikationsentwicklung (Abschnitt 9.4) sowie die Evaluierung des METK (Abschnitt 9.5) sind Themen in der zweiten Hälfte dieses Kapitels.

Eigene Publikationen

Dieses Kapitel basiert u.a. auf folgenden eigenen Publikationen:

- MÜHLER u. a. [2010] Mühler, K.; Tietjen, C.; Ritter, F. & Preim, B. *The Medical Exploration Toolkit: An Efficient Support for Visual Computing in Surgical Planning and Training*. In: **IEEE Transactions on Visualization and Computer Graphics (TVCG)**, 2010
- MÜHLER u. a. [2009] Mühler, K.; Tietjen, C. & Preim, B. *The Medical Exploration Toolkit*. In: **Proc. of Eurographics/IEEE Symposium on Visualization (EuroVis) (Poster)**, 2009
- TIETJEN u. a. [2008] Tietjen, C.; Mühler, K.; Ritter, F.; Konrad, O.; Hindennach, M. & Preim, B. *METK - The Medical Exploration Toolkit*. In: **Proc. of Bildverarbeitung für die Medizin (BVM)**, 2008

9.1 Motivation und konzeptuelle Vorüberlegungen

Die Architektur des METK ist geprägt von mannigfaltigen Erfahrungen in verschiedenen Gebieten der Entwicklung chirurgischer Applikationen. Nach einer Motivation der Entwicklung des METK werden, aufbauend auf den in Kapitel 2 vorgestellten chirurgischen Szenarien, im Folgenden die das METK begleitenden konzeptuellen Überlegungen sowie Anforderungen an chirurgische Applikationen vorgestellt.

9.1.1 Motivation des METK

Chirurgische Applikationen zeichnen sich durch spezielle Anforderungen an die Visualisierungen von und Interaktionen mit Patientendaten aus. Es reicht nicht aus, Schichtbilddaten anzuzeigen und Volumendarstellungen zu erzeugen. Vielmehr ist eine ausführliche Analyse der Bilddaten, verbunden mit angemessenen und ausgefeilten (2D und 3D) Interaktions- und Visualisierungstechniken, nötig. Um die Brücke von der Bildanalyse und einfachen Visualisierungstechniken hin zu chirurgischen Anforderungen an Visualisierung und Interaktion zu schlagen, wurde im Rahmen dieser Arbeit in enger Zusammenarbeit mit Christian Tietjen (TJETJEN [2009]) das Medical Exploration Toolkit (METK) entwickelt (Abb. 9.1).



Abbildung 9.1: Das Icon des METK. Es symbolisiert die Modularität des Toolkits zur Entwicklung von ganzheitlichen Applikationen im chirurgischen Kontext.

Chirurgische Applikationen müssen umfangreiche Informationen im Kontext der Visualisierungen liefern. Diese umfassen neben einer Zuordnung von Strukturen und Pathologien, z.B. Abstands- und Größenmaße, Annotationen oder Resektionslinien sowie Sicherheitsränder. Solche Informationen können nur auf der Basis von Segmentierungen erfolgen, die daher chirurgischen Planungen fast immer vorausgehen und sich abhängig vom chirurgischen Anwendungsbereich nur in ihrem Umfang unterscheiden. Desweiteren sind Segmentierungen ein Weg, den oft schwierigen Abgrenzungen durch überlappende Intensitätswerte der Strukturen in den Bilddaten und Inhomogenitäten innerhalb der Strukturen zu begegnen. Chirurgische Planungen auf der Basis von Segmentierungsergebnissen und dreidimensionalen Visualisierungen finden sich zum Beispiel in der

Abdominalchirurgie, vor HNO-Operationen oder bei orthopädischen Interventionen. Dagegen werden 2D-Schichtbilder bevorzugt in der Neurochirurgie eingesetzt, verbunden mit einem einfachen Volumerendering, wie es auch in der zeitkritischen Notfallmedizin zum Einsatz kommt.

9.1.2 Einordnung des METK

Das METK unterstützt in besonderer Weise die Nutzung segmentierter Daten, hält jedoch keine Funktionen für die Durchführung der Segmentierung selbst bereit. Werkzeuge dafür sind z.B. mit HEPAVISION (BOURQUAIN u. a. [2002]) für die Leberchirurgie und NECKVISION (CORDES u. a. [2006b]) für die HNO-Operationsplanung. Das METK als ein Toolkit zur prototypischen Applikationsentwicklung reiht sich ein in eine Tradition frei verfügbarer Toolkits zur Bildanalyse und Volumenvisualisierung (siehe dazu eine ausführliche Übersicht von CABAN u. a. [2007]). Als Beispiel seien an dieser Stelle das MITK (WOLF u. a. [2005]), der 3DSLICER (GERING u. a. [1999]) und VOLUMESHOP (BRUCKNER & GRÖLLER [2005]) genannt. Jedoch sind diese Toolkits nur mit substantiellen C++-Kenntnissen erweiterbar und auch die Nutzung zur Applikationsentwicklung setzt umfangreiche Programmiererfahrungen voraus. Das METK unterstützt daher einen Applikationsentwicklungsprozess, speziell für chirurgische Applikationen, der keine erweiterten Programmierfähigkeiten erfordert. Grundlage dafür ist die Entwicklungsumgebung MEVISLAB (REXILIUS u. a. [2006]), die eine grafische Programmierschnittstelle zur netzwerkbasierter Entwicklung nutzt. MEVISLAB wurde mit dem METK um entscheidende, meist skriptbasierte Funktionen erweitert:

1. Das METK bietet **fortgeschrittene Visualisierungstechniken** speziell für die chirurgische Anwendung, z.B. Sicherheitsabstände, Gefäßvisualisierungen oder dezidierte Erweiterungen von 2D-Viewern.
2. Es werden **Animationstechniken** angeboten, wie sie in Kapitel 3 vorgestellt wurden.
3. Das METK bietet ein **Fallmanagement**, das das effiziente Laden und Speichern großer Mengen segmentierter Strukturen sowie zusätzlicher Patienteninformationen und Bilddaten in einem Schritt erlaubt.
4. Im METK enthalten ist eine **automatische Synchronisation** zwischen allen Modulen einer Applikation. So können sich Änderungen in einem Applikationsteil automatisch auf andere Teile auswirken.
5. Das METK bietet über die **Keystates** (Kapitel 8) die Möglichkeit, verschiedene Status⁴⁰ einer Visualisierung zu speichern.

⁴⁰ „Das aus dem Lateinischen stammende Substantiv lautet im Genitiv und im Plural gleich wie im Nominativ Singular, also des Status und die Status. Ein Unterschied besteht lediglich in der Aussprache. Im Plural wird das u lang gesprochen.“ (DUDENREDAKTION [2009])

6. Zur Beschleunigung des Applikationsentwicklungsprozesses enthält das METK viele **vorgefertigte Steuerelemente**, wie Strukturlisten und Templates für Benutzeroberflächen.
7. Für die Entwicklung von Applikationen auf der Basis des METK sind nur einfache Kenntnisse der Skriptsprachenprogrammierung in PYTHON nötig.
8. Die **Quelloffenheit** und umfangreiche Dokumentation des METK ermöglichen die Erweiterung um neue Features.

Alle genannten Techniken sind nicht Teil der Entwicklungsplattform MEVISLAB, auf der das METK aufbaut (siehe Abschnitt 9.2.2). Auch sind nur wenige der Techniken in aktuellen Toolkits enthalten.

9.1.3 Konzeptuelle Vorüberlegungen

Ein Problem in der computergestützten Chirurgieplanung, wenn nicht gar das Kernproblem überhaupt, ist das Verständnis von Anforderungen, Erfordernissen und Prioritäten und deren Zusammenspiel. Es hat sich herausgestellt, dass ein schnelles Prototyping von Applikationen und die frühe Diskussion erster Vorschläge und Ideen im Rahmen eines komplexen Entwicklungsprozesses von essenzieller Bedeutung sind. Chirurgische Abteilungen sind zu klein, als dass sie eigene IT-Gruppen unterhalten könnten. Hinzu kommt, dass einzelne neue Funktionen in Applikationen oft schwer verbal zu fassen und zu vermitteln sind. Die frühe Verfügbarkeit von Applikationen zur Demonstration ermöglicht daher Diskussionen mit den späteren Anwendern, den Chirurgen, und führt zu substanziellen Erkenntnissen schon in einem frühen Entwicklungsstadium. Die schnelle Verfügbarkeit solcher Prototypen war daher eine der zentralen Anforderungen bei der Entwicklung des METK.

Ein zweiter wichtiger Aspekt der Entwicklung war die Wiederverwendbarkeit einmal entwickelter Visualisierungs- und Interaktionstechniken innerhalb einer großen Arbeitsgruppe. Die Verfügbarkeit eines breiten Grundstocks an grundlegenden sowie fortgeschrittenen Techniken erlaubt es dem einzelnen Forscher oder Studenten, sich auf die Entwicklung neuer Techniken und Applikationen zu konzentrieren. Dieser Ansatz erfordert zwar zu Beginn einen größeren Implementierungsaufwand, zahlt sich aber später aus. Während die Kernentwicklung durch zwei Forscher aus zwei unterschiedlichen Projekten erfolgte (den Autor sowie Christian Tietjen), sorgte die intensive Anwendung des METK durch eine breite Nutzergruppe für eine Verfeinerung der Anforderungen und Techniken.

Für ein Toolkit zur Applikationsentwicklung ist ein modularer Charakter softwaretechnisch essenziell. So kann jede Applikation aus den wirklich nur benötigten Modulen aufgebaut werden. Weiterhin ist aus softwaretechnischer Sicht die Erweiterbarkeit ein wichtiges Kriterium, da so erst die Anpassung neuer Applikationen an in Teilen sehr spezifische chirurgische Anforderungen möglich ist. Um eine Erweiterbarkeit zu garantieren, sind eine offene Kommunikationsstruktur

sowie ein zentrales Datenmanagement maßgeblich. So können neu geschaffene Module sofort mit allen bereits existierenden Modulen über eine Schnittstelle auf einer gemeinsamen Datenbasis kommunizieren.

Aus diesen Überlegungen lassen sich drei Kernanforderungen an die Entwicklung des METK ableiten:

1. Sofort nutzbare Applikationen sollten schnell mit dem Toolkit erstellt werden können. Dadurch kann wichtiges Feedback schon in einem frühen Stadium der Entwicklung erlangt werden.
2. Der Entwicklungsprozess sollte keine größeren Programmierkenntnisse, wie die Erstellung vieler Klassen für die Applikationslogik, erfordern.
3. Existierende wie neue Applikationen sollten um neue Visualisierungs- und Interaktionstechniken erweiterbar sein, um auf spezielle Anforderungen der Chirurgen wie auch neue Forschungsentwicklungen reagieren zu können.

Nach den Anforderungen aus Entwicklersicht werden im Folgenden Anforderungen an chirurgische Applikationen aus der Sicht der späteren Nutzer definiert.

9.1.4 Anforderungen an chirurgische Applikationen

Die im Folgenden analysierten Anforderungen an chirurgische Applikationen basieren auf den chirurgischen Szenarien aus Kapitel 2 sowie mannigfaltigen Erfahrungen auf dem Gebiet der Applikationsentwicklung, speziell für Chirurgen. Chirurgen sind Experten ihres Faches und haben nur selten herausragende Computerkenntnisse oder Erfahrungen in der 3D-Exploration von Patientendaten. Der Fokus eines Chirurgen liegt auf der Vorbereitung und Durchführung operativer und interventioneller Eingriffe. Im Gegensatz bspw. zum Radiologen verbringt der Chirurg nur wenige Stunden pro Woche am Rechner und ist dann zumeist mit administrativen Aufgaben beschäftigt. Wird der Computer zur Operationsplanung genutzt, so geschieht dies im klinischen Alltag unter enormem Zeitdruck. Daher muss die Exploration, aber auch Präsentationsaufbereitung von für die Operationsentscheidung wichtigen Daten zügig erfolgen. Applikationen zur chirurgischen Operationsplanung und zur Unterstützung der Entscheidungsfindung sollten daher folgenden Eigenschaften aufweisen:

1. Sie sollten zunächst den Entscheidungsfindungsprozess unterstützen.
2. Möglichkeiten zur Vermessung (Distanzen, Volumina, Winkel) sollten vorhanden sein, da solche Maße oftmals eng mit der Therapieentscheidung verknüpft sind.
3. Aufgrund der Wichtigkeit der 2D-Schichtbilddaten sollten 2D- und 3D-Viewer kohärent und synchronisiert sein. Die Exploration von 3D-Daten muss besonders unterstützt werden.

4. Wichtige anatomische und pathologische Strukturen müssen kontexterhaltend hervorgehoben werden.
5. Spezielle Techniken für einzelne chirurgische Felder (z.B. Resektionstechniken für die Abdominalchirurgie oder multimodale Visualisierungen für die Kardiochirurgie) sollten verfügbar sein.
6. Für die Anwendungen ist es nicht nur wichtig, dass sie einfach zu nutzen sind, sondern auch, dass sie durch ihre Darstellung und ihren Aufbau eine Nutzung motivieren („*Joy-of-use*“). Da Chirurgen aus dem OP-Umfeld Anwendungen mit ansprechenden Oberflächen gewohnt sind, erwarten sie dies auch von Applikationen zur Operationsplanung und Forschungsprototypen (siehe dazu auch Abschnitt 10.4).

Eine Anforderung, bei der sich die Interessen von Chirurgen und Entwicklern decken, ist die an ein effizientes Fallmanagement einer Applikation. Dabei wird hier als *Fall* die Menge an patientenrelevanten Daten betrachtet, zuvorderst ein- oder mehrere Bilddatensätze mitsamt Segmentierungen, Strukturinformationen sowie Patientendaten, wie Alter, Größe oder Laborwerte. Diese Informationen sind notwendig, um dem Chirurgen patientenindividuell angepasste Visualisierungen anbieten zu können und so den Entscheidungsfindungsprozess optimal zu unterstützen.

9.2 Verwandte Toolkits und Frameworks

Da die medizinische Visualisierung und die medizinische Bildanalyse sehr aktive Forschungsfelder sind, gibt es eine ganze Reihe entsprechender Toolkits. Die relevantesten werden im Folgenden vorgestellt, wobei aufgrund seiner Bedeutung für das METK MEVISLAB ausführlicher betrachtet wird (siehe Abschnitt 9.2.2). Unberücksichtigt bleiben Arbeiten, die den einzelnen Techniken und Funktionen des METK verwandt erscheinen mögen (z.B. Arbeiten zum illustrativen Rendering). Dabei sei auf die zumeist parallel zur Entwicklung des METK entstandenen Publikationen verwiesen, die im methodischen Teil dieses Kapitels (Abschnitt 9.3) Erwähnung finden.

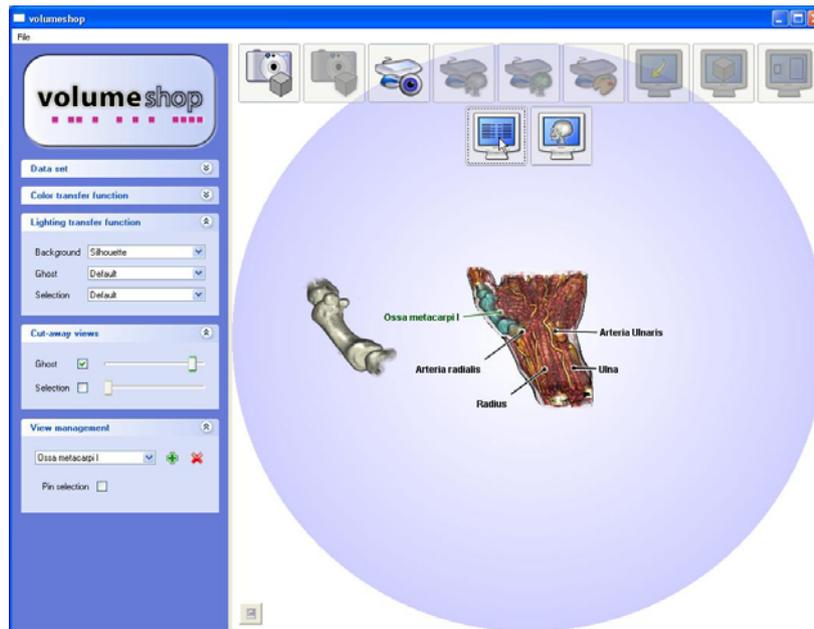


Abbildung 9.3: Die Illustrationsapplikation Volumeshop von Bruckner & Gröller [2005]. Die Applikation dient vornehmlich der Demonstration neuer Techniken des Vollerendings und der Erstellung einzelner ästhetisch anspruchsvoller Illustrationen (Quelle: <http://www.cg.tuwien.ac.at/volumeshop>).

Skripte überführt, um so eine Visualisierung zu erzeugen. Nichtsdestotrotz unterstützen die beiden zuletzt genannten Systeme von BRUCKNER & GRÖLLER [2005] und RÖSSLER u. a. [2008] keine Erstellung eigener Applikationen.

Mit dem 3DSLICER von GERING u. a. [1999] gibt es eine quelloffene Entwicklungsumgebung für eine Reihe von Forschungsanwendungen, vor allem jedoch für die neurochirurgische Operationsplanung. Der 3DSLICER ist auf die Darstellung von 2D-Schichtbilddaten spezialisiert und bietet nur eine kleine Auswahl einfacher 3D-Visualisierungstechniken. Jedoch war der 3DSLICER bei seiner Veröffentlichung eine der ersten Anwendungen, die 2D- und 3D-Darstellungen konsequent kombinierten.

Das Medical Imaging Interaction Toolkit (MITK) von WOLF u. a. [2005] ist ein C++-Framework auf der Basis von ITK und VTK. Das MITK unterstützt vor allem die Bildanalyse und Interaktionen zur Segmentierung und Registrierung medizinischer Bilddaten. Die Entwickler betonen ausdrücklich, mit dem MITK kein Framework zur Applikationsentwicklung anzubieten.

Julius (JANSEN u. a. [2001]) ist ein Framework, welches aus einer Kernapplikation besteht, die durch ein Plug-In-System erweitert werden kann. Es konzentriert sich dabei auf die medizinische Bildanalyse sowie verformbare Modelle und bietet Algorithmen zur Segmentierung, Registrierung und intraoperativen Navigation an. Für Visualisierungen wird VTK genutzt, wobei darüber keine speziellen Visualisierungstechniken hinaus für chirurgische Anwendungen angeboten werden.

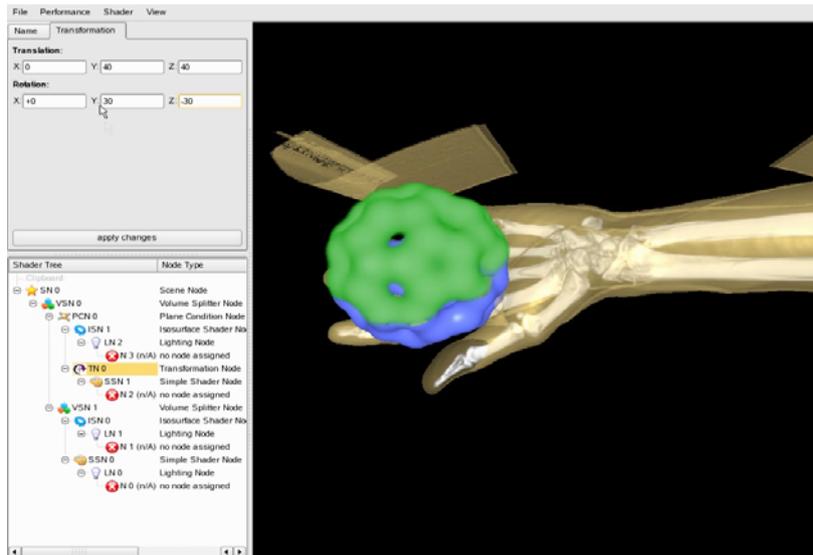


Abbildung 9.4: System zur Erzeugung Shader-basierter Visualisierungen von Rößler u. a. [2008]. Das System nutzt einen Rendergraphen (links), der vom Nutzer interaktiv manipuliert werden kann (Quelle: <http://www.vis.uni-stuttgart.de/eng/research/fields/current/multivol>).

Ein C++-Toolkit, welches sich auf die intraoperative Unterstützung konzentriert, ist das Image Guided Surgery Toolkit (IGSTK) (ENQUOBAHRIE u. a. [2007]). Es unterstützt die Entwicklung von Applikationen der interventionellen Radiologie und bildgestützten Chirurgie, bei denen getrackte Instrumente zum Einsatz kommen.

Ein geschlossenes (nicht quelloffenes) Framework zur Visualisierung multimodaler Daten wurde von MANSSOUR u. a. [2001] präsentiert. Sie konzentrierten sich bei der Entwicklung auf die Segmentierung und Registrierung, konnten jedoch nicht darlegen, wie individuelle Applikationen mit ihrem Framework erstellt werden könnten.

Ein weiteres Framework zur medizinischen Bildanalyse ist CAVASS (GREVERA u. a. [2007]). Hier wurden ein einfaches Volumerendering sowie Oberflächenvisualisierungen angeboten. Allerdings können mit CAVASS keine eigenen Applikationen entwickelt werden, da es sich um ein geschlossenes System handelt, das lediglich über eine quelloffene Schnittstelle erweitert werden kann.

Ein neueres System stellten BOTHA & POST [2008] mit DEVIDE vor (Abb. 9.5). Dabei handelt es sich um eine Entwicklungsumgebung für das schnelle Prototyping von neuen Visualisierungstechniken, wobei besonders auf die dynamische Anpassung des Codes und der zugrunde liegenden Datenflussnetzwerke zur Laufzeit Wert gelegt wurde. Die Entwicklung neuer Visualisierungs- und Segmentierungstechniken wird durch DEVIDE sehr gut unterstützt.

SCIRUN (WEINSTEIN u. a. [2005], www.scirun.org) wurde als System zur Problemlösung eingeführt. Es basiert auf Datenflussnetzwerken (Abb. 9.6) aus ein-

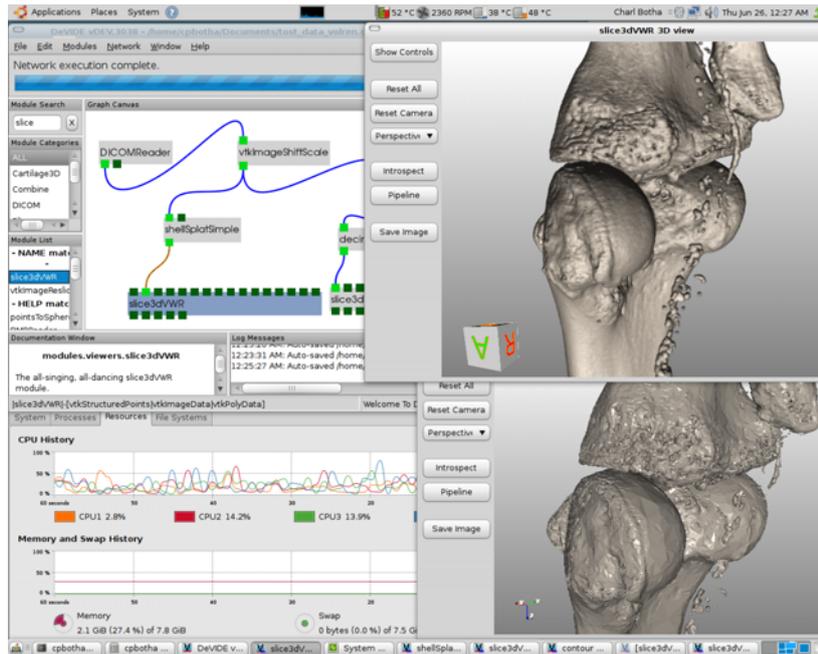


Abbildung 9.5: Das *DeVide*-System von Botha & Post [2008] ist auf die schnelle Entwicklung neuer Visualisierungstechniken spezialisiert (Quelle: <http://graphics.tudelft.nl/Projects/DeVIDE/Gallery>).

zelenen Modulen, z.B. zur Vektorfeldvisualisierung, Simulation oder Bildverarbeitung. Es können eigene Applikationen (sogenannte PowerApps) erstellt werden. Deren Entwicklung erfolgt in einer eigenen Skriptsprache, vergleichbar der *Module Definition Language* von MEVISLAB (siehe Abschnitt 9.2.2), die auf die Generierung konsistenter Oberflächen oberhalb der einzelnen Module abzielt.

Amira (STALLING u. a. [2005]) ist ein kommerzielles Toolkit zur Erzeugung wissenschaftlicher Visualisierungen. Es ist erweiterbar und bietet eine große Zahl von Techniken zur Visualisierung, Analyse und Simulation an. Amira bedient sich eines Ansatzes zur visuellen Programmierung, bei dem die Datenfluss-Metapher genutzt wird, um Module zu einem Netzwerk zu verknüpfen. Obwohl Möglichkeiten angeboten werden, einzelne Benutzeroberflächen zu erstellen, können dennoch keine eigenen Applikationen mit Amira erstellt werden.

9.2.2 Die Entwicklungsumgebung MeVisLab

MEVISLAB (REXILIUS u. a. [2006], www.mevislab.de) ist eine sowohl frei verfügbare wie auch kommerziell erhältliche Plattform zur Erstellung von Prototypen, basierend auf einem Ansatz der visuellen Programmierung. Der Fokus von MEVISLAB liegt auf der medizinischen Bildverarbeitung und Visualisierung (siehe Abbildung 9.7 für eine Beispielansicht). In der kommerziellen Version bietet der Anbieter von MEVISLAB, MeVis Medical Solutions, einen erweiterten Support, während die freie Version auf eine Nutzung im privaten und akademischen

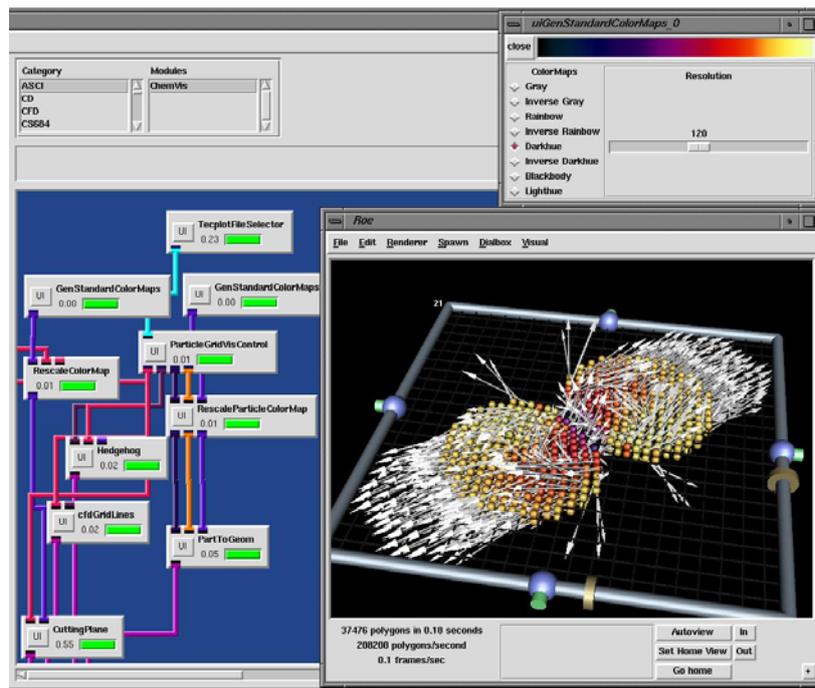


Abbildung 9.6: Das Software-Framework *SciRun* von [Weinstein u. a. \[2005\]](#) erlaubt die Erstellung eigener Applikationen (Quelle: www.scirun.org).

Bereich beschränkt bleibt. Eigene Applikationen können auf der Basis von MEVISLAB plattformübergreifend⁴¹ erstellt werden.

Neben grundlegenden Bildverarbeitungs- und Visualisierungsalgorithmen bietet MEVISLAB fortgeschrittene Algorithmen zur Segmentierung, Registrierung und quantitativen funktionalen Analyse medizinischer Bilddaten. Neue Algorithmen können in Form eigener Module auf der Basis einer einheitlichen Schnittstelle integriert werden. Sogenannte Makromodule erlauben als eine hierarchische Kapselung einzelner Netzwerke die Wiederverwendung bestimmter Teilnetzwerke. Die Definition von Benutzeroberflächen wird durch eine eigene, abstrakte und hierarchische Skriptsprache, die Module Definition Language (MDL), erreicht. Die Funktionalitätssteuerung sowohl der Netzwerke wie auch der Benutzeroberflächen erfolgt über die Skriptsprache PYTHON.

Ein integraler Bestandteil MEVISLABs ist die objektorientierte MeVis Image Processing Library (ML), die ein allgemeines Framework zur Bildverarbeitung anbietet. Die Bearbeitung der Bilddaten erfolgt anfragenbasiert, wobei ausgefeilte Techniken zum *Paging*,⁴² Caching und Multi-Threading zum Einsatz kommen.

⁴¹Derzeit sind Versionen für Windows, Linux und Mac erhältlich.

⁴²Beim *Paging* handelt es sich um eine Technik, bei der umfangreiche Schichtbildvolumen nicht in einer Gesamtheit, sondern in kleinen Teilen (sogenannten Pages) auf Anfrage in den Speicher geladen werden. So müssen immer nur die aktuell für eine Ansicht oder Bildverarbeitungsoperation notwendigen Pages im Speicher vorgehalten werden, was die Anforderung an dessen Menge erheblich reduziert.

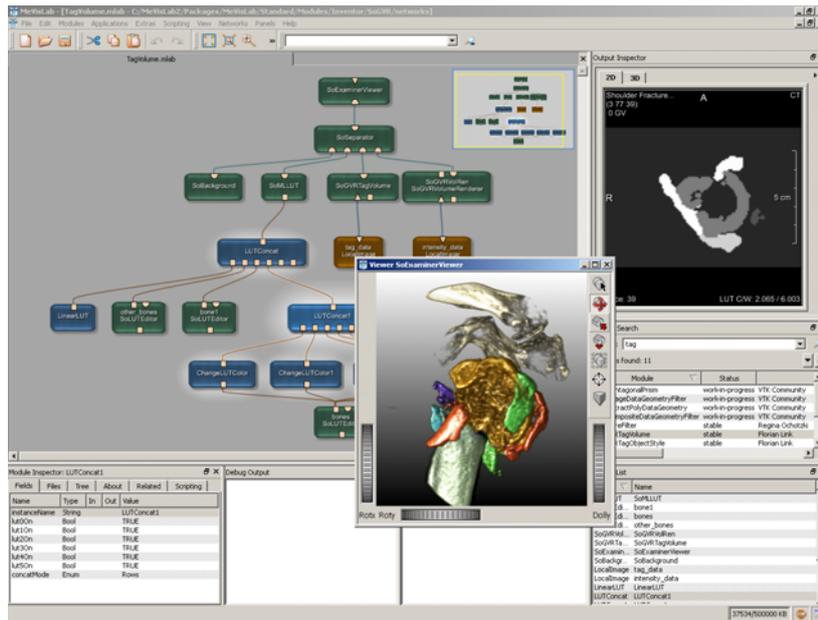


Abbildung 9.7: Screenshot von MeVisLab. Es sind ein Netzwerk sowie einige Panel für Einstellungen und Viewer zu sehen.

Weiterhin wurde das Insight Toolkit (ITK) für spezielle Segmentierungs- und Registrierungstechniken in MEVISLAB integriert.

Zur Visualisierung steht der MeVis GIGAVOXELRENDERER (GVR) zur Verfügung (LINK u. a. [2006]), welcher relativ neue Volumerendingstechniken, wie texturbasierte Multi-Resolution Ansätze und ausgefeiltes Objekt-Shading, bietet. Für oberflächenbasierte Visualisierungen steht die OPENINVENTOR Bibliothek sowie das Visualization Toolkit (VTK) zur Verfügung.

9.2.3 Diskussion und Einordnung existierender Lösungen

Zusammenfassend lässt sich feststellen, dass es viele Toolkits und Frameworks zur medizinischen Bildanalyse und Visualisierung gibt. Jedoch konzentrieren sich einige sehr auf die Generierung einzelner beeindruckender Visualisierungen, z.B. VOLUMESHOP (BRUCKNER & GRÖLLER [2005]), einige auf die medizinische Bildanalyse, z.B. das MITK (WOLF u. a. [2005]) und der 3DSLICER (GERING u. a. [1999]), und nur wenige unterstützen die Entwicklung eigener Applikationen, z.B. SCIRUN (WEINSTEIN u. a. [2005] und MEVISLAB (REXILIUS u. a. [2006])).

MEVISLAB, Amira und SCIRUN weisen viele Gemeinsamkeiten auf. BITTER u. a. [2007] haben eine Gegenüberstellung dieser Systeme sowie zusätzlich dem MITK erstellt. Alle drei bieten die Möglichkeit der visuellen Programmierung. Amira und MEVISLAB nutzen Open Inventor zur 3D-Visualisierung. Der starke Fokus von MEVISLAB auf die medizinische Bildverarbeitung, die quantitative Bilddatenanalyse sowie die Möglichkeit der Applikationsentwicklung ist hervorzuheben.

Dennoch muss festgestellt werden, dass es keine dezidierte Unterstützung zur Nutzung umfangreicher Fälle gibt. Auch wenn alle Funktionen und Techniken des METK mit MEVISLAB entwickelt werden können (was mit dem METK geschehen ist), so bietet MEVISLAB selbst keine dieser High-Level-Module. MEVISLAB konzentriert sich an dieser Stelle auf die Entwicklung einzelner Algorithmen. Durch den Einsatz des METK kann ein Entwickler medizinischer Applikationen wesentlich schneller und effizienter lauffähige Prototypen erzeugen.

9.3 Architektur des METK

In diesem Abschnitt wird die Architektur des METK vorgestellt und es werden ausgewählte Funktionen und Techniken erläutert. Die im Rahmen dieser Arbeit entwickelten Techniken, wie die Wahl guter Sichtpunkte oder die Annotationen, sind Bestandteil des METK. Zusätzlich wurden eine Reihe weiterer Visualisierungs- und Interaktionstechniken im METK integriert, die andernorts umfänglich publiziert wurden. In solchen Fällen wird ebenfalls auf eine genauere Beschreibung der Techniken verzichtet und jeweils an gegebener Stelle auf die Publikationen verwiesen.

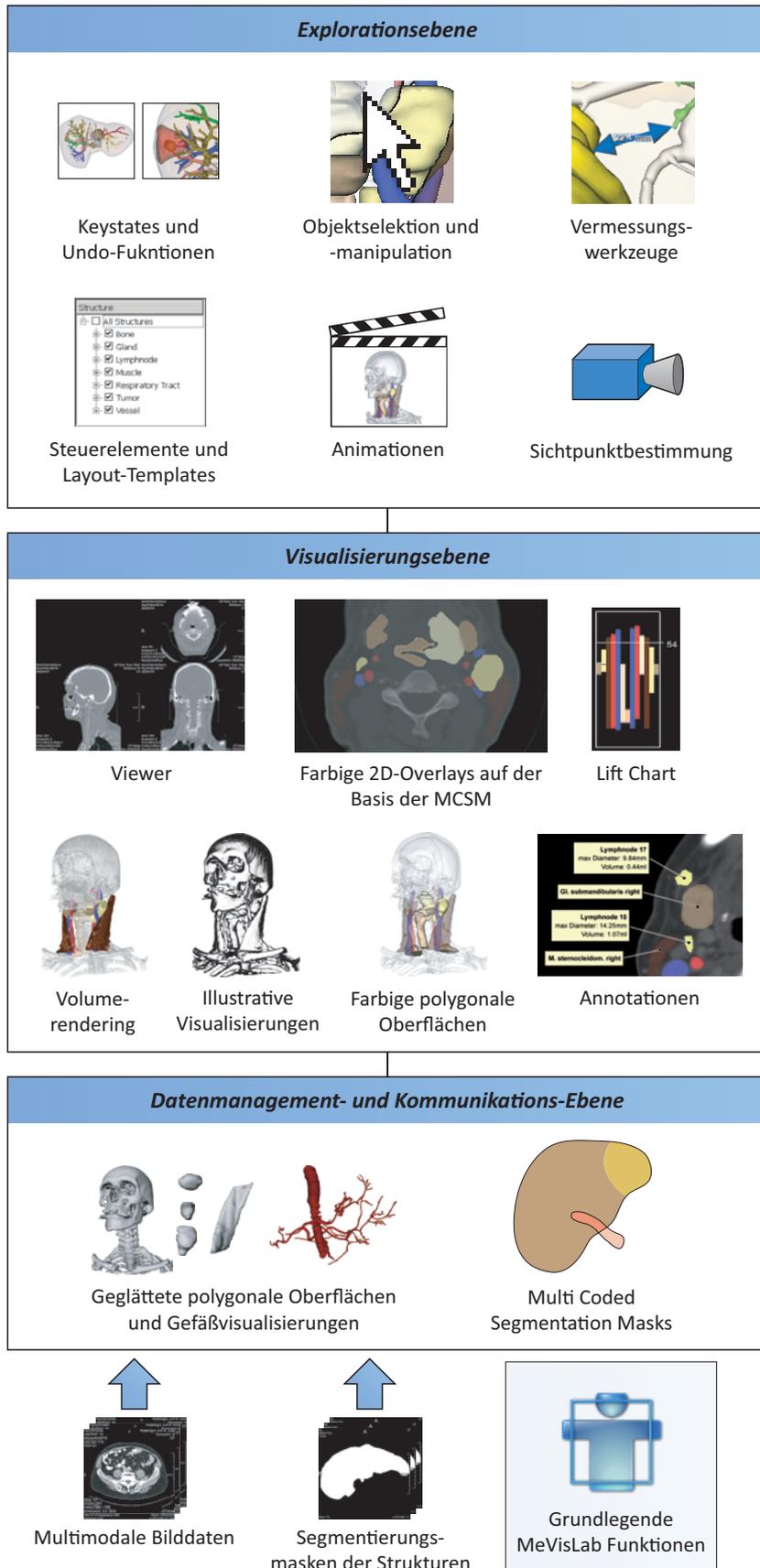
Im METK ist jede Funktion in einem Modul in MEVISLAB gekapselt. Durch die netzwerkbasierte visuelle Programmierung in MEVISLAB können so Applikationen erstellt werden, die genau den gewünschten Funktionsumfang haben. Folglich kann gezielt der spezifische Arbeitsablauf für eine bestimmte Art der Operationsplanung unterstützt werden.

Die Funktionen sind in drei Ebenen organisiert (siehe auch Abb. 9.8):

1. Datenmanagement- und Kommunikations-Ebene
2. Visualisierungsebene
3. Explorationsebene

Die erste Ebene ist für den Import der Daten zuständig und bietet Funktionen der Datenverwaltung für alle übrigen Funktionsteile an. Die Visualisierungsebene umfasst verschiedene Viewer und spezielle Renderingtechniken. Alle High-End-Techniken der Interaktion und Exploration zur Erstellung umfänglicher chirurgischer Applikationen sind in der Explorationsebene enthalten. Die Ebenen sowie die enthaltenen Funktionen werden im Folgenden genauer beschrieben.

Abbildung 9.8: Architektur des METK. Die Module des METK sind in drei Ebenen organisiert und können frei zu Applikationen mit einem individuellen Funktionsprofil kombiniert werden.



9.3.1 Datenmanagement- und Kommunikations-Ebene

Die Ebene des Datenmanagements und der Kommunikation enthält Funktionen des Fallmanagements, der intermodularen Kommunikation und zeichnet für die Erzeugung der polygonalen 3D-Oberflächen verantwortlich. Da DICOM der am weitesten verbreitete Standard für medizinische Bilddaten ist, konzentriert sich das METK auf die Verarbeitung von DICOM-Daten. Zusätzlich können Segmentierungsmasken verarbeitet werden, um daraus automatisch polygonale 3D-Oberflächen für verschiedenste Visualisierungen zu erzeugen. Dies muss nur beim ersten Laden eines Falles erfolgen, da die 3D-Netze für spätere Verwendungen gespeichert werden. Abhängig von der Art der Struktur kommen verschiedene Algorithmen zur Generierung der 3D-Oberflächen zum Einsatz. In den meisten Fällen kommt Marching Cubes in Verbindung mit einer Oberflächenglättung zur Anwendung. Für Gefäße wird eine modellbasierte Oberflächenrekonstruktion verwendet, die die dünne verzweigte Struktur der Gefäße berücksichtigt (SCHUMANN u. a. [2007]). Neben den Segmentierungsmasken können auch existierende polygonale Netze von Strukturen oder anderen Objekten, wie medizinischen Instrumenten, mit dem METK importiert werden.

Multimodale Daten: Wie auch das zugrunde liegende MeViSLAB erlaubt das METK die Kombination verschiedener Modalitäten, wie z.B. CT, MRT oder PET. Sind die Daten registriert, ist mit dem METK eine integrierte Visualisierung, bspw. von CT- und PET-Daten, möglich. Die Registrierung wird dabei von MeViSLAB selbst unterstützt. Es stehen spezielle Module zur manuellen, starren (regiden) und nicht-regiden Registrierung auf Basis von Landmarken und intensitätsbasierten Ähnlichkeitsmaßen zur Verfügung.

Fall- und Speichermanagement: Zur Reduzierung des Speicherbedarfs und zur Beschleunigung des gesamten Lade- und Explorationsprozesses wurde ein effizientes Speichermanagement im METK integriert. Jede Struktur und jeder Bilddatensatz wird nur einmal geladen und virtuell innerhalb einer Applikation verteilt. Selbst wenn eine Struktur in verschiedenen Sichten mit unterschiedlichen Techniken visualisiert wird, so wird sie nur einmal im Applikationscache vorgehalten.

Kommunikation: Neben dem Datenmanagement bietet das METK in seinem Kern eine Kommunikationsstruktur zwischen allen Modulen einer Applikation. Zur direkten Kommunikation zwischen einzelnen Modulen können Nachrichten und Ereignisse zwischen ihnen gesendet werden. Verteilte Nachrichten können verschickt werden, um viele Module auf einmal zu erreichen, wobei der Sender die Art und die Anzahl der Empfänger nicht kennen muss. Jegliche Veränderung an den Daten und den Parametern einer Applikation (z.B. die Farbe einer Struktur oder ein Interaktionsstatus) werden automatisch an die Module weitergeleitet, die über solche Änderungen informiert werden möchten, um ggf. selbst darauf zu reagieren. Dies führt (so gewünscht) zu einem identischen Erscheinungsbild einer Struktur in allen Teilen einer Applikation und somit zu einer konsistenten Sicht auf die Daten.

Synchronisation: Weiterhin wird die aktuell selektierte Struktur⁴³ automatisch allen Modulen des METK bekannt gemacht. Dadurch können verschiedene Viewer synchronisiert werden. Selektiert der Nutzer eine Struktur in einem 3D-Viewer, kann der 2D-Viewer automatisch die entsprechende Schicht zur Struktur anzeigen. Wählt der Nutzer in einem 2D-Viewer eine Struktur, kann sie in allen 3D-Viewern hervorgehoben werden, indem die Kamera zu einem guten Sichtpunkt auf die Struktur bewegt wird. Außerdem können verschiedene 3D-Viewer untereinander synchronisiert werden, indem ihre Kameraparameter, wie Position und Orientierung, über das METK verbunden werden. Dies ermöglicht die Exploration verschiedener Datensätze vom gleichen Sichtpunkt aus, beispielsweise um prä- und post-operative Daten ein und desselben Patienten miteinander zu vergleichen.

Multi-Coded-Segmentations: Im Normalfall (so auch bei MEVISLAB) wird jede Segmentierungsmaske in einer eigenen Datei gespeichert. Das ist unter Speicher- wie unter Performanzgesichtspunkten ineffizient. Für jede Struktur muss eine eigene Segmentierungsmaske in den Speicher (nach)geladen werden. Sollen mehrere Strukturen in einer Schicht dargestellt werden, erfordert dies das Laden aller Segmentierungsmasken. Dies ist speicher- und zeitaufwendig. Die Speicherung aller Segmentierungsmasken in einem einzigen Bildstapel kann dieses Problem beheben. Jedoch ist zu bedenken, dass ein Voxel zu mehr als einer Struktur gehören kann, z.B. wenn sich Strukturen überlappen. So kann zum Beispiel ein Voxel gleichzeitig dem Lebergewebe und dem Tumor innerhalb zugeordnet werden. Daher kann mitunter nicht jedem Voxel einer Struktur dieselbe ID zugeordnet werden. Ein naheliegender Ansatz ist es, jeder Struktur ein Bit eines 8-Byte-Voxelwertes zuzuweisen. Dieser Ansatz ist jedoch auf Datensätze mit wenigen Strukturen begrenzt, da zum Beispiel nur 64 verschiedene Strukturen in einem 8-Byte-Wert (=64bit) gespeichert werden könnten.⁴⁴ Da aber in realen Daten nur ein sehr geringer Teil aller möglichen Überlappungen auftritt, wurde im Rahmen des METK eine effiziente dynamische Datenstruktur entwickelt, die „Multi-Coded-Segmentation-Masks“ (MCSM).

Eine MCSM enthält alle Segmentierungsmasken aller Strukturen eines Datensatzes. Jede Kombination von Strukturen (Überlappung) in einem Voxel wird durch eine eindeutige ID repräsentiert (Abb. 9.9). So erhalten alle Voxel, die gleichzeitig zum Lebergewebe *und* der Lebervene gehören (und zu keiner anderen Struktur), die gleiche ID zugewiesen. Die Abbildung der Voxel-IDs auf die verschiedenen Strukturkombinationen wird separat in einer Liste verwaltet. Eine MCSM wird durch sequenzielles Hinzufügen der einzelnen Segmentierungsmasken zur MCSM erzeugt. Tritt eine neue Kombination von Strukturen an einem Voxel auf, so erhält diese eine neue ID.

⁴³Dies kann durch die Auswahl in einer Liste oder durch ein Anwählen der Struktur in einem Viewer geschehen.

⁴⁴Um Überlagerungen von Strukturen abzudecken, können nicht, wie eventuell erwartet, 2^{64} Werte genutzt werden. Vielmehr müsste jeder Struktur eindeutig ein Bit zugeordnet werden.

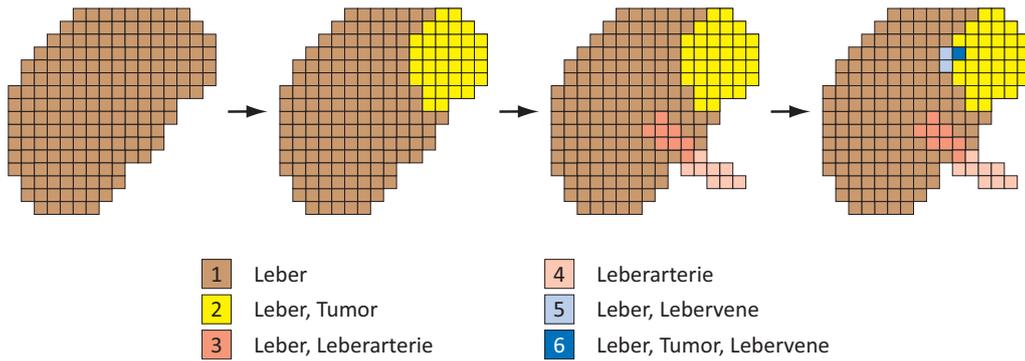


Abbildung 9.9: Die Multi-Coded-Segmentation-Masks. Die Segmentierungsmasken der einzelnen Strukturen werden sequenziell zur Multi-Coded-Segmentation-Mask (MCSM) hinzugefügt. Jede Überlappung von Strukturen in einem Voxel erhält eine eigene ID. Die Zuordnung von IDs zu den Strukturen wird separat gespeichert. So können viele sich überlappende Segmentierungsmasken in einem Datensatz gespeichert werden. Für die vier Strukturen in der Abbildung treten nur 6 von 16 möglichen Kombinationen auf.

Fall	Anzahl der segmentierten Strukturen	Anzahl der benötigten IDs in der MCSM
HLA 1	36	61
HLA 2	21	36
Tumorresektion der Leber	49	361
Leberlebenspende	81	656

Tabelle 9.1: Anzahl der IDs in einer MCSM für verschiedene Fälle. Die Tabelle zeigt eine Übersicht über die verwendeten IDs im Vergleich zur Anzahl der Strukturen in zwei Fällen der Halslymphknotenausräumung (HLA) sowie zwei Fällen der Leberchirurgie. Die größere Anzahl an IDs bei den Leberfällen resultiert aus der größeren Anzahl an Überlappungen innerhalb des Leberparenchyms.

Neben der effizienten Speicherung vieler Segmentierungsmasken sind die MCSM eine geeignete Grundlage für farbige Overlays. Die obere Grenze für die Anzahl von IDs, die mit einer MCSM gespeichert werden können, liegt bei 2^{64} , wenn man für die Voxelwerte einen 8-Byte-Integer annimmt. Diese wird allerdings bei medizinischen Datensätzen nie erreicht, weil selbst im theoretischen Fall bei einer Datensatzgröße von 512^3 Voxeln jeder Voxel eine andere Strukturkombination repräsentieren könnte. In Tabelle 9.1 wurde für die Szenarien der HNO- und Leberchirurgie die Anzahl segmentierter Strukturen sowie die Anzahl der IDs in der MCSM gegenübergestellt. Für die 81 Strukturen im Fall der Leberlebenspende müssen mit der MCSM nur 11 Bits anstelle der 81 Bits genutzt werden, würde man jeder Struktur ein eigenes Bit zuweisen.

9.3.2 Visualisierungsebene

Die Visualisierungsebene bündelt alle Funktionalitäten, die für einfache und erweiterte Visualisierungstechniken benötigt werden, darunter die in Kapitel 7 vorgestellten Annotationstechniken. Jeder im Cache gespeicherten Oberfläche einer Struktur wird Material zugewiesen. Wichtige Strukturen können dabei mit einer hohen Opazität dargestellt werden, während Kontextstrukturen, wie Knochen oder große Organe, mit einer Silhouette visualisiert werden können.

Zur Darstellung von **Volumerenderings** nutzt das METK den GIGAVOXELRENDERER, wie er in MEVISLAB eingebaut ist (LINK u. a. [2006]). Dieser ermöglicht das „tagged volume rendering“ segmentierter Strukturen. Dabei kann jeder Struktur eine eigene lokale Transferfunktion zugewiesen werden. Zur Erhaltung der Konsistenz werden Strukturen im Volumerendering mit der gleichen Farbe wie in ihrer Oberflächendarstellung visualisiert. Zur Darstellung unsegmentierter Bereiche wird eine globale Transferfunktion genutzt. Das Volumerendering kann mit Oberflächenvisualisierungen kombiniert werden.

Erweiterte 2D-Visualisierungstechniken: Das grundlegende Problem schichtbasierter Visualisierungen ist der fehlende Überblick über die Ausdehnung und Lage von Strukturen. Aus diesem Grund wurde im METK ein 2,5D-Ansatz integriert, der zusätzliche Informationen liefert: der LIFTCHART (TIETJEN u. a. [2006b]). Mithilfe dieser Technik kann die Ausdehnung von Strukturen über mehrere Schichten schnell erfasst werden. Ein schmaler Rahmen, der an der Seite eines Schichtbildes eingeblendet wird, entspricht dabei der kompletten vertikalen Ausdehnung des Datensatzes (Abb. 9.10(a)). Jede segmentierte Struktur wird in Form eines kleinen Balkens innerhalb des Rahmens entsprechend ihrer Lage und Ausdehnung über die Schichten dargestellt. Es sind verschiedene Anordnungen der Balken möglich: So können alle Strukturen eines Typs in einer Spalte zusammengefasst oder nur wichtige Strukturen (z.B. Pathologien) angezeigt werden. Die aktuelle Schicht wird durch eine beschriftete Linie innerhalb des Rahmens visualisiert (Abb. 9.10(a)).

Um die mentale Abbildung der Strukturen von den 3D-Visualisierungen auf die 2D-Schichtbilder und umgekehrt zu erleichtern, können segmentierte Strukturen als farbige halbtransparente⁴⁵ Overlays in den Schichtbildern angezeigt werden. Überlappen oder infiltrieren sich Strukturen, so müssen an einer Voxelposition die Overlays mehrerer Strukturen angezeigt werden. Die Farbe an solchen Stellen kann im METK nach verschiedenen Farben bestimmt werden:

1. Nur die Farbe der wichtigeren Struktur wird dargestellt.⁴⁶

⁴⁵Dies garantiert die weitere Sichtbarkeit der Grauwerte im Schichtbild.

⁴⁶Die Wichtigkeit wird aus den Metadaten der Fallbeschreibung gewonnen oder, wenn diese Daten nicht existieren, indirekt aus der Größe der Struktur abgeleitet. In der chirurgischen Operationsplanung sind zumeist kleine Strukturen wichtiger als große.

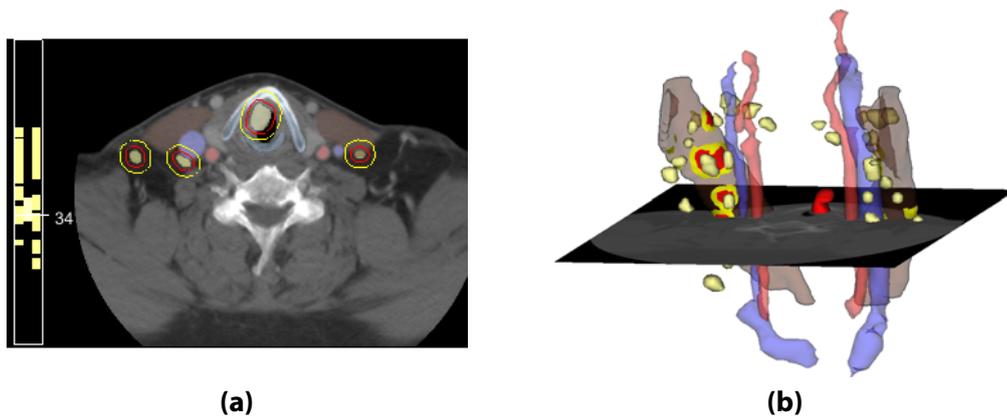


Abbildung 9.10: (a): Darstellung eines LiftChart in 2D sowie die korrespondierende 3D-Ansicht. Die Lage der Strukturen im Datensatz kann über die farbigen Balken des LiftChart identifiziert werden. Die Auswahl einer Struktur im LiftChart springt zu der entsprechenden Schicht im Datensatz. (b) In der 3D-Ansicht sind zusätzlich die Sicherheitsabstände der Lymphknoten auf der Oberfläche eines Muskels zu erkennen (aus [Mühler u. a. \[2010\]](#)).

2. Eine gewichtete Mischung der Farben wird berechnet, wobei die Opazität als Wichtungsfaktor dient.
3. Je nach Applikation kann eine Region mit Überlappungen gesondert hervorgehoben werden. So kann, wenn ein Lymphknoten einen Muskel infiltrierte, dieser kritische Bereich eine völlig neue Farbe zugewiesen bekommen (z.B. rot).

Die Berechnung der Overlays erfolgt auf Basis der Multi-Coded-Segmentation-Masks.

Sicherheitsränder um Tumore oder Metastasen stellen eine wesentliche Information während einer Operationsplanung dar. Daher wird im METK für alle Risikostrukturen eine 3D-Distanztransformation durchgeführt. Klar unterscheidbare Farben für verschiedene Distanzbereiche (z.B. gelb für 10mm und rot für 5mm) haben sich als angemessen herausgestellt. Die Distanzen können sowohl in 3D wie auch in 2D dargestellt werden. In der 2D-Sicht verdeutlichen Silhouetten die wichtigen Distanzen (Abb. 9.10(a)). In 3D-Darstellungen kommen uniform kolorierte Bereiche zur Verdeutlichung der kritischen Distanzbereiche zur Anwendung (Abb. 9.10b).

Illustrative Visualisierungen: Es ist offensichtlich, dass eine Visualisierung der Strukturen allein auf Basis von Farbe und Transparenzen für komplexe Zusammenhänge nicht hinreichend geeignet ist. Gerade die räumliche Dichte anatomischer Strukturen befördert den Einsatz illustrativer Techniken zur besseren Wahrnehmung von Formen und Relationen. Illustrative Techniken kommen in verschiedenen Bereichen der Operations- und Therapieplanung zum Einsatz: So

können Hatching-Linien die Formen und den Verlauf von Oberflächen besser verdeutlichen als herkömmliches Rendering ([INTERRANTE u. a. \[1997\]](#)). Die Nutzung von Silhouetten unterstützt die Wahrnehmung besonders transparenter Strukturen. Die illustrativen Techniken können im METK frei miteinander kombiniert werden. Der Gebrauch von Silhouetten ist der Standard im NECKSURGERYPLANNER, welcher in der klinischen Routine genutzt wird. Dennoch gibt es keinen Beleg dafür, dass illustrative Techniken einen signifikanten Vorteil in der Operationsplanung bringen.

Spezielle Viewer: Die Visualisierungsebene hält weiterhin spezielle Viewer bereit, die eine Erweiterung der in MEVISLAB enthaltenen 2D- und 3D-Viewer darstellen. Die Viewer sind in der Lage, ihre Parameter, z.B. die Kameraposition und -orientierung, untereinander zu synchronisieren oder von anderen METK-Modulen gesteuert zu werden. So können Viewer durch ein Animations-Modul ferngesteuert werden. Die 2D-Viewer können Schichtbilder auf vielfältige Art und Weise anzeigen: Einzeln, mehrere Schichten neben- und übereinander oder als freie multiplanare Reformation. 3D- und 2D-Viewer können frei und in beliebiger Anzahl miteinander in einer Applikation kombiniert werden.

9.3.3 Explorationsebene

Zur Unterstützung der Exploration werden verschiedene Interaktionstechniken und Steuerelemente im METK bereitgestellt. Ein großer Teil wird durch die in dieser Arbeit ausführlich beschriebenen Techniken der **Animation** (Kapitel 3), der **Sichtpunktbestimmung** (Kapitel 4), der **Kamerapfadplanung** (Kapitel 6) und der **Keystates** (Kapitel 8) gebildet. Es finden sich aber weitere ausgefeilte Techniken, die bisher nicht beschrieben wurden und daher hier näher erläutert werden.

Vermessungswerkzeuge dienen zum Beispiel der Entfernungsmessung zwischen zwei Strukturen oder Punkten in einer Darstellung. Die manuellen Werkzeuge werden um automatische Funktionen ergänzt, die zum Beispiel die minimalen Distanzen zwischen zwei Strukturen oder das Volumen und die Ausdehnung einer jeden Struktur berechnen.

Eine wichtige Funktion, gerade für in der 3D-Exploration unerfahrene Chirurgen, ist eine **Undo-Funktion** für die 3D-Darstellungen. Nach jeder Aktion des Nutzers in einem 3D-Viewer (z.B. einer Kamerabewegung) wird die gesamte Szene automatisch in einem Keystate gespeichert. Mehrere Änderungen in einer sehr kurzen Zeit werden dabei in einem Keystate zusammengefasst. Der Nutzer hat so die Möglichkeit, zu vorherigen Sichten oder Darstellungen schnell zurückzukehren.

Automatische Strukturselektion: Das METK bietet verschiedene Techniken zur Selektion von Strukturen in 3D-Darstellungen mit vielen sich überlappenden Strukturen verschiedener Transparenzen. Die Selektion in solchen Darstellungen ist nicht eindeutig, wenn mehrere Strukturen entlang des Selektionsstrahls

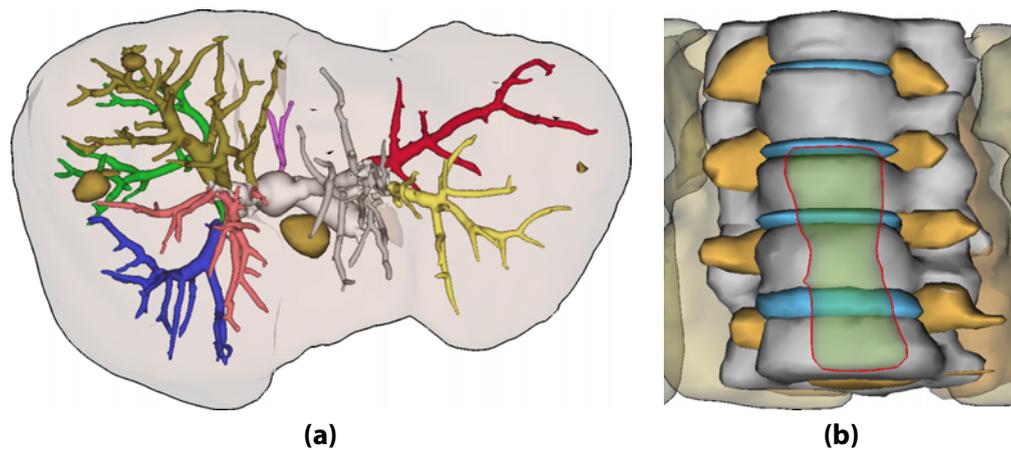


Abbildung 9.11: Objektselektion: Die Selektion der innerhalb des Leberparenchyms liegenden Strukturen in (a) wird durch den neuen Algorithmus zur Objektselektion ermöglicht, während der semi-transparente Oesophagus (grün) im Vordergrund der opaken Strukturen in (b) nicht automatisch selektiert werden kann (aus Mühler u. a. [2010]).

(engl.: *pick ray*) liegen. Der einfachste Ansatz ist die Wahl der ersten getroffenen Struktur. Jedoch kann es sein, dass der Nutzer eine dahinter liegende Struktur selektieren wollte, wenn die vordere sehr transparent dargestellt wird. In manchen Darstellungen sind einige Strukturen komplett von anderen umschlossen, z.B. die Gefäße innerhalb der Leber (Abb. 9.11(a)). Solche Strukturen könnten mit einem einfachen Ansatz niemals selektiert werden.

Daher wurde im Rahmen des METK ein Algorithmus entwickelt, der automatisch entscheidet, welche Struktur entlang des Selektionsstrahls der Nutzer möglicherweise selektieren möchte. Dazu wird angenommen, dass der Nutzer bewusst an eine bestimmte Stelle klickt. Das bedeutet, wenn der Nutzer den Mauscursor zugleich über einer sehr großen und einer sehr kleinen Struktur platziert hat, er sehr wahrscheinlich die kleinere Struktur selektieren wollte. Weiterhin wird angenommen, dass opake Strukturen weit stärker wahrgenommen werden als transparente. Zur Identifizierung der letztendlich zu selektierenden Struktur geht der Algorithmus wie folgt vor (siehe auch Algorithmus 9.1):

- Als erstes werden Strukturen entlang des Selektionsstrahls ermittelt und nach der Entfernung ihres Schnittpunktes mit dem Selektionsstrahl vom Betrachter sortiert.
- Durch die Ermittlung der Strahlabschwächung an jeder Struktur entsprechend ihrer Transparenz wird erreicht, dass nur Strukturen berücksichtigt werden, die zu mindestens 10% sichtbar sind.
- Als nächstes werden die Größen der projizierten Bounding Boxen der betroffenen Strukturen ermittelt.

- Für jede Struktur wird aus der Größe ihrer projizierten Bounding Box im Verhältnis zum Viewport und ihrer Transparenz ein Rating berechnet.
- Die Struktur mit dem größten Rating wird selektiert.

Durch dieses Verfahren können opake Strukturen selektiert werden, die hinter einer größeren transparenten Struktur liegen.

Eingabe : Alle Objekte $\{O\}$, die vom Selektionsstrahl getroffen werden.

Eingabe : Einfluss I_T der Objekt-Transparenz O_T

Eingabe : Einfluss I_S der Größe des Objektes O_S

Eingabe : Viewportgröße V_S

Ausgabe : Objekt O_{result} mit dem größten Rating

Sortiere die Objekte anhand der Entfernung ihres Schnittpunktes mit dem Selektionsstrahl vom Betrachter;

$A_{ray} \leftarrow 1.0$; // Strahlabschwächung

$R \leftarrow 0$; // Initialisierung des später größten Ratings

für jedes $O_i \in \{O\}$ **tue**

$O_S \leftarrow$ Berechne die projizierte Größe des Objektes;

$R_s \leftarrow \frac{(1-O_S)}{V_S} I_S$; // Berechne die Ratings

$R_t \leftarrow O_T I_T$;

wenn $A_{ray} > 0.1$ **und** $R < (R_S + R_T)$ **dann**

$R \leftarrow R_S + R_T$;

$O_{result} \leftarrow O_i$;

Ende

$A_{ray} \leftarrow A_{ray} O_T$;

Ende

Algorithmus 9.1: Automatische Ermittlung des selektierten Objektes bei einer Mehrdeutigkeit durch sich überlagernde semi-transparente Objekte.

Weitere Interaktionstechniken zur Objektselektion: Der zuvor beschriebene Algorithmus stößt an seine Grenzen, wenn eine transparente Struktur vor einer größeren opaken liegt (Abb. 9.11(b)). In solchen Fällen würde immer die hintere, opake Struktur selektiert werden und der Nutzer könnte nie die transparente Struktur auswählen. Daher wurden im METK zwei Interaktionstechniken integriert, die den Prozess der Selektion unterstützen sollen. Auch wenn sich zahlreiche Arbeiten mit der Selektion von Objekten in 3d-Szenen beschäftigt haben, z.B. VANACKEN u. a. [2009], ARGELAGUET u. a. [2008] und ELMQVIST & FEKETE [2008], so ist keine Arbeit bekannt, die sich mit der Selektion semi-transparenter Objekte beschäftigt.

Die erste Methode erlaubt es dem Nutzer, nach einem Klick in die Szene mit dem Mousrad zwischen den Strukturen entlang des Selektionsstrahls zu scrollen. Beginnend bei der durch den automatischen Algorithmus gewählten Struktur kann der Nutzer die Strukturen davor oder dahinter ansrollen. Die jeweils aktuell selektierte Struktur wird hervorgehoben. Durch einen weiteren Klick bestätigt der

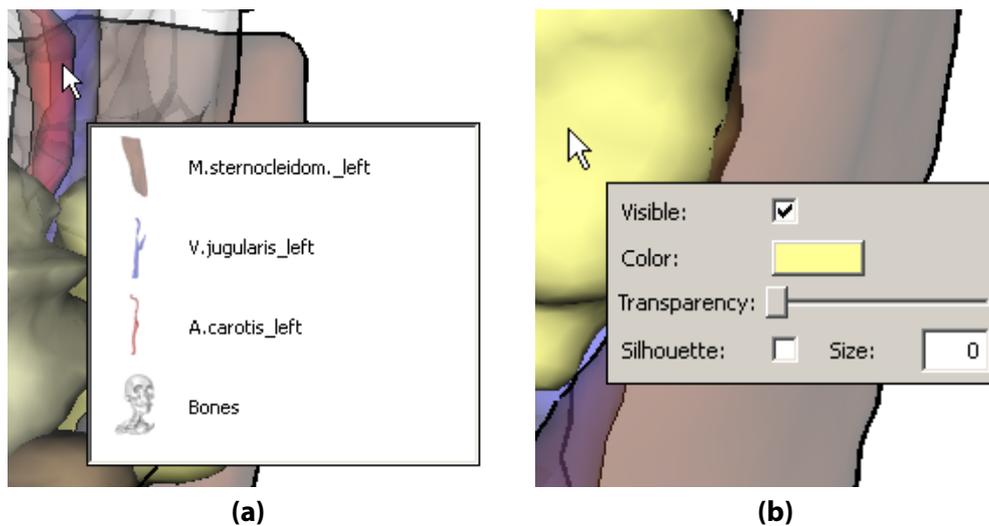


Abbildung 9.12: Kontextmenüs zur schnelleren Exploration. **(a):** Alle Strukturen entlang eines Selektionsstrahls werden dem Nutzer zur Schnellauswahl in einer Liste präsentiert. **(b):** Zur schnellen Manipulation von Visualisierungsparametern kann ein Kontextmenü direkt an der Cursorposition eingeblendet werden (aus [Mühler u. a. \[2010\]](#)).

Nutzer die Auswahl. Die zweite Methode offeriert dem Nutzer in Form eines Kontextmenüs an der Stelle des Klicks eine Liste der selektierbaren Strukturen (Abb. 9.12(a)). Die Liste der Strukturen kann durch Icons der jeweiligen Struktur erweitert werden. Diese Technik zielt vor allem auf erfahrene Nutzer ab, die die Strukturen durch ihren Namen identifizieren können. Für Nutzer eines Lernsystems ist sie weniger geeignet.

Grafische Benutzeroberfläche: Vordefinierte Steuerelemente für wiederkehrende Aufgaben ermöglichen es einem Applikationsentwickler, schneller nutzbare Applikationen zu entwickeln. Das METK bietet daher einige Steuerelemente, mit denen beispielsweise einzelne Visualisierungsparameter von Strukturen, wie Farbe, Transparenz oder Silhouettendicke, angepasst werden können. Ebenso werden Listen zur Auswahl von Strukturen oder diverse Boxen für Patienten- und Strukturinformationen vorgehalten. Die Steuerelemente können in Form einzelner Paneele oder Fenster in eine Applikation integriert werden. Darüber hinaus kann der Nutzer Visualisierungsparameter einzelner Strukturen auch direkt im Viewer manipulieren (Abb. 9.12(b)), wenn dies vom Entwickler einer Applikation gewünscht ist.

Unterschiedliches Feedback von Chirurgen offenbarte klar die Bevorzugung einer gewissen Führung des Nutzers gegenüber einer zu großen Flexibilität. Chirurgen bevorzugen klare und einfach zu verstehende Benutzungsschnittstellen ([CORDES u. a. \[2007a\]](#)) anstelle von Oberflächen mit vielen Schiebereglern und Parameterfeldern. Chirurgen möchten die für ihre Fragestellung passende Visualisierung am besten automatisch geliefert bekommen oder sie maximal aus einer

kleinen Liste von Vorschlägen auswählen können. Das METK unterstützt dies mit vorgefertigten Templates zur Oberflächengestaltung ganzer Applikationen. Diese Oberflächen wurden in vielen Befragungen und Gesprächen mit Chirurgen verfeinert und angepasst (CORDES u. a. [2007a]).

9.4 Applikationsentwicklung mit dem METK

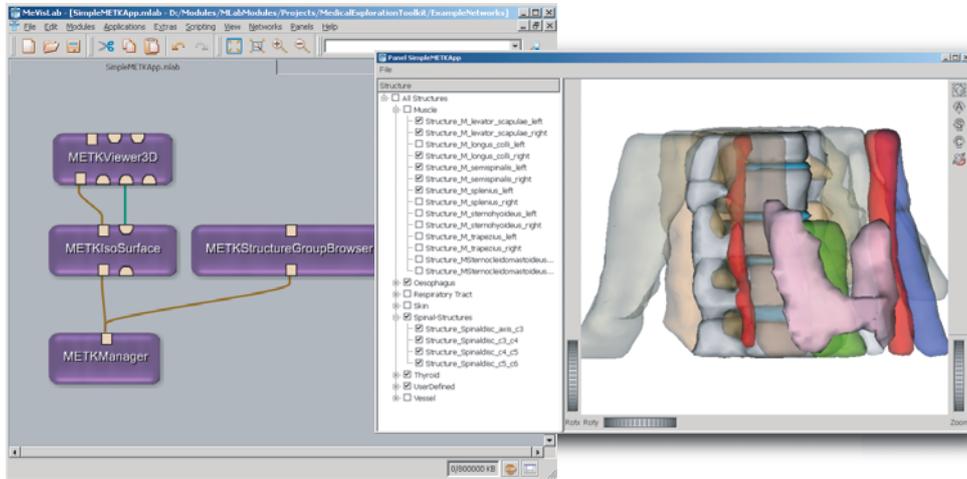
Da das METK eine Erweiterung auf der Basis von MEVISLAB darstellt, können alle Applikationen mit dem METK auf der Basis von Netzwerken in MEVISLAB erstellt werden. Dabei können die METK-Module frei mit anderen MEVISLAB-Modulen kombiniert und so ein individuelles Anwendungsprofil erstellt werden. Durch die Fokussierung des Applikationsentwicklers auf die Applikationslogik und nicht auf die darunter liegenden Visualisierungs- und Interaktionstechniken können schnell nutzbare Applikationen erstellt werden (Abb. 9.13). Die Applikationslogik wird in Form von PYTHON-Skripten beschrieben. Das METK unterstützt diesen Prozess zusätzlich durch die Bereitstellung vieler sofort nutzbarer PYTHON-Skripte. Die Definition der Benutzeroberflächen erfolgt ebenfalls skriptbasiert mithilfe der MEVISLAB eigenen Skriptsprache Module Definition Language (MDL).

Das METK kann von Entwicklern durch eigene Module erweitert und einzelne METK-Module können den individuellen Bedürfnissen angepasst werden. Abhängig von der Komplexität der Erweiterung gibt es zwei Möglichkeiten:

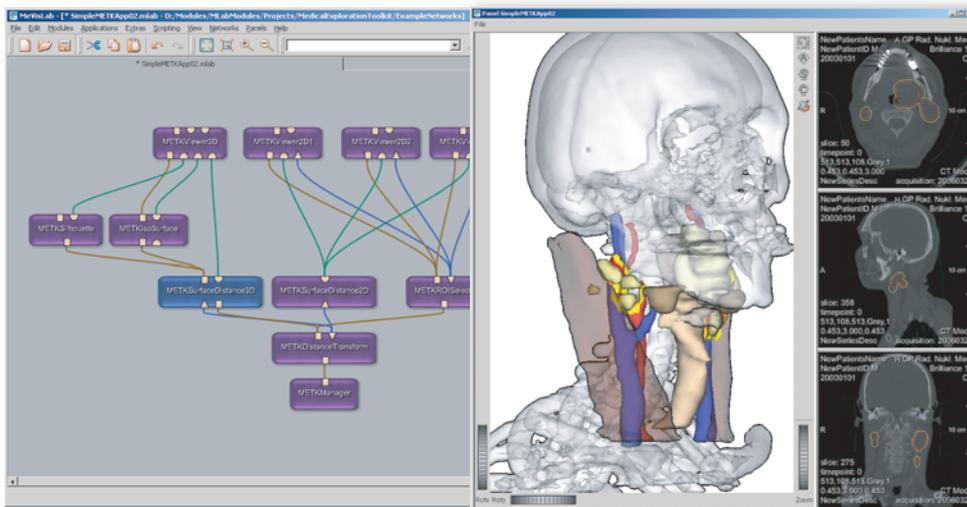
1. Einfache Erweiterungen, wie zum Management von Patientendaten oder neue Steuerelemente und Paneele, können in PYTHON entwickelt werden.
2. Fortgeschrittene, und vor allem performanzkritische, Erweiterungen können in Form von C++-Bibliotheken implementiert werden. Dies ist für spezielle Visualisierungstechniken, die noch nicht im METK enthalten sind, z.B. Visualisierung von DTI-Daten, nötig.

In Abbildung 9.14 werden die Schritte zur Entwicklung eines eigenen METK-Modules, welches das momentan selektierte Objekt (Currently Selected Object (CSO)) mit Namen und anatomischer Zugehörigkeit visualisiert, illustrativ dargestellt.

Bisher wurden verschiedene umfangreiche Applikationen mit dem METK entwickelt. Für das Training von Leberchirurgen wurde der LIVERSURGERYTRAINER (Abb. 9.15(a)) entwickelt, der in Kapitel 10 genauer beschrieben wird. Das Feedback der Chirurgen während der Entwicklung des LIVERSURGERYTRAINERS (CORDES u. a. [2007a]) beeinflusste viele Verbesserungen und Anpassungen des METK. Der NECKSURGERYPLANNER (Abb. 9.15(b)) unterstützt den Entscheidungsfindungsprozess von HNO-Chirurgen für Halslymphknotenausräumungen (TIETJEN u. a. [2006a]). Der SPINESURGERYTRAINER (Abb. 9.15(c)) dient dem Training



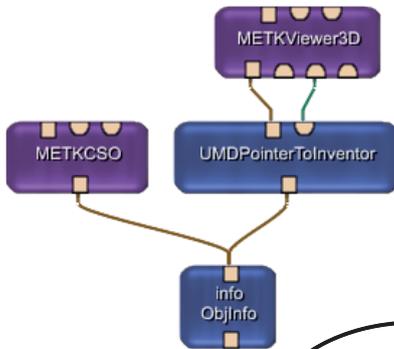
(a)



(b)

Abbildung 9.13: Beispielnetzwerke des METK. **(a):** Ein Netzwerk für eine kleine Applikation zur Darstellung segmentierter Daten in einem 3D-Viewer mit einem Steuerelement, um die Sichtbarkeit einzelner Strukturen zu verändern. Es ist ein Datensatz der orthopädischen Therapieplanung zu sehen. **(b):** Ein Netzwerk für eine Applikation zur synchronen Exploration von 2D- und 3D-Daten. In der 3D-Ansicht werden zusätzlich Silhouetten genutzt. Es ist ein Datensatz der HNO-Operationsplanung zu sehen (aus [Mühler u. a. \[2010\]](#)).

1. Erstellung eines neuen Modules mit dem MeVisLab Module Wizard



2. Erstellung eines Netzwerkes mit den Modulen zur Kommunikation, zur Ermittlung der selektierten Struktur, zum Laden des polygonalen Netzes aus dem Cache und ein Viewer zur Visualisierung.

3. Schreiben eines Skriptes zur Definition der Benutzeroberfläche.

```
Window {
  title = "Current selected object"
  borderless = True
  Vertical {
    margin=5
    spacing=5
    expandY=yes expandX=yes
    Viewer SoCustomExaminerViewer.self (
```

4. Ergebnis: Ein Paneel mit der Visualisierung der aktuell selektierten Struktur.



Abbildung 9.14: Erstellung eines eigenen Moduls mit dem METK. Das Modul soll das aktuell selektierte Objekt anzeigen. Dazu müssen ein Netzwerk erstellt und ein Skript für die Benutzeroberfläche geschrieben werden.

von Orthopäden zur Planung von Interventionen an der Wirbelsäule ([CORDES u. a. \[2008\]](#)).

9.5 Evaluierung des METK anhand der Implementierung einer Referenzapplikation

Um die Effizienz des METK im Vergleich zum herkömmlichen Entwicklungsprozess einer Applikation in MEVISLAB ermessen zu können, wurde eine kleine Referenzapplikation sowohl in der Standardversion von MEVISLAB als auch in der um das METK erweiterten MEVISLAB-Version implementiert. Die Applikation sollte einen gegebenen CT-Datensatz sowie mehrere Segmentierungsmasken einlesen und die 3D-Oberflächen der Strukturen sowie die 2D-Schichtbilder visualisieren. In der 3D-Darstellung sollten die Strukturen in einer für sie typischen Farbe dargestellt werden. Die Sichtbarkeit der Strukturen sollte durch den Nutzer individuell verändert werden können. Im Folgenden werden nach der Beschreibung des jeweiligen Entwicklungsprozesses die Ansätze hinsichtlich ihrer Entwicklungszeit, den resultierenden Applikationen sowie der Usability der Ansätze verglichen.

Bei der Implementierung auf Basis des einfachen MEVISLABs müssen alle Segmentierungsmasken manuell geladen werden. Anschließend müssen für jede Struktur die Module zur Erzeugung der polygonalen Oberflächen hinzugefügt und parametrisiert werden. Um die Strukturen in ihrem jeweils typischen Stil zu visualisieren, müssen die Module nach den entsprechenden Strukturtypen (z.B. Gefäße, Muskeln) gruppiert werden und Materialmodule (Farbe und Transparenz) für jede Gruppe hinzugefügt und parametrisiert werden. Für die 2D-Darstellung muss der CT-Datensatz geladen und mit einem 2D-Viewer dargestellt werden. Die Viewer werden mithilfe eines Skriptes zu einem Applikationsfenster zusammengefügt. Um die Sichtbarkeit jeder einzelnen Struktur in der Applikation ändern zu können, muss der Sichtbarkeitsparameter jeder Struktur aus dem zugrunde liegenden Netz in das Applikationsskript eingetragen werden. Dieser gesamte Prozess muss für jeden Datensatz neu durchgeführt werden, da nur das strukturspezifische Netzwerk gespeichert werden kann und kein Fallmanagement vorhanden ist. Das Netzwerk sowie die fertige Applikation auf der Basis der einfachen Version von MEVISLAB sind in [Abbildung 9.16](#) zu sehen.

Bei der Nutzung des METK muss zunächst ein Managermodul hinzugefügt werden, welches das Laden und Speichern eines Falles, die Applikationskommunikation sowie die Erstellung der polygonalen Netze der Strukturen übernimmt. Ein zweites Modul übernimmt das Importieren des CT-Datensatzes und der Segmentierungsmasken. Zur 3D-Visualisierung werden ein Modul zur Oberflächenvisualisierung und der 3D-Viewer zum Applikationsnetzwerk hinzugefügt. Anschließend werden die Viewer sowie ein zusätzliches Steuerelement, welches eine Liste aller Strukturen anbietet, über ein Skript zum Applikationsfenster zusammengefügt und angeordnet. Nachdem die Applikation ausgeführt wurde, kann

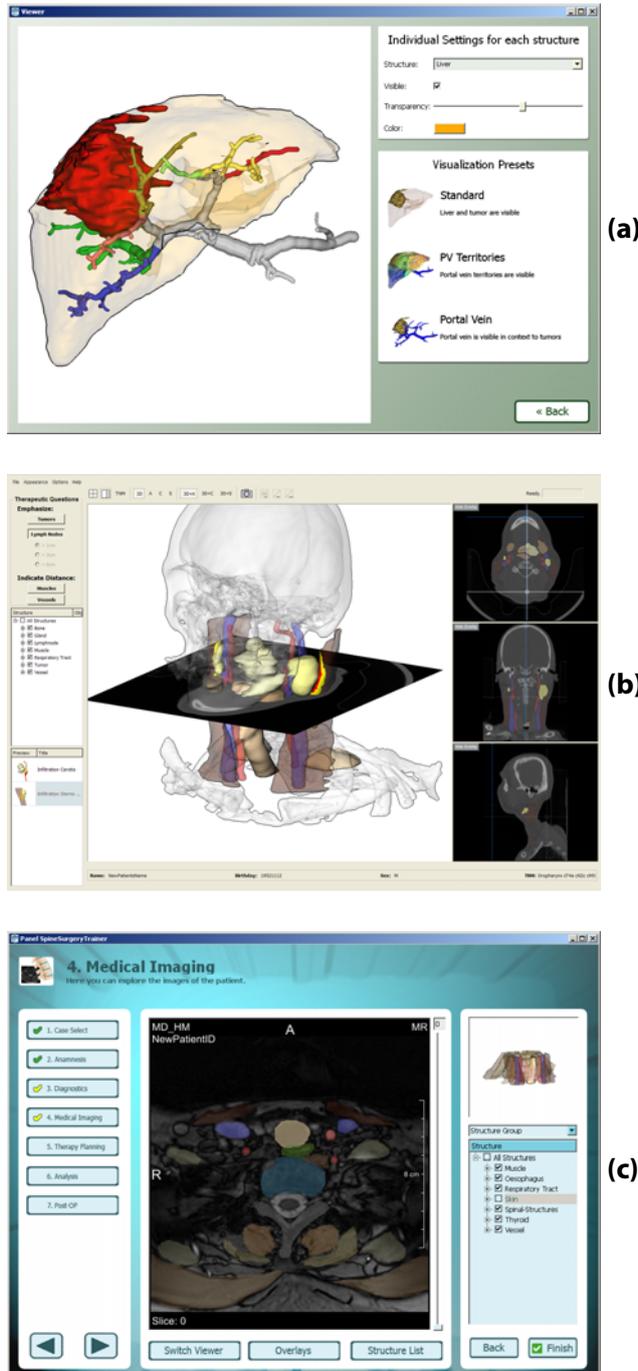


Abbildung 9.15: Komplexe Applikationen, die mit dem METK erstellt wurden. **(a):** Der LiverSurgeryTrainer ist eine Applikation zum Training von Leberchirurgen und wird in Kapitel 10 näher vorgestellt. **(b):** Der NeckSurgeryPlanner ist eine Applikation zur Unterstützung der Planung von Halslymphknotenausräumungen. Für einen umfassenden Blick auf die Daten werden sowohl 2D- wie 3D-Sichten präsentiert. Im linken Bereich sind zusätzlich Steuerelemente für die Keystates und zur Strukturauswahl vorhanden. **(c):** Der SpineSurgeryTrainer ist ein Trainingsprogramm zur Ausbildung von Orthopäden in der Planung von Interventionen an der Wirbelsäule.

durch den Nutzer ein neuer Fall angelegt werden, wobei die Segmentierungsmasken automatisch importiert werden. Zu jeder Struktur kann ein Name sowie ihre anatomische Zugehörigkeit angegeben werden. Dies ist allerdings nur notwendig, wenn der Fall nicht mit einer METK-kompatiblen Anwendung erstellt wurde.⁴⁷ Zur späteren Wiederverwendung kann der Fall gespeichert werden, so dass die Operation des Importierens und Zuordnens der Strukturen nur einmal vorgenommen werden muss. Das Netzwerk sowie die fertige Applikation auf der Basis des METK sind in Abbildung 9.17 zu sehen.

Da mit der einfachen Version von MEVISLAB für jeden Fall ein komplett neues Netzwerk manuell erstellt werden muss, dauerte es 51 Minuten, um alle 148 Module für den Beispielfall zu erzeugen und zu parametrisieren (Abb. 9.16). Die Erstellung des Skriptes für die Applikationslogik dauerte 8 Minuten. Die in der Summe für jeden Fall benötigte Zeit hängt stark von der Anzahl der segmentierten Strukturen eines Falles ab.

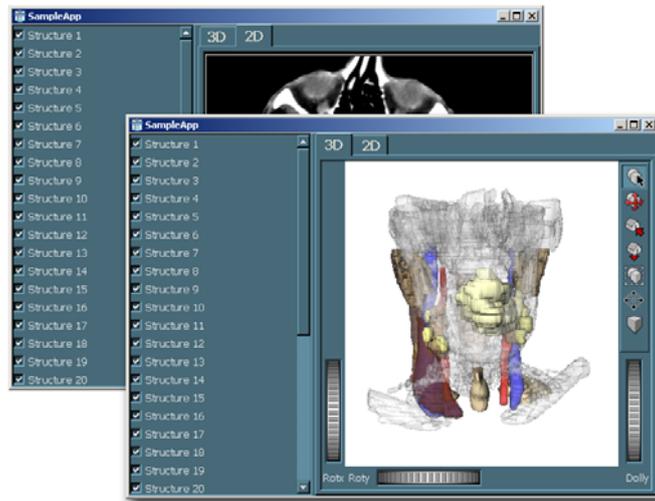
Mit dem METK dauerte es weniger als eine Minute, das Netzwerk aus nur 6 METK-Modulen zu erzeugen (Abb. 9.17). Weitere 5 Minuten wurden für das Erstellen des Applikationsskriptes und 8 Minuten für das Importieren und Zuordnen der Segmentierungsmasken benötigt.

Vom Anfang bis zur fertigen Visualisierung der Daten dauerte es 14 Minuten mit dem METK und 59 Minuten mit MEVISLAB ohne METK. Für das Laden eines neuen Falles (so er denn nicht METK kompatibel ist) wird mit dem METK nur noch die Zeit für das Importieren der Segmentierungsmasken benötigt, wo bei der einfachen MEVISLAB-Version das Netzwerk immer wieder neu erstellt werden muss.

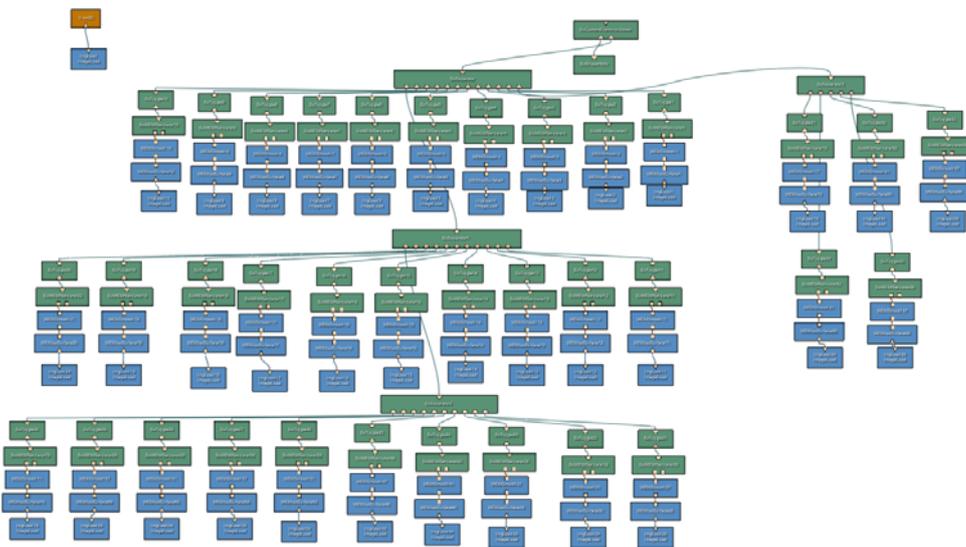
9.6 Zusammenfassung

Es gibt viele Entwicklungswerkzeuge im Bereich der wissenschaftlichen Visualisierung. Einige unterstützen fortgeschrittene Techniken der Entwicklung, wie graphische Netzwerkprogrammierung (z.B. SCIRUN (WEINSTEIN u. a. [2005]), MEVISLAB und DEVIDE (BOTH & POST [2008])). Auch wenn SCIRUN und MEVISLAB grundlegende Funktionen zur Erstellung eigener Applikationen anbieten, so ist es doch immer noch sehr aufwendig und komplex, sie zu erstellen und einen Zustand zu erreichen, der ihre Benutzung auch durch weniger erfahrene Benutzer garantiert. In einer Vergleichsstudie von vier Visualisierungs-Frameworks (darunter MEVISLAB und SCIRUN) wurde MEVISLAB als das beste Framework zur Applikationsentwicklung ermittelt (BITTER u. a. [2007]). Daher wurde, auf MEVISLAB aufbauend, das METK entwickelt, welches speziell den Entwicklungsprozess chirurgischer Applikationen beschleunigt und vereinfacht.

⁴⁷Solche Applikationen sind beispielsweise HEPAVISION (BOURQUAIN u. a. [2002]) und NECKVISION (CORDES u. a. [2006b]).

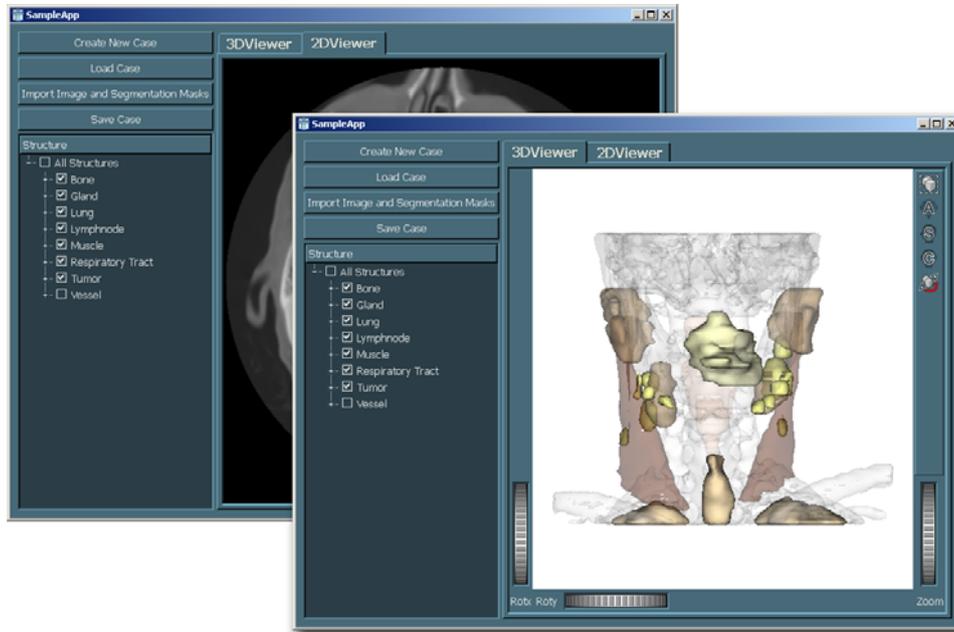


(a)

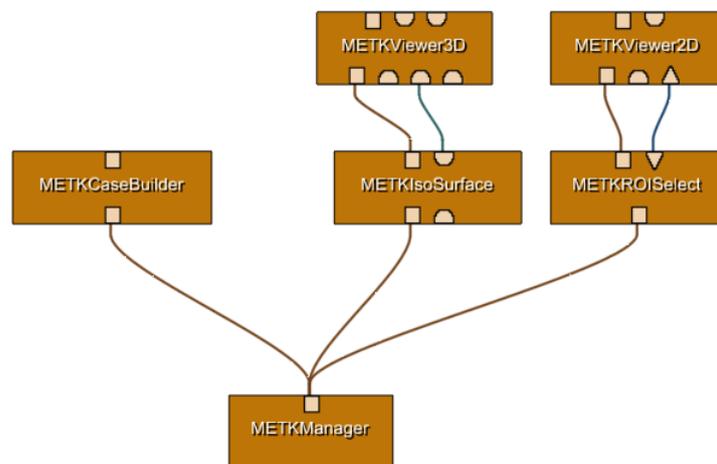


(b)

Abbildung 9.16: Beispielapplikation (a) und Netzwerk (b), wie sie mit der einfachen Version von MeVisLab erstellt wurden. Das umfangreiche Netzwerk muss für jeden Fall manuell erstellt werden – ebenso die Liste der Strukturen im linken Teil des Fensters zur Änderung der Sichtbarkeiten.



(a)



(b)

Abbildung 9.17: Beispielapplikation (a) und Netzwerk (b), erstellt auf der Basis des METK. Das Netzwerk ist übersichtlich und muss nur einmal erzeugt werden. Neue Fällen können einfach geladen oder importiert werden.

Auch wenn gewisse Kenntnisse in PYTHON und der Skriptsprache MDL von MEVISLAB nötig sind, um Applikationen zu entwickeln, so ist es doch mit dem METK wesentlich einfacher geworden, schnell einen funktionsfähigen Applikationsprototypen zu erstellen, da viele Visualisierungs- und Explorationstechniken schon fertig im METK implementiert sind. Die Applikationsnetzwerke sind zudem wesentlich übersichtlicher und einfacher zu warten und zu erweitern.

Das METK steht unter <http://www.metk.net> als quelloffene Erweiterung von MEVISLAB zum Download zur Verfügung.

10

Der LiverSurgeryTrainer – Eine Lernsoftware für Viszeralchirurgen

Im Rahmen der vorliegenden Arbeit wurde im Zuge eines Projektes ein Trainingssystem für Viszeralchirurgen, der LIVERSURGERYTRAINER, entwickelt. Mit dieser Applikation soll die Planung von leberchirurgischen Eingriffen, vor allem onkologischen Tumorresektionen und Leberlebendspenden, trainiert werden. Der LIVERSURGERYTRAINER stellt zusätzlich eine Plattform für den praktischen Einsatz und die Evaluierung der im Rahmen dieser Arbeit neu entwickelten Techniken dar. Mit den über den LIVERSURGERYTRAINER gewonnenen Erkenntnissen konnten u.a. die Techniken der Sichtpunktbestimmung und der Animationsgestaltung verfeinert werden.

Im LIVERSURGERYTRAINER wurden neueste didaktische Konzepte angewendet und adaptiert (Abschnitt 10.2.2) und zahlreiche der in dieser Arbeit vorgestellten Techniken umfänglich integriert. Im Rahmen einer ersten ausführlichen Evaluierung der Software (Abschnitt 10.3) traten besonders Aspekte der Gestaltung von Benutzerschnittstellen speziell für Chirurgen in den Vordergrund, die in diesem Kapitel eine gesonderte Betrachtung finden (Abschnitt 10.4).

Eigene Publikationen

Dieses Kapitel basiert u.a. auf folgenden eigenen Publikationen:

- [MÜHLER u. a. \[2008b\]](#) Mühler, K.; Cordes, J.; Tietjen, C. & Preim, B. *Richtlinien für die Gestaltung chirurgischer Planungs- und Trainings-Software*. In: **Proc. of Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie e. V. (CURAC)**, 2008
- [MÜHLER u. a. \[2007a\]](#) Mühler, K.; Cordes, J.; Preim, B.; Hillert, C. & Oldhafer, K. *Liver-SurgeryTrainer 1.0 (DVD)*, 2007
- [LOGGE u. a. \[2007\]](#) Logge, C.; Cordes, J.; Mühler, K.; Preim, B. & Hillert, C. *Training in Living Donor Liver Transplantation - The LiverSurgeryTrainer*. In: **Proc. of ILTS 13th Annual International Congress**, 2007
- [CORDES u. a. \[2007a\]](#) Cordes, J.; Mühler, K.; Oldhafer, K.; Stavrou, G.; Hillert, C. & Preim, B. *Evaluation of a training system of the computer-based planning of liver surgery*. In: **Proc. of Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie e. V. (CURAC)**, 2007
- [CORDES u. a. \[2006a\]](#) Cordes, J.; Mühler, K. & Preim, B. *Die Konzeption des LiverSurgeryTrainers*. In: **Proc. of Workshop Softwareassistenten**, 2006
- [CORDES u. a. \[2007b\]](#) Cordes, J.; Mühler, K.; Oldhafer, K. J.; Stavrou, G.; Hillert, C. & Preim, B. *Szenariobasierte Entwicklung eines chirurgischen Trainingssystems*. In: **eLearning in der Medizin und Zahnmedizin (Proc. of Workshop der GMDS AG)**, 2007

10.1 Projekthintergrund, Analyse und Zielsetzung

Die chirurgische Aus- und Weiterbildung ist immer noch stark von den lokal an einer Klinik zur Verfügung stehenden Experten abhängig. Auch ist für die Qualität der Ausbildung mitentscheidend, welche Ausprägung das örtlich vorhandene Fallspektrum hat. In der Ausbildung ist neben dem Erlernen von Operationstechniken die Auswahl der bestmöglichen Therapie entscheidend. Im Bereich der Leberchirurgie geht es dabei beispielsweise um die Auswahl einer geeigneten Resektion, die Einschätzung der Operabilität des Patienten oder die Vorbereitung möglicher Gefäßrekonstruktionen. Um von den lokalen Faktoren unabhängiger zu werden und die Qualität der Ausbildung insgesamt anzuheben, bietet sich das Training der Therapiewahl, also der Planung, an. In diesem Bereich ist man nicht auf reale Operationen vor Ort oder umfangreiche Technik angewiesen, wie sie z.B. bei Simulatoren zum Training von minimalinvasiven Operationen zum Einsatz kommen ([DELINGETTE & AYACHE \[2005\]](#), [LIU u. a. \[2003\]](#), [PFLESSER u. a. \[2002\]](#), [SIMBIONIX USA CORP.](#)). Zum Training der Planung von leberchirurgischen Eingriffen wurde daher der LIVERSURGERYTRAINER als Trainingssoftware entwickelt. Wesentliche Ziele des LIVERSURGERYTRAINERS sind:

- die Ausbildung angehender Viszeralchirurgen in der Planung von Eingriffen auf der Basis von rekonstruierten 3D-Modellen und
- die Sensibilisierung für das noch neue Feld der 3D-Planung und dessen Verbreitung.

Der LIVERSURGERYTRAINER liegt aktuell in einer ersten Version vor. Diese wurde einer ausführlichen Evaluierung unterzogen (siehe Abschnitt 10.3), auf deren Basis momentan eine zweite Version entwickelt wird. Ziel ist es, den LIVERSURGERYTRAINER einem breiten Nutzerkreis an angehenden Viszeralchirurgen für die Ausbildung zur Verfügung zu stellen. Neben einem Vertrieb der Software an individuelle Nutzer werden derzeit auch ein Einsatz im Rahmen von größeren Kursen zum chirurgischen Training sowie ein Einsatz in der Umgebung der im Rahmen des FUSION-Projektes neu entwickelten Techniken diskutiert und vorbereitet.

Der LIVERSURGERYTRAINER entstand im Rahmen des BMBF-geförderten Projektes FUSION (FUTURE Environment for Gentle Liver Surgery Using Image-Guided Planning and Intra-Operative Navigation). Eingebettet war und ist das FUSION-Projekt in das Forschungsprogramm SOMIT (Schonendes Operieren mit innovativer Technik), welches neben dem FUSION-Projekt weitere Projekte zur orthopädischen Chirurgie sowie zur Augenheilkunde umfasst. Ziel des FUSION-Projektes ist die Entwicklung und Etablierung neuer Techniken zur Unterstützung der Leberchirurgie. Dabei liegt der Fokus auf dem intra-operativen Bereich, bei dem durch neue Navigationstechniken und Geräte Operationen genauer, effizienter und schonender für den Patienten werden sollen. Als Beispiel sei an dieser Stelle ein Smartes Infrarot Laser-Skalpell (SILAS) zur schonenden Gewebedissektion zu nennen (THEISEN-KUNDE u. a. [2007]). Weitere Aufmerksamkeit im FUSION-Projekt gilt der Planung von leberchirurgischen Eingriffen sowie der technischen Realisierung eines Transfers der Planungsdaten in den Operationsraum. Der LIVERSURGERYTRAINER greift dabei die Problematik der Planung eines Eingriffes unter dem Aspekt der Ausbildung auf.

10.2 Konzept des LiverSurgeryTrainers

Der LIVERSURGERYTRAINER wurde im Duktus der klassischen Softwareentwicklung von Analyse, Spezifikation, Programmierung und Validierung entwickelt. In den folgenden Abschnitten werden die Analyse (Abschnitt 10.2.1), das didaktische Konzept (Abschnitt 10.2.2) sowie die implementierten Funktionen des LIVERSURGERYTRAINERS (Abschnitt 10.2.3) beschrieben. Der Evaluierung ist ein eigener Abschnitt gewidmet (Abschnitt 10.3).

10.2.1 Anforderungsanalyse

Der Spezifikation des LIVERSURGERYTRAINER ging eine Analyse der Zielgruppe sowie der durchzuführenden Aufgaben voraus. Diese Analysen werden im Folgenden kurz vorgestellt.

Anwenderanalyse

Die späteren Nutzer des LIVERSURGERYTRAINER sind Fachärzte bzw. angehende Fachärzte oder Assistenzärzte der Chirurgie, zumeist mit dem Ziel der Subspezialisierung Abdominalchirurgie. Sie verfügen über ein umfangreiches Anatomiewissen, so dass diese vom Trainingssystem vorausgesetzt werden können. Die Erfahrungen mit Therapieentscheidungen sind dagegen sehr heterogen, ebenso die Erfahrungen im Umgang mit Computern. Ein Trainingssystem muss daher sowohl dem unterfahrenen Anwender in allen Bereichen eine sehr gute Unterstützung geben, darf dem erfahreneren Anwender allerdings auch nicht zu viel „vorschreiben“ und muss ebenso für ihn ein flüssiges Arbeiten und Lernen ermöglichen. Aufgrund der geringen Verbreitung von computergestützten Planungssystemen kann von nicht vorhandenen Kenntnissen in diesem Bereich für die Zielgruppe ausgegangen werden.

Aufgabenanalyse

Der Lernende soll nach Absolvierung des Trainings mit dem LIVERSURGERYTRAINER in der Lage sein, den spezifischen Arbeitsablauf auf dem Weg zu einer Therapieentscheidung nachvollziehen und selbst abbilden zu können. Das Training soll der Förderung der Regelautomatisierung und mentalen Schemakonstruktion für den Arbeitsablauf dienen. Ebenso soll es die Erlernung der verwendeten Interaktionstechniken ermöglichen. Das Training der Therapieentscheidung umfasst unter anderem die folgenden Aspekte:

- Die Einschätzung und Bewertung von Voruntersuchungsergebnissen im Hinblick auf eine mögliche Inoperabilität des Patienten.
- Das Erkennen und Bewerten von anatomischen Besonderheiten – wie Gefäßanomalien – sowohl auf der Basis von 3D-Patientenmodellen, als auch auf Basis der schwieriger einzuschätzenden 2D-Schichtbilddaten.
- Das Erstellen von Resektionsebenen unter Einhaltung von Randbedingungen, wie Restvolumina und Sicherheitsabständen.
- Das Erstellen von komplexen Strategien auf der Basis mehrere Resektionsvorschläge in Kombination mit interventionellen Methoden, wie der Tumorabletation.

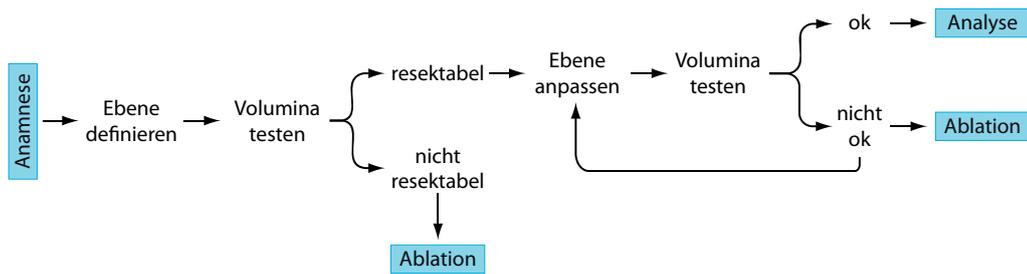


Abbildung 10.1: Verschiedene Entscheidungen, die während der Planung einer Resektion getroffen werden müssen. Das Ablaufdiagramm basiert auf der Spezifikation des LiverSurgeryTrainers. Aufgrund entwicklungstechnischer Erfordernisse konnten nicht alle Planungsschritte in der endgültigen Version des LiverSurgeryTrainers umgesetzt werden.

- Das Kennenlernen möglicher Komplikationen und Modifikationen eines Planungsvorschlags im Verlauf einer Operation.

Teil der Analyse war eine genaue szenariobasierte Ausarbeitung vieler möglicher Anwendungsfälle. Zur genaueren Beschreibung der dazu genutzten und entwickelten Konzepte sei auf eine Dissertation in Vorbereitung (MÖNCH [In Vorbereitung]) sowie CORDES u. a. [2007b] verwiesen. Im Ergebnis dieser Analysen standen verschiedene Arbeitsabläufe, die umfangreich beschrieben werden konnten. Ein Beispiel für verschiedene Schritte und Entscheidungen auf dem Weg zu einer Ablation als Therapieoption ist in Abb. 10.1 dargestellt.

10.2.2 Didaktisches Konzept

Die didaktischen Hintergründe des LIVERSURGERYTRAINERS werden aktuell in einer weiteren Dissertation genauer beleuchtet und vorgestellt (MÖNCH [In Vorbereitung]). Daher wird an dieser Stelle nur skizzenhaft auf die wesentlichen Aspekte des zugrunde liegenden Modells eingegangen.

Dem LIVERSURGERYTRAINER liegt als didaktisches Konzept ein angepasstes Vier-Komponenten-Instruktionsdesign-Modell (4C/ID) nach VAN MERRIËNBOER u. a. [2002] zugrunde. Bei diesem Modell steht die Vermittlung von Handlungswissen im Vordergrund. Das Modell wurde zur Nutzung für den LIVERSURGERYTRAINER in vielen Punkten angepasst. So sollen aufgabenspezifische Teilfertigkeiten separat trainiert werden. Dies wird zum Beispiel durch das gesonderte Training der Resektionstechnik auf abstrahierten Modellen ohne Patientenbezug erreicht. Auch wurden die Informationen und Hilfen, die dem Nutzer präsentiert werden, in zwei Bereiche unterteilt: *unterstützende Informationen*, die vor allem fachlichen Charakter haben und das Vorwissen des Nutzers mit der Teilaufgabe verbinden, und *einsatzsynchrone Informationen*, die im Wesentlichen Hilfen bei der Bedienung der Software und der Durchführung der Aufgaben enthalten. Je nach Vorwissen des Nutzers und schon absolvierter Aufgaben sollen die präsentierten Informationen dynamisch angepasst werden. So sollen zu Beginn mehr

einsatzsynchrone Informationen präsentiert werden, die bei der Bedienung der Software helfen. Diese Informationen nehmen mit der Erfahrung des Nutzers und der Nutzungsdauer der Software stetig ab und werden später lediglich auf Anfrage präsentiert.

10.2.3 Aufbau und Funktionen

Der Aufbau des LIVERSURGERYTRAINERS orientiert sich an den klinischen Abläufen und Entscheidungsfindungsprozessen zur Planung von Therapien. Die Datengrundlage des LIVERSURGERYTRAINERS sind dabei reale Patientenfälle. Die Fälle umfassen onkologische Tumorresektionen und -ablationen sowie Leberlebendspenden, sowohl für Kinder wie auch erwachsene Patienten. Die Falldaten umfassen dabei:

- Daten der Bildgebung (zumeist mehrphasige CT-Aufnahmen),
- Aufbereitete Daten, bei denen eine Vielzahl wichtiger Strukturen segmentiert wurden, z.B. Gefäßbäume und Metastasen,
- Daten einer genauen Versorgungsanalyse, die eine automatische Einteilung der Leber in ihre Segmente erlaubt,
- Grundlegende Daten des Patienten, wie Alter, Gewicht und Vorerkrankungen,
- Ergebnisse von Voruntersuchungen sowie für die Einschätzung der Leberqualität Laborwerte (z.B. der Child-Pugh-Score zur Stadieneinteilung einer Leberzirrhose),
- Ein oder mehrere Planungsvorschläge von Experten, zumeist den Chirurgen, die diesen Fall real operiert haben,
- Multimediales Material der durchgeführten Operation selbst. Dazu zählen neben Videos und Fotos aus dem OP-Saal auch Operationsprotokolle.

Der Lernende wählt zu Beginn einen Fall aus einer Liste von Fällen aus, analysiert die Krankengeschichte des Patienten, den Befund und wird mit den präoperativen 2D-Schichtbilddaten konfrontiert. Anschließend untersucht er die Gefäßanatomie und kommt durch die Platzierung von verschiedenen Resektionsebenen und Applikatoren zu einer Therapieentscheidung. Diese kann er darauf folgend mit einem oder mehreren Expertenvorschlägen vergleichen, die im System zusammen mit den Falldaten hinterlegt sind. Abschließend werden ihm unterschiedliche operative und post-operative Daten, wie Videos und Fotos der durchgeführten Operation, präsentiert. Die verschiedenen Schritte sind in Abb. 10.2 schematisch dargestellt und werden im Folgenden genauer beschrieben.

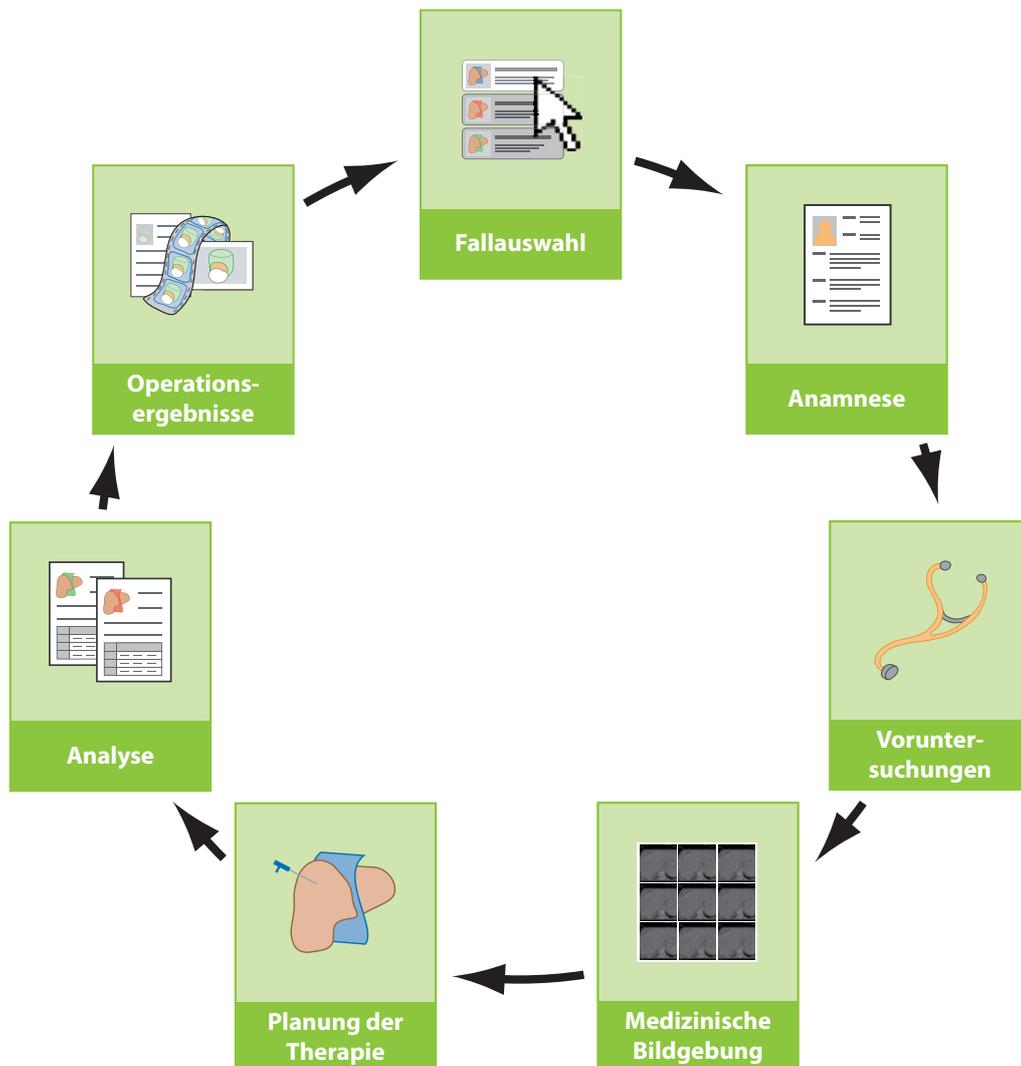


Abbildung 10.2: Die verschiedenen Schritte des LiverSurgeryTrainers, die ein Lernender bei der Bearbeitung eines Falles durchläuft.

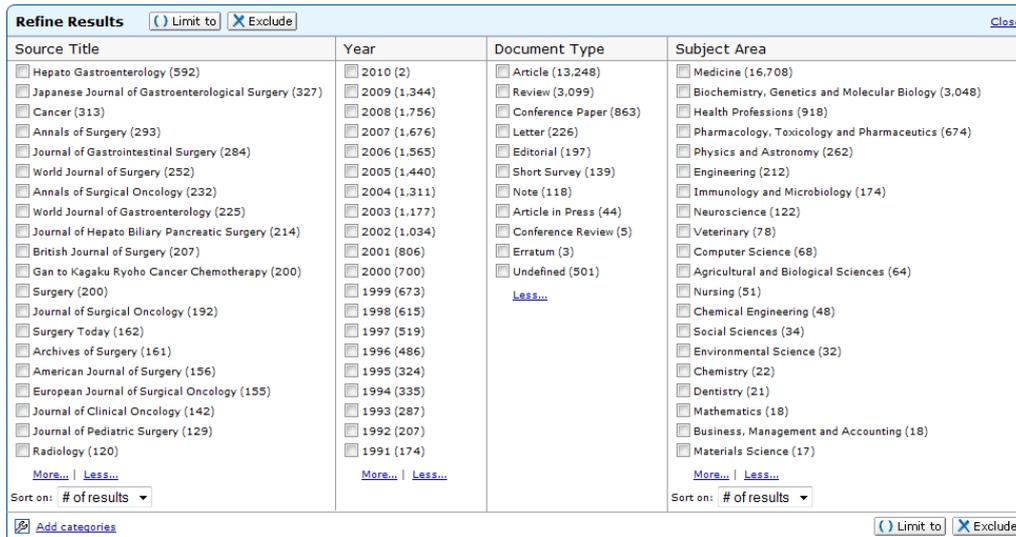


Abbildung 10.3: Verschiedene Möglichkeiten der Filterung im Literturrecherche-system Scopus, wie sie sich auch für eine Fallauswahl in einem Lern- oder Planungssystem anbieten würden.

Fallauswahl

Bei der Auswahl eines Falles, den der Lernende „trainieren“ möchte, kann er grundsätzlich zwischen Fällen der Onkologie und der Leberlebenspende wählen. Zusätzlich werden ihm Informationen zu den einzelnen Fällen präsentiert, wie die Schwierigkeit der Bearbeitung, die Anzahl der Metastasen, die Art der durchgeführten Resektion oder ein Hinweis, inwieweit er diesen Fall schon bearbeitet hat. Die Auswahl erfolgt aus einer tabellarischen Übersicht mit unterstützenden grafischen Elementen. Es wurden zwar Untersuchungen zu einer Vielzahl anderer Darbietungsformen für die Fallauswahl entwickelt und untersucht, z.B. parallele Koordinaten oder mehrfach sortierte Tabellen (BADE u. a. [2008]), jedoch stand für die übersichtliche Anzahl von Fällen im LIVERSURGERYTRAINER (~25) am Ende die einfache listenhafte Darstellung als die einzig sinnvolle. Selbst bei einer größeren Anzahl an Fällen (~100) sollte eine einfache Fallauswahl im Lernsystem verwendet werden. Eine gute Lösungsidee, die leider nicht von BADE u. a. [2008] diskutiert wurde, stellen gefilterte Listen (Facetten-Browser) dar, wie sie z.B. im Literturrechchesystem SCOPUS zum Einsatz kommen (Abb. 10.3). Dort können für verschiedene Faktoren, z.B. Jahr, Fachgebiet, Publikationsart, Werte ein- und ausgeschlossen werden. Ein solch intuitives Konzept wäre auch für eine Fallauswahl in einem Lernsystem denkbar. Hier könnte u.a. nach Schwierigkeitsgrad, Art der Operation oder Bearbeitungsstand einfach gefiltert werden.

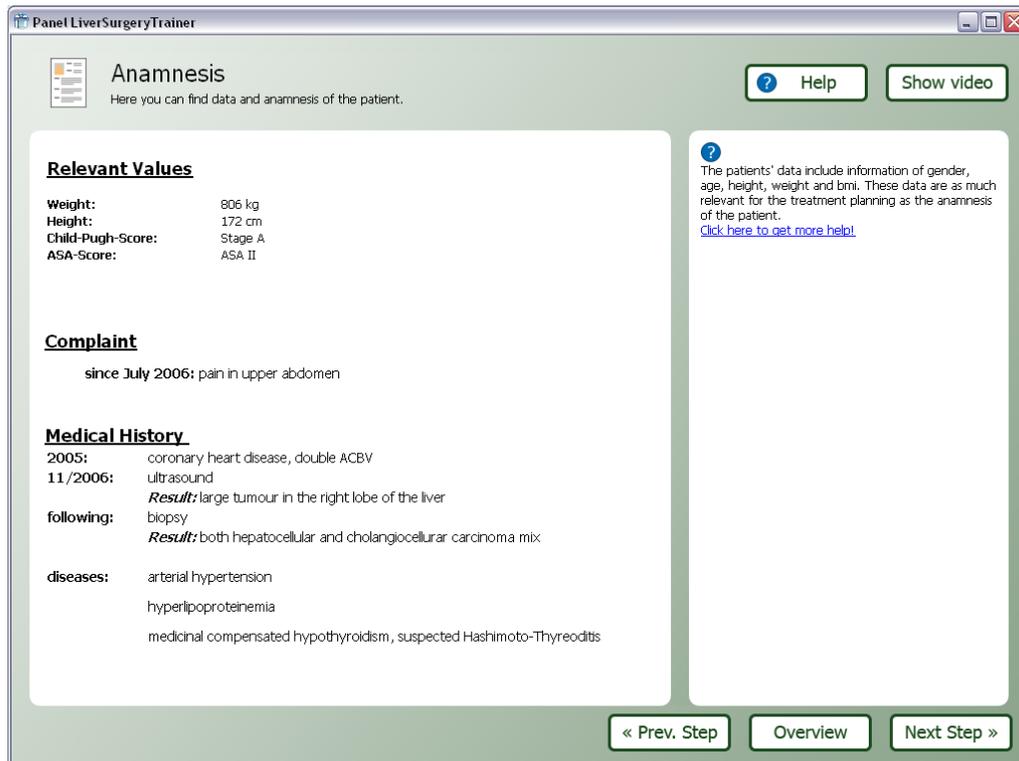


Abbildung 10.4: Präsentation der relevanten Patientendaten und Voruntersuchungen für eine erste Einschätzung des vorliegenden Falles durch den Lernenden.

Präsentation der Patientendaten

In den ersten Schritten einer jeder Falldarstellung werden dem Lernenden die Patientendaten präsentiert (Abb. 10.4). Diese reichen von grundlegenden Informationen, wie Alter und Geschlecht, bis hin zu leberspezifischen Werten, wie Child-Pugh-Score und ASA-Score. Diese Informationen werden ergänzt durch den relevanten Teil der Krankheitsgeschichte des Patienten sowie Daten zu durchgeführten Voruntersuchungen. Dazu zählen auch die zur Planung angefertigten CT-Aufnahmen. Diese werden dem Lernenden als scrollbarer Schichtbildstapel präsentiert (Abb. 10.5). Durch einen Experten wurden zuvor wichtige und aussagekräftige Schichtbilder ausgewählt, die der Lernende direkt ansteuern kann. Durch die Kombination von individueller Exploration in Verbindung mit einer Führung durch einen Experten wird der Lernende optimal an die Interpretation der Schichtbilder herangeführt. Bewusst wurde an dieser Stelle auf eine Präsentation der 3D-Daten verzichtet.

Bei den chirurgischen Ausbildungsexperten stehen das Verständnis und die Interpretation von 2D-Schichtbildern vor der Konfrontation mit den 3D-Darstellungen, die oft noch als reine Zugabe angesehen werden. Da sich auch der Planungsablauf generell erst langsam der Thematik der 3D-Darstellungen annähert, ist das in einem Lernprogramm in dieser Übergangsphase zu berücksichtigen.

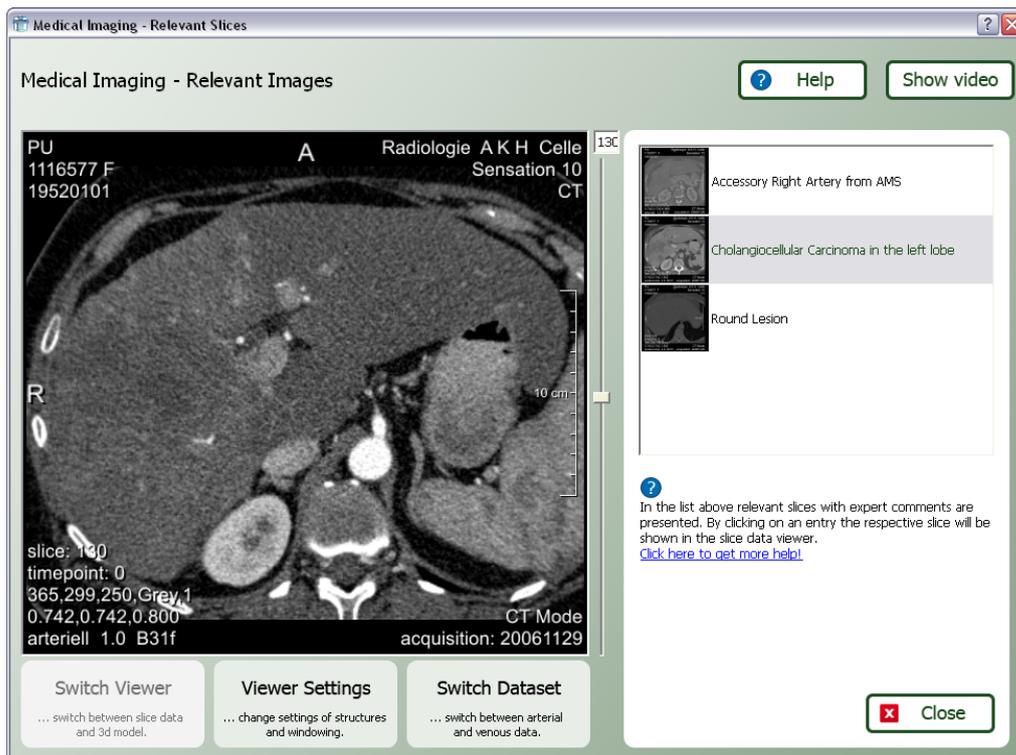


Abbildung 10.5: Darstellung der 2D-Schichtbilder im LiverSurgeryTrainer. Der Lernende kann vom Experten zuvor ausgewählte Schichten gezielt anspringen und sich mit der individuellen Patientenanatomie vertraut machen.

Erstellen von Therapiestrategien

Nachdem der Lernende sich mit den Daten vertraut gemacht hat und zu einer grundlegenden Einschätzung über den Patienten gelangt ist, folgt der zentrale Teil der Bearbeitung eines Falles: Die Planung einer oder mehrerer konkreter Strategien. Eine Strategie besteht bei Leberlebenspenden aus einer einzelnen Resektionsebene zur Teilung der Spenderleber. Bei onkologischen Tumorresektion kann sie aus mehreren Resektionsebenen bestehen (je nach Anzahl und Lage der Metastasen). Hinzu können verschiedene Applikatoren kommen, wenn eine Metastase nicht reseziert werden kann oder werden soll, sondern stattdessen durch eine Thermoablation zerstört werden soll.

Beurteilung der Gefäßanatomie. Da die Gefäßanatomie bei der Planung von Resektionsebenen und bei der Applikatorplatzierung von grundlegender Bedeutung ist, ist der Lernende zunächst angehalten, besondere Anomalien in den Gefäßbäumen der Leber des Patienten zu identifizieren. Dies erfolgt aus oben genannten Gründen auf den 2D-Schichtbilddaten, wobei die 3D-Darstellung der Gefäße hinzugezogen werden kann. Dem Lernenden stehen je nach Gefäßbaum verschiedene Optionen zur Auswahl (z.B. eine akzessorische Lebervene von Segment IV).

Erstellen einer Resektionsebene. Zumeist wird der Lernende anschließend eine Resektionsebene erstellen. Diese Ebene wird nach einem Verfahren von [KONRAD-VERSE u. a. \[2004\]](#) berechnet. Dabei zeichnet der Lernende den geplanten Verlauf in einigen Schichtbildern (Abb. 10.6(a)) oder auf der Organoberfläche ein. Aus der so generierten Punktwolke wird automatisch eine erste Ebene generiert. Diese kann dann in einem weiteren Schritt genau angepasst werden (Abb. 10.6(b)). Daneben sind vor allem größere Gefäße und der Sicherheitsabstand zur Metastase zu beachten. Die Ebene kann wahlweise auch direkt in der 3D-Darstellung erzeugt werden. Dazu wird der Verlauf der Ebene auf der Leberoberfläche eingezeichnet. Die daraus generierte Ebene kann ebenso in 3D manipuliert werden, indem sie an gewünschten Stellen eingedrückt oder herausgezogen wird (Abb. 10.7).

Ein weiteres Kriterium bei der Ebenenerstellung ist das verbleibende und das resezierte Lebervolumen. Diese Werte werden automatisch berechnet und dem Lernenden präsentiert. Auf der Basis dieser Werte kann dieser einschätzen, ob ausreichend Restvolumen im Körper des Patienten verbleibt oder ob bei einer Lebendspende das entnommene Parenchym des Spenders für den Empfänger ausreicht.

Platzierung eines Applikators. In Fällen, in denen eine Metastase nicht entnommen werden kann (z.B. weil sie zu dicht an wichtigen Gefäßen liegt) oder nicht entnommen werden soll, ist die Platzierung eines Applikators vorgesehen. Für den Chirurgen ist dann entscheidend, aus welcher Richtung der Applikator eingeführt werden muss und welchen Typs er sein soll. Ebenso ist eine Planung der Ablationsdauer wichtig, um schon vor dem Eingriff beurteilen zu können, ob es gelingen kann, sämtliches Tumorgewebe durch eine Ablation zu zerstören oder ob möglicherweise weitere zusätzliche Applikatoren platziert werden müssen. Im

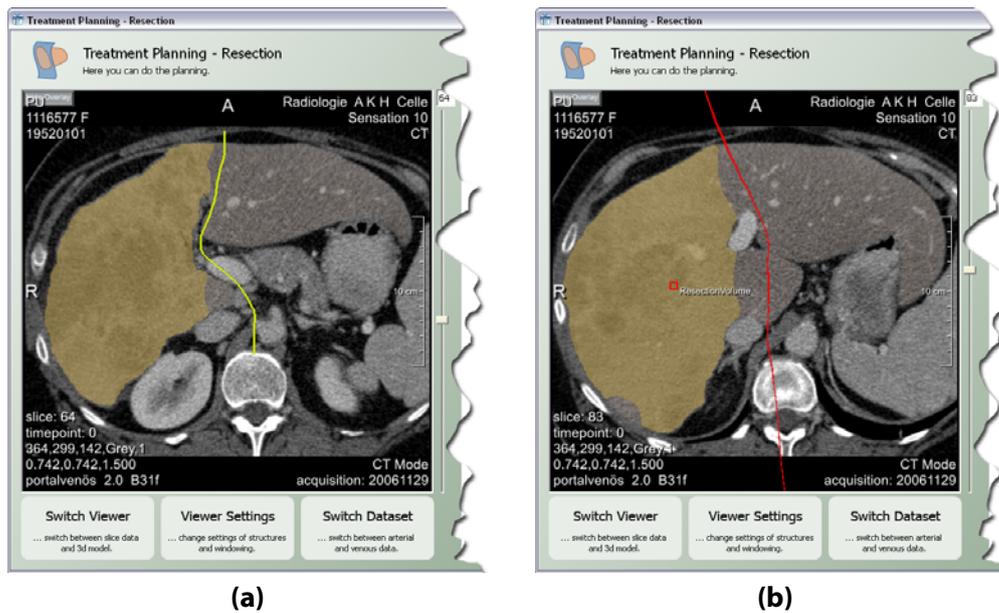


Abbildung 10.6: (a): Im Schritt der Therapieplanung kann der Lernende durch Einzeichnen des Verlaufs (gelb) in verschiedenen Schichtbildern eine Resektionsebene erzeugen lassen. (b): Diese Ebene kann nachträglich noch genau angepasst werden, z.B. um einen ausreichenden Sicherheitsrand zur Metastase zu gewährleisten.

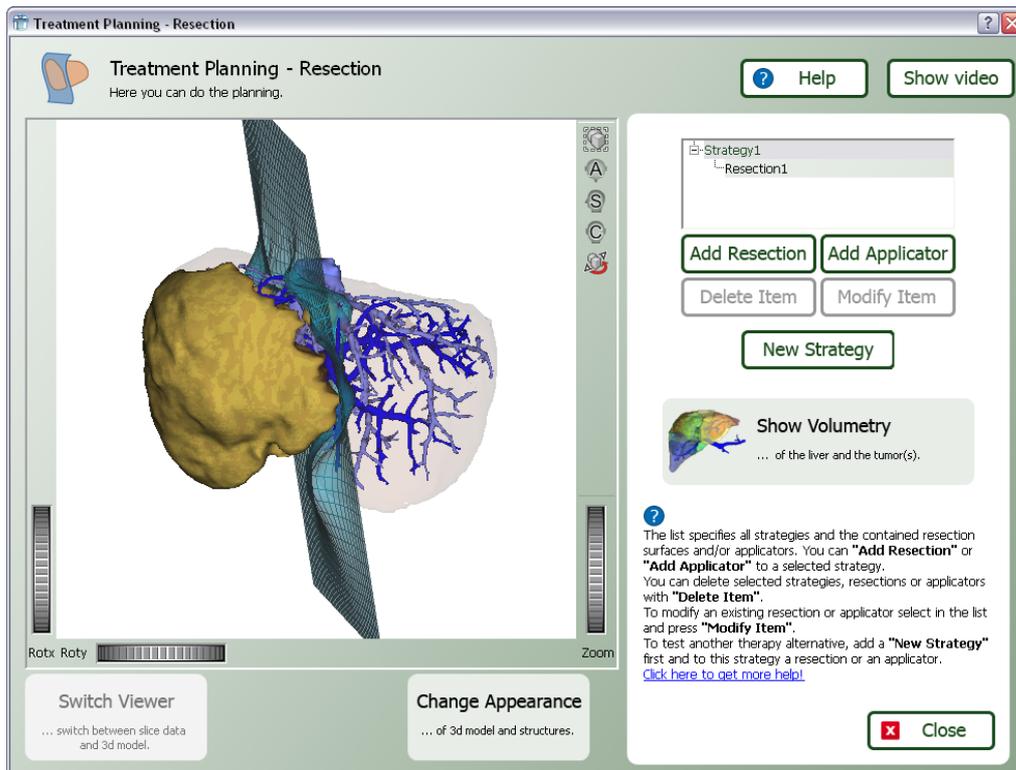


Abbildung 10.7: Darstellung einer geplanten Resektionsebene in 3D. Der Lernende kann die Szene explorieren und die Ebene im Kontext verschiedener Strukturen betrachten.

LIVERSURGERYTRAINER wird ein Applikator durch die Angabe des Zielpunkts in der Metastase sowie dem Eintrittspunkt im Körper des Patienten platziert. Aktuell werden Verfahren zur Simulation von Applikatorplatzierungen entwickelt (KRÖGER u. a. [2009], WELP u. a. [2006]), mit denen es dann möglich ist, die Zerstörung des Tumorgewebes durch die Hitzebringung grob vorherzusagen.

Analyse der Planung

Nach dem Abschluss der eigentlichen Planung kann der Lernende seine Lösung mit der eines oder mehrerer Experten vergleichen. Die Lösung der Experten ist für jeden Fall hinterlegt. Der Lernende kann zuvor mehrere verschiedene Strategien in Form von Varianten geplant haben, aus denen er eine auswählt. Der Vergleich mit einem Expertenvorschlag erfolgt rein visuell. Dabei werden dem Lernenden die 3D-Darstellung seiner Lösung und die des Experten parallel und synchronisiert visualisiert (Abb. 10.8). Ebenso kann der Lernende die Lösungen in den 2D-Darstellungen vergleichen. Unterfüttert wird dies mit Hinweisen des Experten, warum er so vorgegangen ist und wo aus seiner Sicht die kritischen Punkte lagen, die es zu beachten gilt. Eine Form der analytischen Bewertung von Resektionsflächen befindet sich derzeit in Entwicklung (HANSEN u. a. [2010]). Allerdings entfällt ein rein quantitativer Vergleich, da es für jeden Fall eine Vielzahl an möglichen Resektionsvorschlägen geben kann – je nachdem, welche chirurgische Schule man durchlaufen hat und welche Operationstechnik anzuwenden man in der Lage ist.

Operationsergebnisse

Als letztem Schritt werden dem Lernenden Ergebnisse der tatsächlich durchgeführten Operation des Patienten präsentiert. Dies können je nach Fall Videos von wichtigen Ereignissen während der Operation oder Fotos der entnommenen Gewebepreparate sein (Abb. 10.9). So bekommt der Lernende zum Abschluss einen guten Eindruck vermittelt, wie die Planung in der Operation umgesetzt werden konnte, welche Komplikationen auftraten und wo aufgrund von neuen Befunden in der Operation von der Planung abgewichen werden musste.

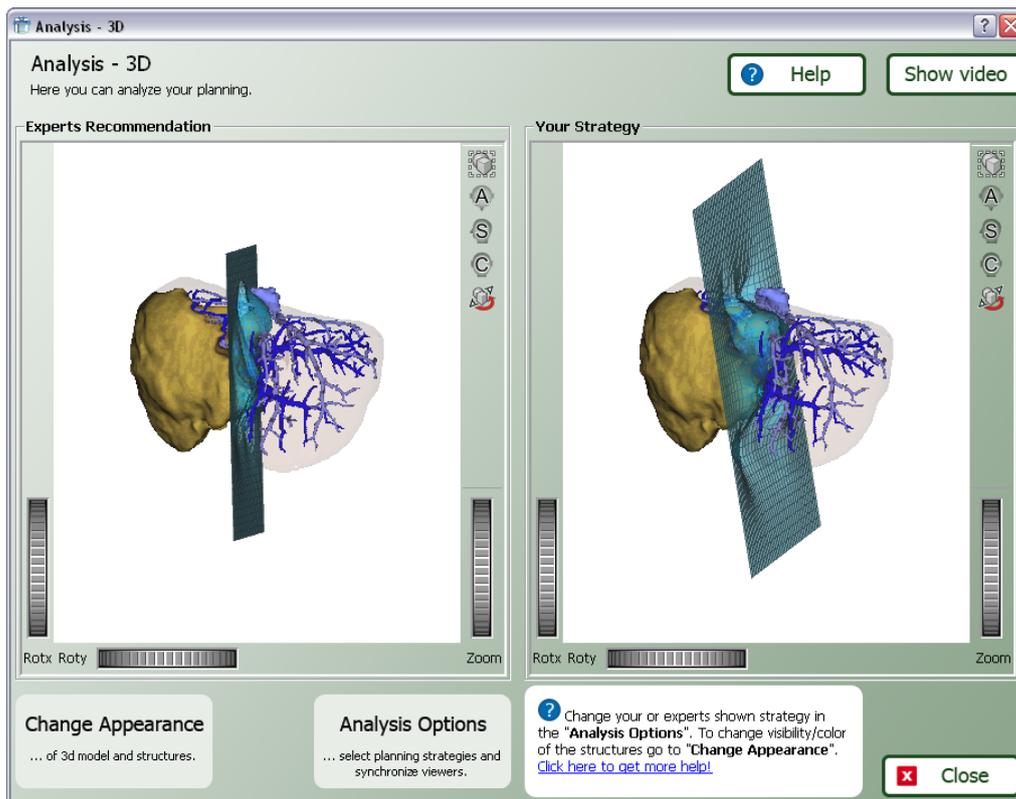


Abbildung 10.8: Zum Vergleich des eigenen Ergebnisses kann der Lernende seine Strategie optisch und textuell mit der eines Experten vergleichen. Die Viewer (in diesem Fall die 3D-Viewer) werden dabei synchronisiert, so dass der Lernende die beiden Darstellungen immer aus dem selben Blickwinkel betrachten kann.



Abbildung 10.9: Foto eines Leberresektates während einer Operation. Die anschließende histologische Untersuchung des Präparates ergab einen tumorfreien Saum von 2mm.

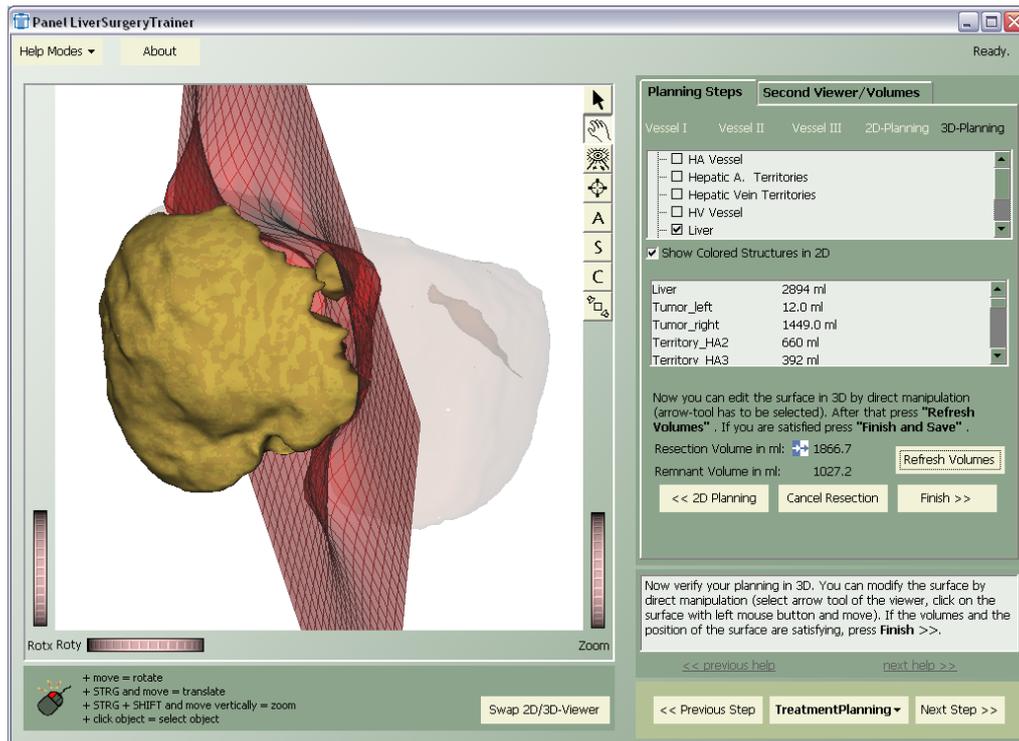


Abbildung 10.10: Screenshot der ersten veröffentlichten Version des LiverSurgery-Trainers. Diese Benutzeroberfläche wurde im Ergebnis der Evaluierung grundlegend überarbeitet und angepasst.

10.3 Evaluierung

Der LIVERSURGERYTRAINER wurde bisher auf der Basis eines ersten Prototyps evaluiert (siehe Screenshot in Abb. 10.10). Das Ziel der Evaluierung bestand in der Erlangung von Feedback für weitere Entwicklungsschritte. Im Detail sollte die Bedienbarkeit der Software sowie die didaktische Schlüssigkeit überprüft werden.

10.3.1 Durchführung der Evaluierung

Zielgruppe der Evaluierung waren im Gegensatz zur späteren Endversion des LIVERSURGERYTRAINERS nicht Ärzte in der Ausbildung, sondern erfahrene Viszeralchirurgen. Dieser Ansatz wurde aus zwei Gründen gewählt: Zum einen sollte so der generelle didaktische Aufbau und die Abfolge der Schritte durch Experten des Fachs in diesem Gebiet überprüft werden. Zum anderen waren die zu erwartenden Erfahrungen der Chirurgen, was die Bedienung der Software und den Umgang mit Planungssystemen anging, auf einem ähnlichen Niveau wie die der späteren Nutzer des LIVERSURGERYTRAINERS.

Der LIVERSURGERYTRAINER wurde auf einer DVD mit beiliegender Installationsanleitung an 32 Chirurgen weltweit verschickt,⁴⁸ die zuvor einer Teilnahme an der Evaluierung zugestimmt hatten. Neben der Installationsanleitung lagen auch umfangreiche Anleitungen zur Nutzung des Programms sowie zwei Beispielszenarien zur Behandlung kompletter Fälle in analoger und digitaler Form bei. Die verwendete Version des LIVERSURGERYTRAINERS enthielt zwei Fälle, einen onkologischen Fall und einen zur Kinderleberlebenspende.

Nach dem Durchlaufen eines bzw. beider Fälle wurden die Chirurgen gebeten, verschiedene Fragen zu beantworten. Dies konnte analog über ein beiliegendes Formular oder digital durch einen Fragebogen im Internet erfolgen. Das Design des Fragebogens orientierte sich dabei an einschlägigen Richtlinien (z.B. SHNEIDERMAN & PLAISANT [2004]). Der Fragebogen enthielt zu Beginn Fragen zur Person (Alter, Geschlecht) und zum Erfahrungsgrad im Umgang mit dem PC und 3D-Modellen. Anschließend wurden die Chirurgen zu ihrer Einstellung zur computergestützten Planung von Leberoperationen sowie zum Einsatz solcher Techniken in ihren Kliniken befragt. Im Hauptteil des Fragebogens wurden für jeden Schritt des LIVERSURGERYTRAINERS Bewertungen für folgende Aspekte verlangt:

- Verwendete Terminologie
- Angemessenheit für das Training
- Durch die Anwendung gegebenes Feedback
- Wartezeiten
- Aufgetretene Fehler
- Erlernbarkeit der Bedienung der Software
- Angemessenheit des Schrittes für die Erreichung des Lernzieles

Zu jeder Bewertung konnte optional ein zusätzlicher Kommentar abgegeben werden. Abschließend sollten die Testpersonen neue Ideen für weitere Entwicklungen des LIVERSURGERYTRAINERS bewerten, darunter den Einsatz von textuellen Annotationen.

⁴⁸Im Nachgang der Evaluierung wurde der LIVERSURGERYTRAINER an insgesamt ca. 25 Kliniken auf Anfrage verschickt.

10.3.2 Ergebnisse der Evaluierung

Es konnten 13 vollständig beantwortete Fragebögen erfasst werden (Rücklauftrate 41%). Dies erlaubt zwar keine statistisch aussagekräftige Analyse, jedoch konnte die Evaluierung wichtige und hilfreiche Hinweise für die weitere Entwicklung des LIVERSURGERYTRAINERS geben. Dies wird vor allem in der Anpassung des Layouts des LIVERSURGERYTRAINERS deutlich.

Analyse der Testpersonen

Die Evaluierung wurde von 11 männlichen und 2 weiblichen Chirurgen mit einem Durchschnittsalter von 32 Jahren (min.: 28 Jahre, max: 44 Jahre) absolviert. Zehn der 13 Testpersonen hatte mehr als 5 Jahre Erfahrungen in der Chirurgie. Die Erfahrungen im Bereich der Leberchirurgie waren heterogener: Erfahrungen von <2 Jahren, 2–5 Jahren und >5 Jahren waren gleich verteilt. Bis auf eine Testperson bewerteten alle ihre Erfahrungen im Umgang mit Computern als gut bzw. sehr gut. Ihre Erfahrung im Umgang mit 3D-Darstellungen war mannigfaltiger: Einige Testpersonen verfügten über große Erfahrungen in diesem Bereich, die meisten hatten jedoch nur mittlere Kenntnisse. Drei Viertel der Testpersonen führten selbst computerunterstützte Operationsplanungen durch.⁴⁹ Die Mehrheit davon vollzog die Planung in der eigenen Klinik. Einige nutzten dafür externe Dienstleister. Zwei Drittel der befragten Chirurgen waren sich sicher, dass die Computerunterstützung einen großen Vorteil bei der Operationsplanung bietet. Die Notwendigkeit eines Trainings wurde von allen Testpersonen als hoch bis sehr hoch eingeschätzt.

Generelles Feedback und die Beurteilung der Benutzerschnittstelle

Betrachtet man die geringe Zahl an ausgewerteten Fragebögen und die Wahl der Testpersonen,⁵⁰ so müssen die Ergebnisse sorgfältig unter diesen Gesichtspunkten analysiert werden. Auf einer Skala von (sehr gut) 1–7 (sehr schlecht) wurde der LIVERSURGERYTRAINER mit 2,6 (gut/zufriedenstellend) beurteilt. Der evaluierte Prototyp bildete damit eine gute Grundlage für weitere Entwicklungen. Die Bewertung der Nutzbarkeit (Usability) wurde mit „zufriedenstellend“ bewertet und bot ein großes Potential für zukünftige Entwicklungen. Aus der überraschend großen Menge an zusätzlichem Feedback über die Kommentarfunktion zu jeder Frage konnte der Wunsch vieler Testpersonen nach einer noch stärkeren Führung des Nutzers durch das gesamte Training entnommen werden. Auch wurde die Benutzerschnittstelle oft als zu verwirrend empfunden und eine klarere Hervorhebung der zur Verfügung stehenden Möglichkeiten und Funktionen wurde gewünscht. Auftretende Fehler wurden als wenig schwerwiegend eingeschätzt.

⁴⁹Dies ist nicht repräsentativ für die Gesamtheit aller Chirurgen.

⁵⁰Nicht alle Testpersonen konnten außerhalb des Projektumfeldes gewählt werden, da am Projekt selbst viele erfahrene Leberchirurgen teilnahmen.

Ein weiterer wichtiger Aspekt, der dem Feedback entnommen wurde, ist die mangelnde Performance des getesteten Prototypens. Es war anzunehmen, dass eine Verbesserung der Performance zu einer vergrößerten Akzeptanz des LIVERSURGERYTRAINERS führen würde.

Ein indirektes Ergebnis des gegebenen Feedbacks waren die kaum genutzten Hilfetexte. Es wurde bei der Entwicklung des LIVERSURGERYTRAINERS zwar auf eine kontextbezogene Anzeige der Hilfetexte Wert gelegt. Auch wurden viele Hilfetexte an mehreren Stellen (digitales Handbuch, in der Anwendung, in gedruckter Form) angeboten. Dennoch wurden sie nicht gelesen oder wahrgenommen.

Bewertung des Lernaufwands

Der Aufwand zur Erlernung der Bedienung des LIVERSURGERYTRAINERS wurde mit „vertretbar“ (3,3 auf einer Skala von (gering) 1–7 (sehr hoch)) bewertet, die Schwierigkeit dieses Lernprozesses mit 2,4 (nicht schwierig). Jedoch wurde in beiden Feldern, dem Aufwand und der Schwierigkeit der Erlernung der Softwarebedienung ein Potential für Verbesserung durch die Testpersonen gesehen. Das grundlegende Konzept des LIVERSURGERYTRAINERS konnte durch das Feedback der Testpersonen jedoch bestätigt werden.

Die Erstellung einer Resektionsebene wurde als die anspruchsvollste Aufgabe während der Bearbeitung eines Falles angesehen. Die verwendete Resektionsmethode (KONRAD-VERSE u. a. [2004]) wurde für ein Trainingssystem als wenig geeignet angesehen. Ein Problem stellte die für viele Testpersonen oft nicht nachvollziehbare automatische Generierung eines Ebenenvorschlags auf Basis der Nutzereingaben dar.

Bewertung zukünftiger Entwicklungen des LiverSurgeryTrainers

Den Testpersonen wurde als erste neue Technik ein simulierter Ultraschall in den 3D-Planungsdaten vorgeschlagen. Dies wurde als wenig hilfreich oder unwichtig bewertet. Im Gegensatz dazu wurde der Einsatz von textuellen Annotationen als sehr hilfreich angesehen. Viele Testpersonen fanden eine mögliche Zuordnung der Anatomie in der 2D-Darstellung durch Annotationen wünschenswert. Andere konnten diesem Aspekt keinen Mehrwert zugestehen. Die unterschiedliche Bewertung in diesem Punkt kann aus unterschiedlichen Erfahrungsgraden im Umgang mit radiologischen Bilddaten herrühren.

Die Testpersonen gaben als eigene Vorschläge für zukünftige Entwicklungen eine automatische Analyse der Gefäßversorgung nach einer Resektion an. Da solche Funktionalitäten in klinischen Planungssystemen vorhanden sind, sollten sie auch im Lernsystem verfügbar sein.

10.3.3 Konsequenzen aus der Evaluierung

Als Ergebnis der Evaluierung wurde der LIVERSURGERYTRAINER umfassend umgebaut. Der Fokus der Anpassungen lag dabei auf einer noch stärkeren Führung des Lernenden und einer größeren Klarheit der angebotenen Sichten der Benutzerschnittstelle. Das Training wurde in zusätzliche kleinere Schritte unterteilt. Es wurden Grundthesen für chirurgische Benutzerschnittstellen definiert (siehe den folgenden Abschnitt 10.4). Die Ergebnisse flossen in die Entwicklung anderer Trainingssysteme ein, wie z.B. den SPINESURGERYTRAINER (CORDES u. a. [2008]).

Es wurde versucht, das Resektionsverfahren grundlegend neu zu entwickeln – ein Prozess, der derzeit anhält. Dabei soll die Ebenenvisualisierung verbessert und die nachträgliche Manipulation der Ebenen nachvollziehbarer und intuitiver gestaltet werden. Für das Hilfesystem wurde nach neuen Ansätzen gesucht (siehe Abschnitt 10.4.3). Der Umbau des LIVERSURGERYTRAINERS hält derzeit an und soll mit einer umfassenden zweiten Evaluierung Mitte 2010 abgeschlossen werden. Aktuelle Zwischenevaluierungen auf informeller Ebene lassen aber schon erkennen, dass wesentliche Verbesserungen erreicht wurden.

10.4 Spezielle Aspekte von Benutzerschnittstellen chirurgischer Applikationen

Ein wesentliches Ergebnis der Evaluierung (siehe Abschnitt 10.3 und CORDES u. a. [2007a]) war die nur zufriedenstellende Qualität der Benutzerschnittstelle und der Benutzerführung. Im Zuge der ersten Spezifikation des LIVERSURGERYTRAINERS wurden im Bewusstsein der Entwicklung eines Lernsystems Entscheidungen für ein ansprecheres Design getroffen. Jedoch konnten die Überlegungen in der ersten Entwicklungsphase kaum umgesetzt werden. Dieser Umstand war zum einen einer gewissen Fokussierung auf die technische Realisierung der Software im Allgemeinen und der softwaretechnischen Beschränkung der Entwicklungsplattform hinsichtlich der gestalterischen Möglichkeiten im Speziellen geschuldet. Beides konnte im Nachgang der Evaluierung in einer zweiten Entwicklungsphase des LIVERSURGERYTRAINERS überwunden werden. Die Ergebnisse dieses Prozesses werden im Folgenden als eine Sammlung von Richtlinien und Empfehlungen zur allgemeinen Entwicklung chirurgischer Softwaresysteme unter dem Aspekt der Benutzerschnittstellen vorgestellt. Dabei fließen neben den Erfahrungen mit dem LIVERSURGERYTRAINER auch Erfahrungen aus der Entwicklung anderer chirurgischer Applikationen, wie dem NECKSURGERYPLANNER (JANKE u. a. [2006]) und dem SPINESURGERYTRAINER (CORDES u. a. [2008]), ein.

10.4.1 Style Guides in der Literatur

Richtlinien für die Gestaltung von Benutzerschnittstellen (Style Guides) sind weit verbreitet, da sie eine aktive Hilfestellung bei der Entwicklung (neuer) Applikationen geben oder helfen, verschiedenen Applikationen eines Unternehmens ein einheitliches Erscheinungsbild zu verleihen (GELB & GARDINER [1997]). Die freie Verfügbarkeit solcher Style Guides ist allerdings stark eingeschränkt, da sie meist strategisches Wissen und Erfahrungen eines Unternehmens darstellen, die es gilt, vor der Konkurrenz zu verbergen. Für große Unternehmen mit umfangreichen Software- oder Betriebssystemen sind öffentliche Style Guides jedoch unerlässlich. Sie sind auf eine breite Schicht verschiedenster Entwickler angewiesen, die eigene Entwicklungen an die des großen Unternehmens anpassen wollen und müssen. So gibt es Style Guides für die generelle Gestaltung von Applikationen unter Apples Betriebssystem MacOS (APPLE [2009a]) oder Microsoft Windows (MICROSOFT CORPORATION [1999]). SAP stellt einen eigenen Style Guide zur konkreten Gestaltung SAP R/3 konformer Oberflächen zur Verfügung (SAP [2001]) und die „Open Software Foundation“ definiert allgemeine Regeln zur Gestaltung grafischer Benutzeroberflächen im „Osf/Motif Style Guide“ (OPEN SOFTWARE FOUNDATION [1992]).

Vereinzelt lassen sich in der Literatur Style Guides für spezielle Anwendungsgebiete finden. So bieten z.B. CHARITON & CHOI [2002] eine Anleitung zur Gestaltung für Online-Buchungssysteme von Fluggesellschaften und KAPPEL u. a. [2006] Richtlinien für DVD-Menüs an.

Die verschiedenen Style Guides zeichnen sich zumeist durch eine sehr ausführliche textuelle Beschreibung der empfohlenen Gestaltungsmöglichkeiten aus. Anschauliche Beispiele fehlen oft oder werden nur sporadisch angeboten – obwohl dies gerade bei der Gestaltung von grafischen Oberflächen eine wesentliche Inspirationsquelle ist. SCHEMENAUR & PAWLICK [2007] haben dies umfassend für einen Teil der von Sun für Java-Applikationen genutzten Style Guides (SUN MICROSYSTEMS [2001]) untersucht und bei einer Befragung von Interface-Entwicklern bei SUN herausgefunden, dass diese die angebotenen Style Guides nur sporadisch nutzten, da das Übermaß an Text einen zu großen Leseaufwand erforderte. Es ist zu konstatieren, dass das Ziel eines Style Guides neben der Vermittlung grundlegender Eigenschaften bestimmter Applikationen (durch Text) auch ein großes Angebot an Beispielen und Varianten sein sollte, um so die Entwicklung neuer Benutzeroberflächen zu inspirieren.

Für den Bereich der Gestaltung medizinischer Applikationen sind einige Style Guides bekannt. So beschrieben GULLIKSEN & SANDBLAD [1995] das HELIOS-System zur Verwaltung von Patientendaten. Sie erweiterten dabei den im „Osf/Motif Style Guide“ (OPEN SOFTWARE FOUNDATION [1992]) ebenso wie ELLIOTT u. a. [1996] mit ihrem TOSCA-System. YODER u. a. [1998] beschrieben MEDIGATE als ein System zur Erfassung von Untersuchungen in der klinischen Routine. Sie konzentrierten sich dabei auf die Aspekte der Texterfassung, was ihre Beschreibungen nur begrenzt für die chirurgische Planung anwendbar macht. MONTGOMERY

u. a. [2004] versuchten Paradigmen zur Gestaltung von chirurgischen Planungssystemen aufzustellen. Leider lassen sich aus der Beschreibung eines längeren Projektes zur VR-basierten Planung keine auf andere Applikationen übertragbaren Schlüsse ableiten. Einzig die Erkenntnis, dass eine Zusammenarbeit unter den Chirurgen ein wichtiger Erfolgsfaktor einer Planung ist, mag für zukünftige Entwicklungen zu verallgemeinern sein. Ebenso gehen die Kriterien zur Gestaltung von Benutzeroberflächen chirurgischer Assistenzsysteme von ZHANG u. a. [2008] wenig über allgemeine Kriterien einer Oberflächengestaltung hinaus (z.B. die Verwendung einer geringen Anzahl von Farben und einfache, kurze Sätze in den textuellen Beschreibungen). ZHANG u. a. [2008] beschrieben vielmehr eine konkrete Applikation zur Planung und Operationsunterstützung von Eingriffen am Knie. Solche konkreten Beschreibungen sind dennoch bei der Entwicklung neuer Systeme hilfreich und können positive wie negative Anregungen zu eigenen Entwicklungen geben.

Parallel zu der vorliegenden Arbeit entstand eine weitere Arbeit (RITTER u. a. [2009]) speziell zur Gestaltung von Benutzeroberflächen im intra-operativen Umfeld. Dabei wurde neben der übersichtlichen Gestaltung der Oberfläche auch auf neue, berührungslose Interaktionstechniken Wert gelegt. Dies kann als Grundlage für Style Guides in diesem Teilbereich chirurgischer Anwendungen dienen.

10.4.2 Besonderheiten chirurgischer Benutzerschnittstellen

Die im Rahmen dieser Arbeit ermittelten Besonderheiten chirurgischer Benutzerschnittstellen und deren Nutzer werden im Folgenden beschrieben, um daraus Grundthesen für die Entwicklung passender Applikationen zu definieren. Diese werden abschließend anhand einiger konkreter Beispiele in ihrer Umsetzung vorgestellt (Abschnitt 10.4.3).

Analyse chirurgischer Software und Nutzergruppen

Blickt man auf das vorhandene Softwarespektrum im medizinisch/klinischen Bereich, so lässt sich der Vielzahl an Applikationen eine erkennbare Gruppierung entnehmen:

- Radiologische Workstations im Umfeld der Bildgebung (CT/MRT) und Befundung,
- Gerätegebundene Software, z.B. bei Ultraschallgeräten oder Patientenmonitoren der Anästhesie,
- Administrative Software zur Verwaltung und zum Krankenhaus-Management,
- Intraoperative Software, zumeist zur Navigationsunterstützung,
- Software zur Planung von chirurgischen Eingriffen (operative Fächer).

Anhand dieser groben Einteilung lässt sich vermuten, dass für die verschiedenen Softwaregruppen unterschiedliche Nutzgruppen existieren, die andere Anforderungen an die Bedienung stellen. So muss Software im intraoperativen Bereich besonders robust sein und über klar gegliederte Oberflächen mit nur wenigen Interaktionsmöglichkeiten verfügen – der Fokus im OP-Saal liegt auf dem Patienten und dem operativen Eingriff, nicht auf der Bedienung der Software.

Die Benutzerschnittstellen gerätegebundener Softwaresysteme sind meist eng mit der Hardware des Gerätes verbunden und unterliegen strengen Richtlinien. So müssen beispielsweise Anästhesie-Monitore die Vitalparameter eines Patienten klar darstellen. Die Schwierigkeit besteht in der heterogenen Anwendergruppe und der notwendigen Eingliederung solcher Systeme in ein komplexes Umfeld, ohne den einzelnen Nutzer mit den präsentierten Informationen zu überlasten (KINDSMÜLLER u. a. [2009]).

Radiologische Workstations sind für eine Nutzergruppe ausgelegt, die den Großteil ihrer täglichen Arbeitszeit mit den Applikationen verbringt. Demnach können sie sehr funktionsmächtig und komplex in ihrem Aufbau sein (Abb. 10.11). Die Lernphase für solche Software kann zwar länger andauern, bringt aber im täglichen massenhaften Einsatz letztendlich aber den erwünschten Effizienzgewinn.

Betrachtet man den noch recht neuen Bereich der chirurgischen Planungssoftware, so lassen sich mit Blick auf die Zielgruppe solcher Software folgende Charakteristika ermitteln:

- Chirurgen sind medizinische Experten mit oft nur geringen PC-Kenntnissen. Ihre maßgebliche Erfahrung sind chirurgische Fähigkeiten, wie der Einsatz chirurgischer Instrumente oder die Bedienung der Ausstattung eines OP-Saals. Chirurgen lassen sich im Allgemeinen nur ungern von diesem Fokus ablenken.
- Im Gegensatz zu anderen medizinischen Disziplinen, wie der Radiologie, kommt bei Chirurgen der PC nicht täglich, und wenn dann eher für administrative Aufgaben, zum Einsatz.
- Chirurgen profitieren besonders von dreidimensionalen Darstellungen, die die Anatomie des Patienten so wiedergeben, wie sie sie real vorfinden.
- Chirurgen weisen meist keine besonderen Fähigkeiten in der Handhabung von 2D- und 3D-Darstellungen auf.

Das Hauptziel chirurgischer Planungssoftware ist daher die Unterstützung des Chirurgen bei seiner eigentlichen Aufgabe, der Planung und Durchführung von chirurgischen Eingriffen. Der dabei zu fördernde Entscheidungsfindungsprozess kann exemplarisch folgende Entscheidungen betreffen:

- „Ist der Patient operabel oder gibt es andere Behandlungsoptionen?“
- „Welche Interventionsstrategie ist die erfolgversprechendste?“
“Welche Strategie ist die sicherste?“

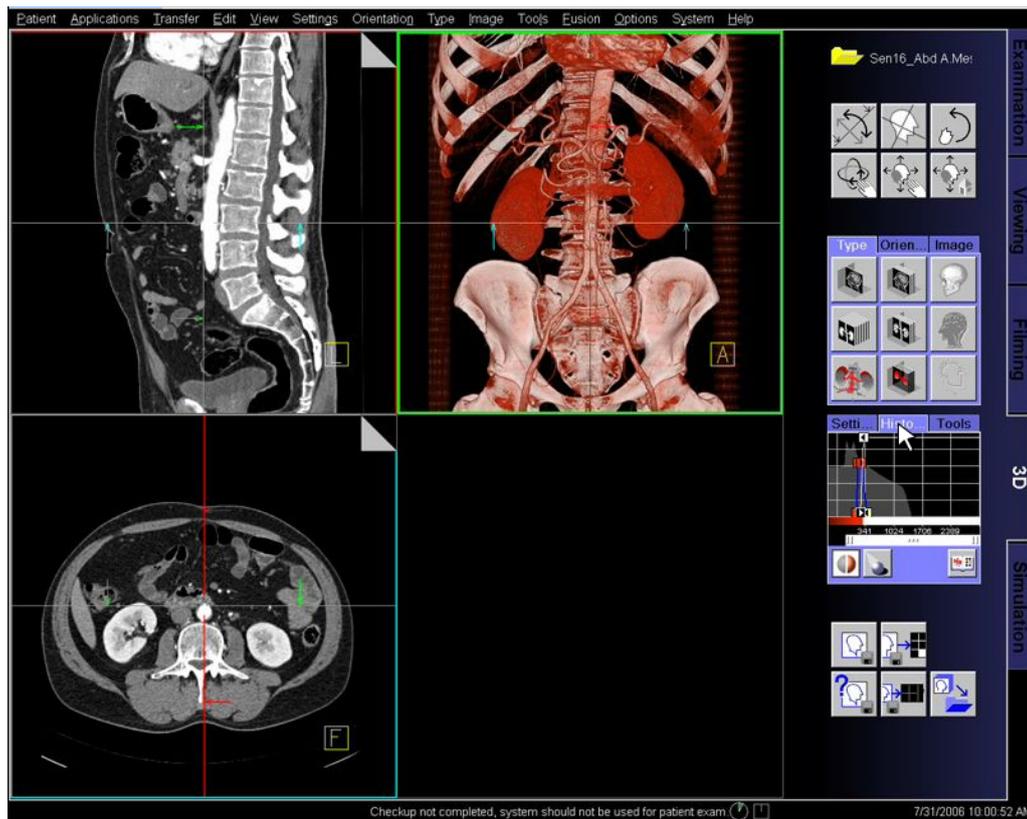


Abbildung 10.11: Screenshot der radiologischen Applikation SOMATOM syngo CT 2007 von Siemens. Die Anwendung ist sehr komplex in ihrem Aufbau. Dem Radiologen wird eine Vielzahl an Funktionen und Möglichkeiten zur gleichzeitigen Auswahl angeboten.

- „Welche Handlungen und Werkzeuge müssen für eine Operation vorbereitet werden?“

Sehr konkret möchte ein Chirurg durch die Nutzung von Planungssoftware beispielsweise Antworten auf Fragen wie:

- „Ist das Gefäß von Tumorgewebe infiltriert?“
- „Wie wirkt sich die Größe des Sicherheitsrandes auf die Versorgung wichtigen Gewebes aus?“
- „Wie groß ist der Abstand zwischen Lymphknoten und Gefäß?“

Chirurgische Applikationen sollten die Beantwortung solche Fragen direkt unterstützen.

Wiederkehrende Aufgaben

Aus der Analyse der Anwender und Aufgaben chirurgischer Planungssoftware lassen sich eine Reihe oft wiederkehrender und damit allgemeingültige Funktionen solcher Software ermitteln:

3D-Exploration: Chirurgen explorieren die Patientendaten zumeist in dreidimensionalen Darstellungen, um so einen optimalen Eindruck der räumlichen Verhältnisse zu bekommen. Dabei rotieren und verschieben sie die Szene, zoomen zu wichtigen Strukturen und blenden Strukturen ein oder aus.

Exploration einzelner Strukturen: Einzelne wichtige oder kritische Strukturen werden zumeist im Detail analysiert. Dies kann ein Tumor, ein exponiertes Gefäß oder zu resezierendes Gewebe sein. Der Chirurg ist dabei an Abmessungen und Volumen sowie kritischen Strukturen der Umgebung interessiert.

Annotationen: Textuelle Annotationen helfen dem Chirurgen sowohl in 2D- wie in 3D-Darstellungen, Strukturen zu identifizieren. Zusätzlich stellen eigene Annotationen, die der Chirurg selbst anbringen kann, eine große Hilfe dar.

Dokumentation: Die Dokumentation des Planungsprozesses sowie der getroffenen Entscheidungen stellen eine wesentliche Grundlage nicht nur für folgende interdisziplinäre Diskussionen dar, sondern auch für die Patientenakte.

Visualisierungs-Presets: Da Chirurgen meist nur wenige einzelner Parameter einstellen möchten, um zu einer gewünschten Visualisierung zu gelangen, sind Voreinstellungen (Presets) hilfreich. Solche Presets müssen wohl überlegt und an die Bedürfnisse, Fragestellungen und Anwendungskontexte angepasst sein, liefern dann aber schnelle Ergebnisse und erlauben die Konzentration des Chirurgen auf die wesentlichen Aspekte einer Szenenexploration.

Hilfe: Da Chirurgen verhältnismäßig selten mit Planungssoftware in Kontakt kommen (im Vergleich zu ihrer eigentlichen OP-Tätigkeit und im Vergleich zu Radiologen u.ä.) ist ein gut durchdachtes Hilfefkonzept erstrebenswert. Dabei

sind vor allem kurze, prägnante Hilfen denkbar, die aufgabenspezifisch genau an den Stellen erscheinen, an denen sie gebraucht werden. Im Kontrast dazu sind lange Hilfetexte oder gar Handbücher, die es erst aufwendig zu studieren gilt, abzulehnen.

Grundthesen für chirurgische Applikationen

Aus der bisherigen Erfahrung mit der Entwicklung chirurgischer Applikationen sowie der vorangegangenen genauen Analyse der Anwender, Ziele und wiederkehrenden Aufgaben lassen sich zwei Leitlinien für die Entwicklung chirurgischer Applikationen ableiten:

„Weniger Steuerelement sind mehr“

Viele Applikationen sind oft mit Steuerelementen, vielen Toolbars, Kontextmenüs und Tastaturkürzeln geradezu überfrachtet. Im Gegensatz zu solchen Applikationen, die bspw. von Radiologen sehr erfolgreich genutzt werden, bevorzugen Chirurgen einfachere, aufgeräumte Oberflächen mit weniger Steuerelementen.

„Weniger Flexibilität, mehr Führung“

Viele Applikationen bieten dem Nutzer die maximale Flexibilität, indem er nahezu alle Funktionen zu jeder Zeit in jedem Teil der Applikation aufrufen und nutzen kann. Dieser Ansatz ist bei chirurgischen Applikationen durch ein wesentlich strukturierteres Schritt-für-Schritt-Verfahren, welches dem Chirurgen klar die Reihenfolge der durchzuführenden Methoden vorgibt, zu ersetzen.

Diese Leitlinien können unabhängig von der konkreten Planungsaufgabe oder dem Applikationskontext zur Anwendung kommen. Sie sind des Weiteren sowohl in Applikationen der klinischen Routine wie auch in Lernsystemen zur Ausbildung von Chirurgen gut umsetzbar und führen zu mutmaßlich besseren Anwendungen.

10.4.3 Umsetzung verschiedener exemplarischer Lösungen für chirurgische Benutzerschnittstellen

Die in den folgenden Abschnitten präsentierten Beispiele für einzelne Aspekte chirurgischer Benutzerschnittstellen entstammen einer Vielzahl von Gesprächen und Diskussionen mit Chirurgen und anderen Ärzten. Einige der Ideen gehen direkt auf Vorschläge der Anwender zurück, andere wurden zunächst als Studie entwickelt und dann den Chirurgen in einem Set von Möglichkeiten präsentiert. Wie schon in Kapitel 9 angedeutet, stellen solche Diskussionen auf der Basis von Studien und Prototypen ein Kernelement der Entwicklung chirurgischer Software dar. Im Anschluss wird ein Einblick in die designerische Entwicklungsarbeit gegeben.

Vorgegebener Ablauf

Die Vorgabe eines bestimmten Ablaufs der zu absolvierenden Schritte bei der Bearbeitung eines Problems innerhalb einer Applikation zählt zu den Kerneigenschaften chirurgischer Applikationen. Die Vorgabe solcher Workflows ist nicht nur in Trainingsanwendungen äußerst dienlich, sondern kann auch einem erfahrenen Arzt in einer Applikation der täglichen Routine durch die Führung eine gewisse Routine und Sicherheit bieten, nichts Wesentliches zu vergessen (Abb. 10.12). Die Ausprägung der in der Softwareentwicklung oft als „Wizards“ bezeichneten Schritt-für-Schritt-Systeme kann sich an verschiedenen Punkten orientieren.

So kann es strikte Auslegungen geben, die ein Aufrufen des nächsten Schrittes erst nach einer vollständigen Abarbeitung des aktuellen Schrittes (z.B. der Klassifizierung eines Tumors) erlauben. Solche strikten Systeme bieten sich vor allem für Trainingssysteme wie den LIVERSURGERYTRAINER an, da hier ein noch unerfahrener Nutzer wirklich alle Aspekte einer Planung ausführlich lernen und absolvieren soll. Ein Durchführen der Resektion ist bspw. aus didaktischer Sicht wenig sinnvoll, wenn zuvor nicht die Gefäßanomalien in der Leber bestimmt wurden.

Weniger strikt sollte die Einhaltung der Abläufe bei Applikationen für erfahrene Anwender verfolgt werden. Hier kann die Vorgabe eines Ablaufs eine Hilfe sein, jedoch sollte der erfahrene Nutzer immer zwischen den Schritten seiner Wahl springen und Schritte überspringen können (Abb. 10.12). Die Erfahrung zeigt aber, dass sich auch erfahrene Anwender an die vorgegebenen Schritte halten, so diese denn sinnvoll und gut an den Arbeitsablauf des Chirurgen angepasst sind.

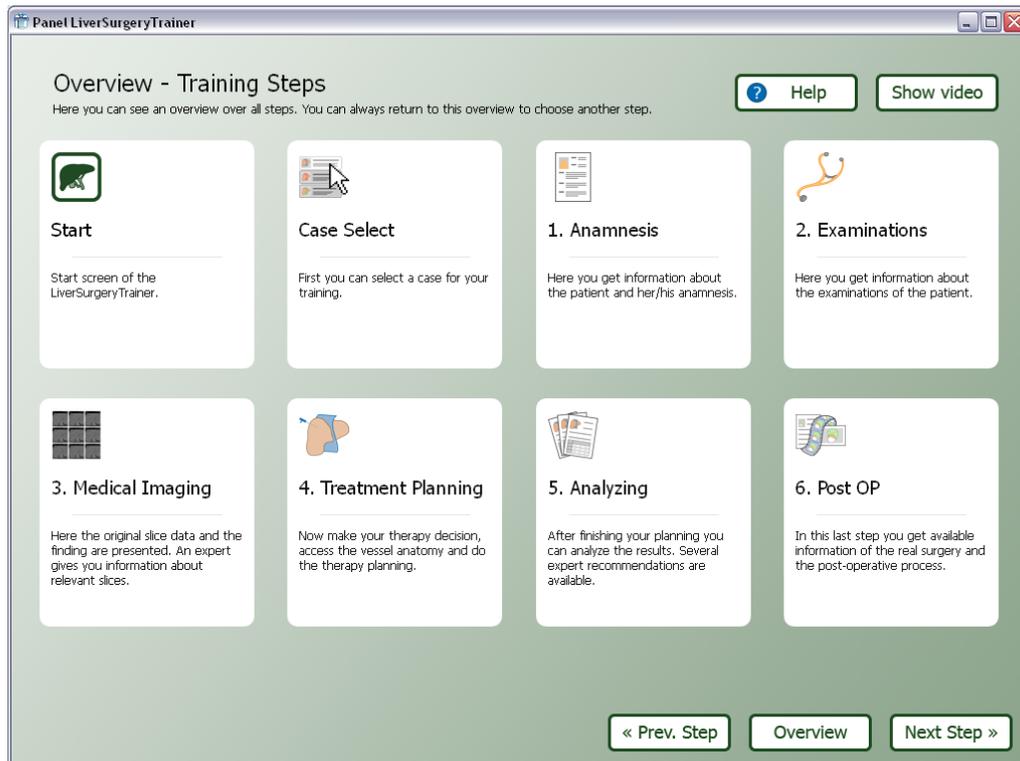


Abbildung 10.12: Der Lernende wird Schritt für Schritt durch das Trainingssystem geführt. Er kann dabei Schritte überspringen oder wiederholen.

Wenige Steuerelemente

Eine Designziel, welches in den Grundthesen in Abschnitt Abschnitt 10.4.2 festgehalten ist, ist die extreme Minimierung der verfügbaren Steuerelemente in einer Ansicht. Der schrittweise Aufbau reduziert die Anzahl der zur Verfügung stehenden Steuerelemente schon erheblich. Für ein positives Beispiel sei auf die Applikation „Interactive 3D Liver Anatomy“ in Abb. 2.17 auf Seite 34 verwiesen. Ein negatives Beispiel ist mit dem MEVISLIVERANALYZER in Abb. 2.13 auf Seite 25 zu sehen. Die Anzahl der Steuerelemente kann durch gezieltes Design weiter reduziert werden. So werden bspw. beim LIVERSURGERYTRAINER in einer 2D-Viewer-Ansicht (Abb. 10.13) die Steuerelemente für das Ein- und Ausblenden von Strukturen, für die Fensterung oder die Auswahl des Phase des CT nur auf Anfrage und nie parallel eingeblendet. Die Anzahl der Steuerelemente, die immer verfügbar sind, ist auf ein Minimum reduziert (Schaltflächen für die Hilfe sowie zum Navigieren in den Schritten u.ä.).

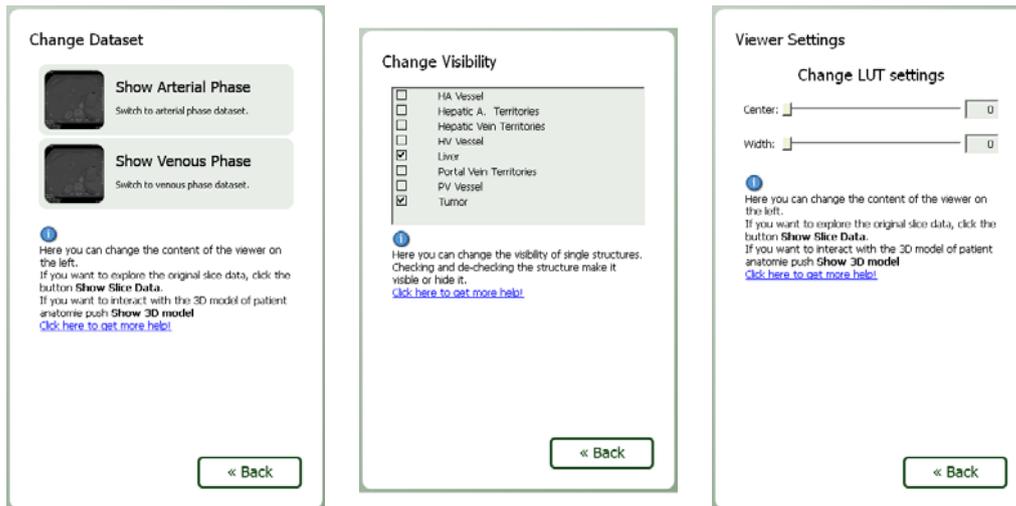


Abbildung 10.13: Drei verschiedene Bereiche zur Auswahl der CT-Phase, zum Einblenden von zusätzlichen Strukturen und zur Fensterung der 2D-Ansicht. Jeder Bereich wird nur auf ausdrücklichen Wunsch des Nutzers eingeblendet und auch nur allein stehend, was die parallele Ansicht mehrerer Konfigurationsbereiche ausschließt.

Große Schaltflächen und viel Freiraum

Der durch die stark reduzierte Anzahl an Steuerelementen gewonnene Freiraum kann durch größere, aussagekräftigere Schaltflächen genutzt werden. So können neben größeren Symbolen auch kurze, erklärende Texte in den Schaltflächen (Abb. 10.14(a)) die Erwartung des Nutzers wecken oder ihn auf die kommenden Sichten vorbereiten, als dies z.B. sehr kleine Schaltflächen mit nur einem Symbol vermögen (Abb. 10.14(b)).

Unterstützung bei der 3D- und 2D-Navigation

Ein wichtiger Aspekt bei der Gestaltung von chirurgischen Applikationen ist die Unterstützung des Nutzers bei der Interaktion mit den komplizierten Viewern – sowohl 2D als auch 3D. Neben den in dieser Arbeit vorgestellten komplexeren Techniken, wie der automatischen Sichtpunktbestimmung (Kapitel 4) oder Annotation von Strukturen (Kapitel 7), kann aus designerischer Sicht einiges zur Unterstützung des Nutzers beigetragen werden. So ist generell auf den Einsatz von ansonsten in diesem Kontext sehr beliebten Tastaturkürzeln oder komplizierten Tastatur-Maus-Kombinationen zu verzichten. Auch ist einem Nutzer nur schwer zu vermitteln, eine Vielzahl an Tastatur-Maus-Kombinationen zu erlernen (siehe Tabelle 10.1). Der Versuch in der ersten Version des LIVERSURGERYTRAINERS, durch prominente Hinweise und Anleitungen die Handhabung der Viewer über die üblichen Tastatur-Maus-Kombinationen zu vermitteln, schlug fehl, weil a) die Hinweise trotz ihrer Prominenz nicht wahrgenommen wurden und b) sich die



(a)



(b)

Abbildung 10.14: (a): Die Reduzierung der Zahl der Steuerelemente ermöglicht größere Schaltflächen mit aussagekräftigen Texten und Icons. (b): Kleine Schaltflächen in Kombination mit schwer zu erkennenden Schaltflächen, wie hier aus dem MeVisLiverAnalyzer, verwirren den Nutzer und lassen ihn im Unklaren ob der zur Verfügung stehenden Funktionen.

	Interaktionsart	Vorgenommene Veränderung
3D-Viewer	linke Maustaste + Mausbewegung	Rotation
	linke & rechte Maustaste + Mausbewegung	Zoom
	mittlere Maustaste + Mausbewegung	Verschiebung
2D-Viewer	Mausrad	durch die Schichten scrollen
	mittlere Maustaste + Mausbewegung	schnell durch die Schichten scrollen
	STRG + mittlere Maustaste + Mausbewegung	in eine Schicht zoomen
	SHIFT + mittlere Maustaste + Mausbewegung	eine Schicht verschieben
	rechte Maustaste + horizontale Mausbewegung	Verändern der Fensterbreite der Transferfunktion
	rechte Maustaste + vertikale Mausbewegung	Verändern des Fensterzen- trums der Transferfunktion

Tabelle 10.1: Die hier dargestellten Kombinationen aus Tastaturkürzeln und Mausbewegungen sind in den 2D- und 3D-Viewern in MeVisLab nötig, um mit den entsprechenden Darstellungen interagieren zu können.

Anwender schnell mit der aus anderen Applikationen bekannten Unzulänglichkeit der Viewer zufrieden gaben und bestenfalls durch Probieren versuchten, eine Möglichkeit zu der von ihnen gewünschten Interaktion zu finden.

Um die Interaktion in den 3D-Viewern zu vereinfachen, bietet sich eine eindeutige Auswahl und Kennzeichnung der jeweiligen Modi Rotation, Verschiebung und Zoom in 3D und Scrollen, Fensterung, Zoom und Verschiebung in 2D an (Abb. 10.15). Durch die Auswahl genau eines Modus aus einer grafischen Darstellung (und nicht über versteckte Tastaturkombinationen) wird dem Nutzer zum einen verdeutlicht, in welchem Modus er sich gerade befindet (nach [NIELSEN \[1993\]](#) ein fast zwingendes Erfordernis). Zum anderen wird durch die grafische Darstellung dem Nutzer die Verfügbarkeit und gesamte Bandbreite der Möglichkeiten zur Interaktion dargelegt.

Als vorteilhaft hat sich bei den 2D-Viewern ebenfalls erwiesen, die verschiedenen Interaktionsmöglichkeiten auf konkrete Steuerelemente zu projizieren. Dabei kommt z.B. ein Schieberegler für das Scrollen zwischen den Schichten zum Einsatz. Dieser ermöglicht das schnelle Springen zu einer Zielschicht und gibt

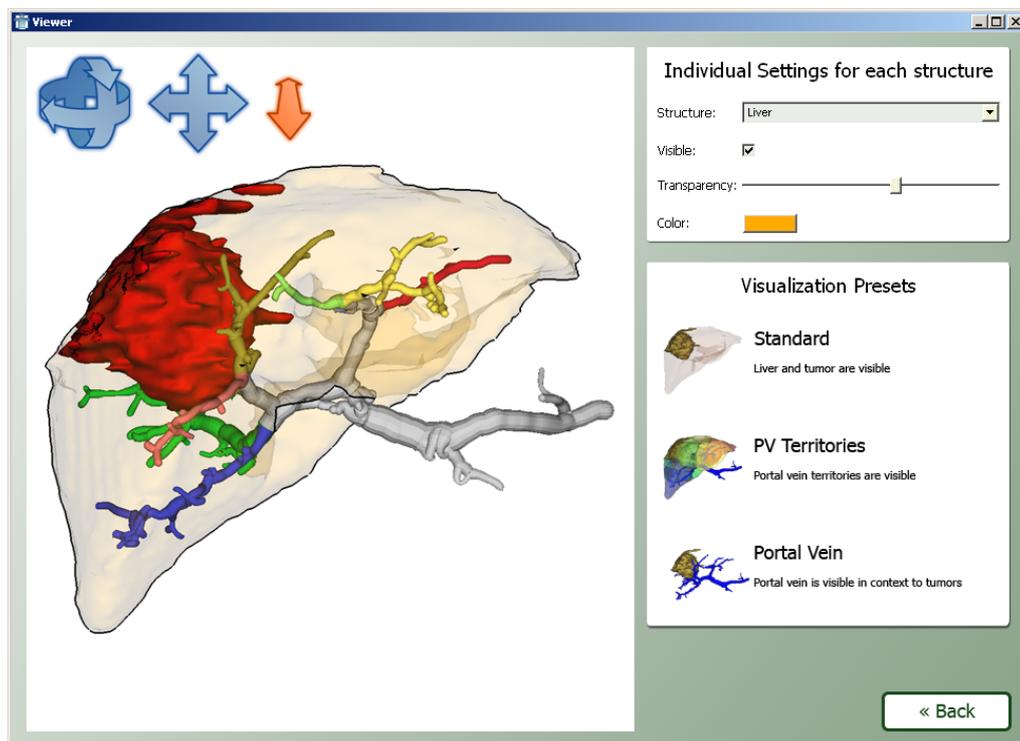


Abbildung 10.15: Konkrete Vorgabe der drei möglichen Modi zur Interaktion in einem 3D-Viewer einer kleinen Demo-Applikation für Chirurgen: Rotation, Verschiebung und Zoom. Durch Auswahl des Nutzers kann genau ein Modus aktiviert und genutzt werden. Es sind keine zusätzlichen Tastaturkürzel oder Kombinationen von Maustasten notwendig.

gleichzeitig eine Ahnung der Ausmaße des Datensatzes, was normale 2D-Viewer so bisher nicht konnten. Gleichzeitig kann die Einstellung der Parameter der Transferfunktion (Fensterbreite und Fensterzentrum) über Schieberegler transparenter und nachvollziehbarer gestaltet werden (Abb. 10.13).

Neue Wege bei der Hilfe

Die Gestaltung der Hilfen für eine chirurgische Applikation kann sich aufgrund der Zeitknappheit der im klinischen Alltag eingespannten Nutzer nicht an den immer noch häufig verwendeten Handbüchern oder textuell geprägten Hilfen anderer Software orientieren. Vielmehr sind hier innovative neue Lösungen anzustreben, die eine direkte Nutzung der Software ohne größere Einarbeitung ermöglichen. Eine längere Einarbeitungszeit muss gerade auch für Trainingsprogramme entfallen, deren primäres Ziel nicht der Umgang mit der Software selbst, sondern das Training anderer Fähigkeiten ist.

Im Rahmen der ersten Version des LIVERSURGERYTRAINERS wurden verschiedene Methoden der kontext- und nutzersensitiven Einblendung von Hilfetexten erprobt. Dabei wurde ein Hilfetext dann eingeblendet, wenn der Nutzer an einer bestimmten Stelle des Programms nicht weiterkam (auf Anforderung) oder wenn ihm ein neuer Bereich präsentiert wurde, den er so bisher noch nicht kennen konnte (automatische Hilfe). Gerade die Präsentation von automatischen Hilfetexten prominent direkt in den Sichten wurde jedoch von den Nutzern nicht angenommen und größtenteils ignoriert. Dies ergaben nachträglich durchgeführte Befragungen aber auch viele Fragen, die im Zusammenhang mit der Nutzung des LIVERSURGERYTRAINERS gestellt wurden und die durch automatisch angezeigte kurze Hilfetexte eigentlich hätten beantwortet gewesen sein sollen.

Das lässt auf eine nahezu vollständige Ausblendung jeglicher textuellen Hilfestellungen schließen, so sie nicht aktiv vom Nutzer angefragt werden. Dem kann in gewissen Grenzen durch eine noch prominentere Darstellung der Hilfetexte begegnet werden – z.B. durch größere Schriften, mit Screenshots unterlegte Hilfetexte oder eine zwingende Einblendung, die den Nutzer erst nach einer gewissen Zeit den Hilfetext „wegklicken“ lässt. Anstelle einer solchen recht tiefgreifenden Gängelung des Nutzers kann man Hilfen aber auch attraktiver gestalten. In den letzten Jahren haben sich im Zuge der fortschreitenden Ausbreitung von Onlinevideos Videotutorials als ein nützliches Instrument zur Vermittlung von Bedienkonzepten und einzelnen Problemlösungen herausgestellt (TEMPELMAN-KLUIT [2006], OUD [2009]). Videotutorials können auf den aktuellen Schritt der Applikation zugeschnitten sein und einen Nutzer bei der Lösung einer genau spezifizierten Aufgabe unterstützen (sogenannte *screen casts*). Die visuelle Vermittlung von Abläufen und Interaktionsreihenfolgen hat sich (zumindest beim LIVERSURGERYTRAINER) als ein wesentlich wirksameres Hilfsmittel im Vergleich zu rein textuelle Darstellungen erwiesen.

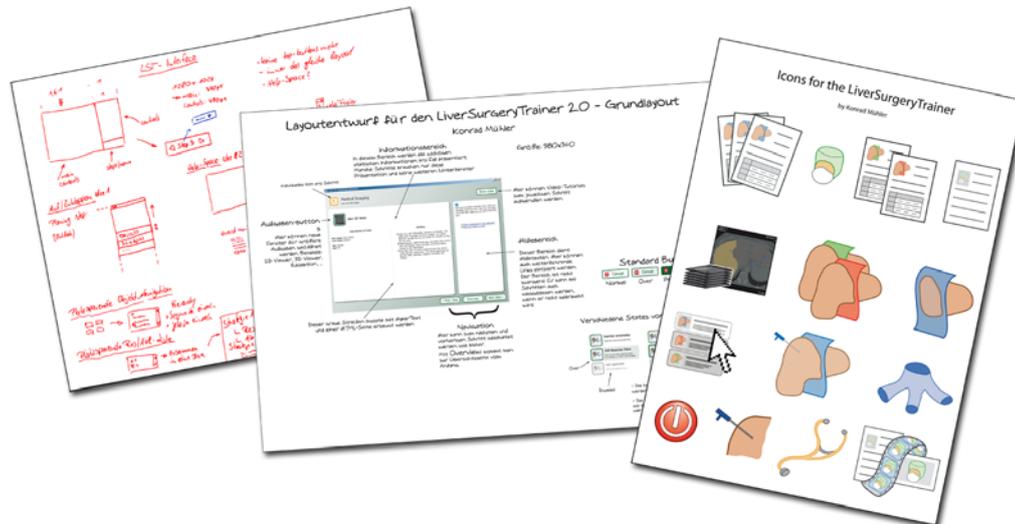


Abbildung 10.16: Designentwürfe für das Layout des LiverSurgeryTrainers – von ersten Skizzen (links) über Konzeptdarstellungen und -studien zur intensiven Diskussion bis hin zum individuellen Design der Icons für alle Bereiche und Schaltflächen des LiverSurgeryTrainers.

Vorgehen bei der Designentwicklung

Vor der ersten prototypischen Umsetzung der verschiedenen Ideen stand zunächst die skizzenhafte Darstellung verschiedener Entwürfe als Diskussions- und Entscheidungsgrundlage (Abb. 10.16). Dabei konnte festgestellt werden, dass allein die intensive Beschäftigung mit sehr kleinen Bereichen der gesamten Benutzerschnittstelle zu neuen Ideen und Ansätzen führte. Ein Merkmal, welches zu einem konsistenten und in sich schlüssigen Designentwurf beiträgt, ist die Gestaltung der Symbole (Icons). Um ein durchweg gleiches Erscheinungsbild der Symbole zu garantieren, wurden alle per Hand entsprechend der ihnen zugeordneten Aufgabe gezeichnet und koloriert (Abb. 10.16). Vorgefertigte Symbolbibliotheken oder gar aus verschiedenen Bereichen „gesammelte“ Symbole können ein solch schlüssiges Gesamtbild nicht erzeugen und scheitern zumeist an den sehr speziellen Anforderungen der individuellen Applikation (z.B. ein Symbol für „Resektionsebene hinzufügen“).

10.5 Zusammenfassung

In diesem Kapitel wurde der LIVERSURGERYTRAINER vorgestellt. Neben der Beschreibung der Analysen der Aufgaben und Zielgruppe wurde der schrittweise Aufbau des Trainingsprogramms dargestellt. Aus einer Evaluierung der ersten Version des LIVERSURGERYTRAINERS resultierend wurde im Besonderen auf die grafische Gestaltung der Benutzerschnittstelle genauer eingegangen. In diesem

Zusammenhang konnten Grundthesen für die generelle Entwicklung chirurgischer Applikationen hergeleitet werden. Diese wurden anhand einiger konkreter Beispiele des LIVERSURGERYTRAINERS vorgestellt, können in ihrer Generalität ebenso für andere chirurgische Applikationen Anwendung finden.

Ausblickend sei darauf verwiesen, dass der LIVERSURGERYTRAINER aktuell in einer zweiten Version entwickelt wird. Diese wird die Ergebnisse der Evaluierung, speziell die neue Benutzerschnittstelle, sowie wesentlich mehr Fälle enthalten und voraussichtlich Mitte 2010 fertiggestellt sein.

11

Zusammenfassung und Ausblick

Der Einsatz immer besserer medizinischer Visualisierungen in der chirurgischen Operationsplanung gewinnt immer mehr an Bedeutung. Ziel der Arbeit war daher die Entwicklung neuer Techniken zur Unterstützung des chirurgischen Operationsplanungsprozesses. Der Fokus lag hierbei auf Techniken der 3D-Exploration. Es konnte ein neuartiges Verfahren zur automatischen Bestimmung von Sichtpunkten in Szenen anatomischer Strukturen entwickelt und validiert werden. Dabei wurde zum ersten Mal semantisches Wissen neben rein geometrischen Informationen der Szene genutzt, um Sichtpunkte an die Bedürfnisse des Nutzers im jeweiligen Anwendungskontext anzupassen. Die durch das Verfahren der automatischen Sichtpunktbestimmung geschaffenen Grundlagen wurden genutzt, um zum Beispiel automatisch gute Kamerapfade zur Exploration anatomischer Szenen zu berechnen oder ähnliche Sichtpunkte zu clustern.

Zur Darstellung zusätzlicher Informationen in Visualisierungen wurden zwei- und dreidimensionale Szenen mit textuellen Annotationen versehen. Die dabei entwickelten Techniken erlauben die automatische Annotation beliebiger Strukturen mit einer großen Bandbreite an verschiedenen Stilen. Die so in die Darstellungen eingebrachten Informationen können dem planenden Chirurgen wichtige Hinweise auf kritische Strukturen liefern. Einem Lernenden in einem computer-gestützten Trainingssystem bieten Annotationen eine Hilfe bei der Erkennung und Zuordnung anatomischer Strukturen in den verschiedenen Darstellungen.

Als weiterer unterstützender Aspekt der chirurgischen Planung wurden dreidimensionale Animationen von Patientenmodellen untersucht. Dabei wurde ein Framework zur automatischen Generierung solcher Animationen entwickelt. Mit den Keystates wurde ein Konzept vorgestellt, das vor allem den komplexen Prozess der Definition von Animationsinhalten vereinfacht und einmal getroffene Visualisierungsentscheidungen wiederverwendbar macht.

Anwendung fanden die entwickelten Techniken in einem neuartigen chirurgischen Trainingssystem, dem LIVERSURGERYTRAINER. Durch dessen Entwicklung konnten viele der Techniken in einem realen Umfeld validiert werden, was zu zahlreichen Anpassungen führte. Die im Rahmen der Entwicklung gewonnenen Erkenntnisse bezüglich der Gestaltung von Benutzeroberflächen für chirurgische Anwendungen konnte in einen ersten Style-Guide für spätere Arbeiten in diesem Bereich überführt werden.

Es wurde ein spezielles Toolkit für die verbesserte Erstellung chirurgischer Applikationen entwickelt – das Medical Exploration Toolkit. Zusammen mit der Entwicklungsplattform MeVisLab ist es mit dem METK möglich, nutzerfreundliche Applikationen für den klinischen Einsatz schnell und effektiv zu erstellen. Dabei kann der Funktionsumfang einer Applikation aus einem breiten Angebot aus verfügbaren Visualisierungs- und Explorationstechniken individuell zusammengestellt werden.

11.1 Kritische Anmerkungen und Ausblick

Im Folgenden soll auf zwei besondere Aspekte, die im Rahmen dieser Arbeit nicht vollständig oder gar nicht behandelt wurden, näher eingegangen werden: die Evaluierung von Animationen und deren Authoring sowie die Nutzung der entwickelten Techniken im Bereich neu entstehender internetbasierter Anwendungen.

11.1.1 Internetbasierte Dienste

In den letzten Jahren ist eine Entwicklung zu beobachten, bei der viele Anwendungen im medizinischen Bereich von dezidierten Workstations auf internetbasierte „Web Services“ verlagert werden. Als Beispiel seien hier die vielfältigen PACS-Stationen⁵¹ (ASSMANN & HOEHNE [1983]) oder webbasierte Lernsysteme (siehe Seite 31) zu nennen. Der Vorteil solcher Dienste liegt zum einen in der örtlich unabhängigen ständigen Verfügbarkeit: Die Daten werden von einem Server zentral ausgeliefert und von einem Client aus oft nur noch betrachtet. Für die im Rahmen dieser Arbeit entwickelten Techniken ergibt sich mit Blick auf solche Dienste ein breites Anwendungsfeld. Da die Techniken mehrheitlich auf segmentierten Daten beruhen und als polygonale Oberflächen dargestellt werden, ist eine einfache Übertragung in viele der aktuell verfügbaren 3D-Datenstrukturen zur Nutzung über einen Webbrowser möglich. Solche Formate sind z.B. UNIVERSAL-3D (ECMA INTERNATIONAL [2007]), VRML (WEB3D CONSORTIUM [2003]) oder O3D (GOOGLE [2010]). Techniken wie die automatische Sichtpunktbestimmung oder die Annotation von Szenen ist nicht auf ein bestimmtes Format oder eine spezielle Anwendungsumgebung festgelegt und kann praktisch in jedem Client

⁵¹Picture Archiving and Communication Systeme

zur Betrachtung von 3D-Darstellungen im Internet integriert werden. Ein weiterer Vorteil der Nutzung von polygonalen Netzen besonders mit Blick auf die oft vorhandenen technischen Grenzen der eingesetzten Clients im klinischen Umfeld sind die geringen Anforderungen an deren grafische Leistung. Im Gegensatz zum direkten Volumerendering stellen polygonale Darstellungen und darin eingesetzte Techniken keine großen Anforderungen an die genutzten Clients.

Die Verfügbarkeit einer Segmentierung als Voraussetzung der vorgestellten Techniken sowie deren Anwendung im Internet ist jedoch auch ein Nachteil. Der Prozess der Segmentierung ist, auch wenn er für quantitative Analysen unerlässlich ist, immer noch mit einem großen Aufwand verbunden. Dies stellt eine Hürde zu einer breiteren Anwendung von Techniken wie der automatischen Sichtpunktbestimmung dar. Dennoch können z.B. bei der Sichtpunktbestimmung einzelne Aspekte wie die Vorzugsblickrichtung des Chirurgen extrahiert werden, um in anderen Systemen auch ohne vorherige Segmentierung eingesetzt zu werden. Als Beispiel sei hier das LIVESYNC-System von KOHLMANN u. a. [2008] zu nennen, welche genau dies für Anwendungen mit direktem Volumerendering getan haben.

11.1.2 Evaluierung von Animationen

Die Evaluierung von Animationen wurde bisher kaum in der Forschung thematisiert (z.B. LIND & KJELLIN [2005], TUNUGUNTLA u. a. [2008] und DANCY & BEICHNER [2006]). Dies mag zum einen an der wesentlich jüngeren Geschichte von computergenerierten Animationen im Vergleich zu statischen Bildern liegen. Zum anderen spielen neben den verwendeten Visualisierungstechniken, wie sie auch bei statischen Bildern zum Einsatz kommen, eine ganze Reihe weiterer Parameter bei der Bewertung von Animationen eine Rolle. Das sind z.B. die Geschwindigkeit einer Animation, die Kameraführung, die Kohärenz der nacheinander dargestellten Informationen und der Grad der Interaktion. Aktuelle Arbeiten zur Evaluierung einzelner Visualisierungstechniken (BAER u. a. [2009]) zeigen zudem, dass selbst bei grundlegenden Visualisierungstechniken ein erheblicher Nachholbedarf besteht, was die Untersuchung der Effektivität in der Vermittlung räumlicher Verhältnisse und die Erkennbarkeit anatomischer Strukturen betrifft. Solange hier keine ausreichenden Grundlagen geschaffen wurden, wird es schwer sein, Animationen, die auf solchen Visualisierungstechniken aufbauen, objektiv zu evaluieren. Im Folgenden sollen dennoch einige Aspekte einer möglichen Evaluierung von Animationen diskutiert werden.

Generell kann davon ausgegangen werden, dass Animationen stets im Kontext zu anderen Techniken evaluiert werden sollten, z.B. im Vergleich mit statischen Bildern oder vollständig interaktiven Darstellungen. BAER u. a. [2009] haben die Erkennung von kritischen anatomischen Strukturen mit verschiedenen Visualisierungstechniken untersucht. Ein ähnlicher Ansatz ließe sich für Animationen realisieren: Es könnte evaluiert werden, wie viele kritische Strukturen einer bestimm-

ten Art, z.B. Lymphknoten, ein Betrachter in einer vordefinierten Animation, z.B. einer einfachen Rotation, in einer Serie statischer Bilder und durch individuelle Exploration findet. Die Ergebnisse in Relation zur benötigten Zeit und Genauigkeit könnten Rückschlüsse über den Erfolg oder Misserfolg von Animationen zulassen.

VAN WIJK & NUIJ [2004] haben die Teilnehmer einer Studie die Geschwindigkeit verschiedener Animationen individuell anpassen lassen, um so eine mittlere Animationsgeschwindigkeit zu ermitteln, die für bestimmte Anwendergruppen als Standardeinstellung sinnvoll war. Eine solche Methode könnte z.B. genutzt werden, um zunächst auf einer möglichst breiten Nutzerebene verschiedene Geschwindigkeiten z.B. für Animationen im Tumorboard oder für Patientenaufklärungen zu ermitteln.

Ob bestimmte hervorgehobene Strukturen in einer Animation überhaupt (ausreichend) wahrgenommen werden, kann mittels der Techniken des Eye-Tracking untersucht werden. Erste eigene Untersuchungen zeigen, dass es durchaus möglich ist, die Augenbewegungen eines Betrachters einer Animation verschiedenen Strukturen einer Animation zuzuordnen und so die fokussierten Strukturen über die Zeit extrahieren zu können.

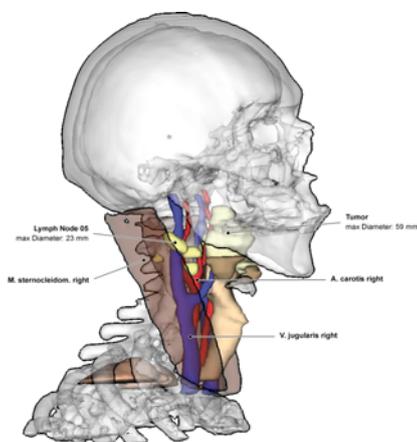
Genauso, wie es bei der Bewertung von Benutzerschnittstellen neben der Messung von Mauswegen auch den „Joy-Of-Use“ gibt, so gibt es bei Animationen neben der möglichen Verbesserung des Lerneffektes und der Unterstützung der räumlichen Wahrnehmung einen „Joy-Of-View“. Dieser kann bei gut durchdachten Animationen dazu beitragen, den „Joy-Of-Use“ einer Applikation als Ganzes zu erhöhen. Auf der anderen Seite kann durch zu lange und nicht angepasste Animationen der umgekehrte Effekt eintreten.

Es hat sich gezeigt, dass Animationen als ein Teil des komplexen Explorations- und Planungsprozesses nützlich sein können. Jedoch sollten sie in ihrer Art nicht überbewertet und stets im Kontext anderer unterstützender Maßnahmen gesehen und mit diesen (z.B. der automatischen Sichtpunktbestimmung) sehr sorgfältig verzahnt und abgestimmt werden. Für den Erfolg einer Applikation zur Unterstützung des chirurgischen Planungsprozessen bedarf es eines wohl dosierten Zusammenspiels mehrerer Komponenten: der Benutzerschnittstelle, der Performanz und Stabilität der Applikation, dem Erkennen der konkreten Nutzerbedürfnisse, der Nachvollziehbarkeit und Vorhersehbarkeit der Handlungen und Techniken wie der Animation.

A

Evaluierung der Annotationsdarstellungen

Im Folgenden sind die einzelnen Bilder, die im Erkundungsexperiment zur Evaluierung der neuen Annotationstechniken (Abschnitt 7.6) verwendet wurden, abgebildet. Zusätzlich finden sich zu jedem Bild einige der abgegebenen Kommentare sowie die Bewertungen der Darstellung und der Zuordnung (Skala von 1 (*sehr gut*) bis 5 (*sehr schlecht*)).



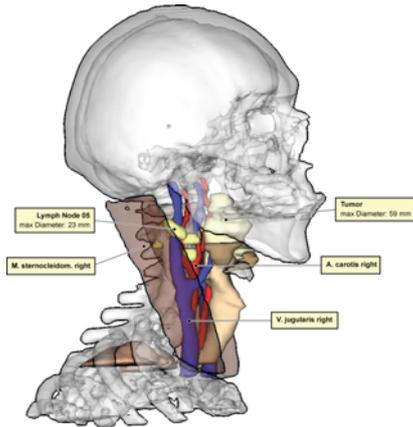
Bewertung der ...

... **Darstellung:** 2,6 ($\sigma = 0,92$)

... **Zuordnung:** 2,6 ($\sigma = 0,92$)

Kommentare:

- „*wirkt eleganter*“
- „*Zuordnungsstriche auf Grund der Länge teils schwer zu verfolgen*“

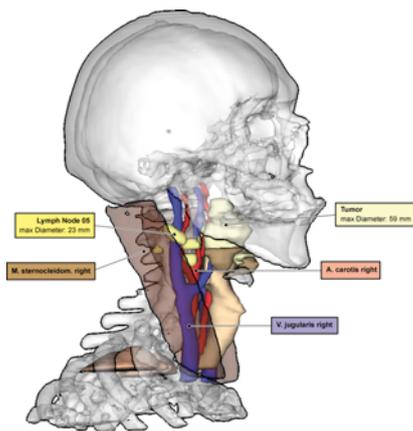


Bewertung der ...

... **Darstellung:** 2,1 ($\sigma = 0,89$)
 ... **Zuordnung:** 2,4 ($\sigma = 0,92$)

Kommentare:

- „Die Boxen sind vorteilhaft.“
- „bisl klobig, Lesbarkeit sicher überall gut. Heben sich aber nicht so gut von der Szene ab.“
- „Label zu erdrückend.“
- „praktischer, da die Informationseinheiten sichtbar gruppiert und abgegrenzt sind“

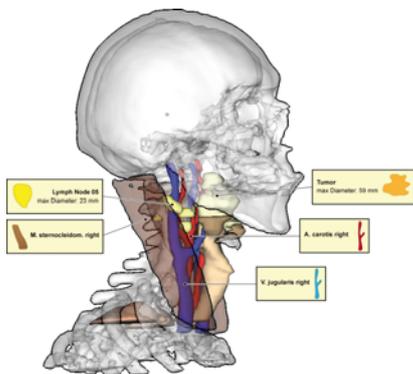


Bewertung der ...

... **Darstellung:** 2,1 ($\sigma = 1,21$)
 ... **Zuordnung:** 1,9 ($\sigma = 0,87$)

Kommentare:

- „Optisch sind die Farben kein Vorteil; für die Zuordnung aber klar.“
- „Schriftgröße der Label ist etwas klein“
- „Bisl zu bunt, dadurch heben die labels sich nicht gut von der scene ab, allerdings ist die zuordnung zu den strukturen etwas leichter.“
- „farbliche Korrelation sehr gut, da man auch gleich eine Strukturart-Zuordnung hat und es geht schneller da zwei Hinweise (Farbe + Pfeil) gegeben werden“

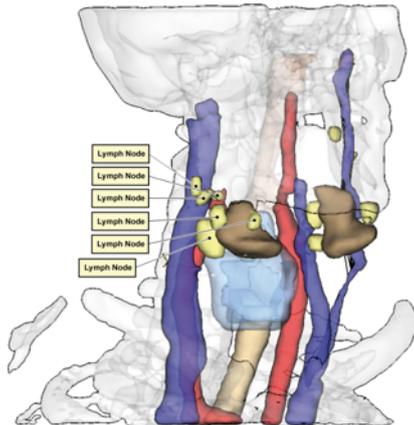


Bewertung der ...

... **Darstellung:** 2,4 ($\sigma = 1,15$)
 ... **Zuordnung:** 1,9 ($\sigma = 0,97$)

Kommentare:

- „Sehr gute Zuordnung, nette Idee, nur die Bilder machen die Labels etwas klobig“
- „Icons sind eine nette Idee.“
- „Gefällt mir sehr gut!“
- „Wirkt etwas verspielt mit den Silouetten daneben.“

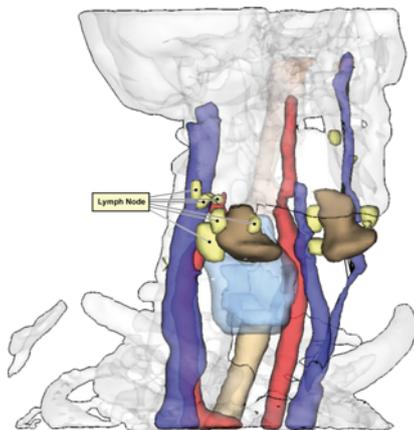
**Bewertung der ...**

... **Darstellung:** 3,6 ($\sigma = 0,83$)

... **Zuordnung:** 2,9 ($\sigma = 1,02$)

Kommentare:

- „Wirkt erstmal sehr chaotisch und erschlägt einen.“
- „Man erkennt gleich die Anzahl der Lymphknoten, aber die Zuordnung ist bei der Strichendenkomplexität eher schwer.“

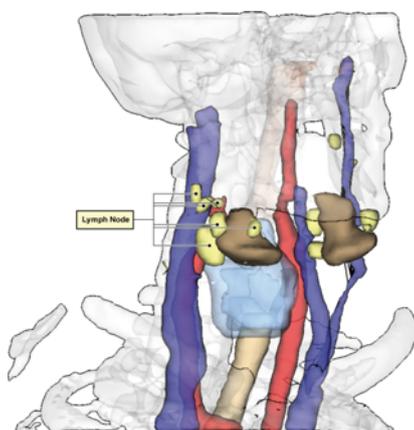
**Bewertung der ...**

... **Darstellung:** 2,1 ($\sigma = 0,76$)

... **Zuordnung:** 2,1 ($\sigma = 0,63$)

Kommentare:

- „Nicht unnötig komplex.“
- „Denke es muss nicht auf alle Lymphknoten gezeigt werden. Eventuell auf 1,2 zeigen und einen Hinweis(Icon, angedeutete Linien) das es mehrere gibt“
- „übersichtlicher und leichter nachvollziehbar“

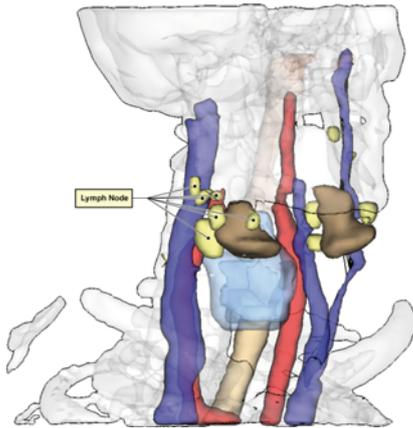
**Bewertung der ...**

... **Darstellung:** 3,1 ($\sigma = 0,91$)

... **Zuordnung:** 2,9 ($\sigma = 1,13$)

Kommentare:

- „Zuordnung sehr schwer durch den zu verfolgenden Linienknick.“
- „Zuordnung könnte sich bei der Beschriftung weiterer Strukturen überschneiden->da keine direkte Verbindung zwischen Struktur und Beschriftung->Verwirrung“
- „definitiv nicht intuitiv, visuell schwierig zu erfassen und nachzuvollziehen“

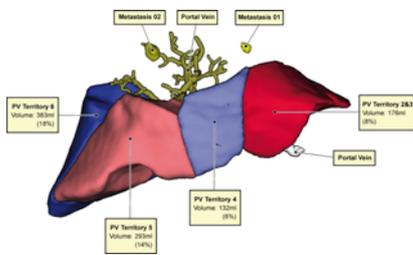


Bewertung der ...

- ... Darstellung:** 2,4 ($\sigma = 0,93$)
- ... Zuordnung:** 2,6 ($\sigma = 0,99$)

Kommentare:

- „Denke, dass auch hier Überschneidungen zu Verwirrung führen“
- „Von der Programmstruktur her bestimmt sinnvoller, aber nicht besonders schön“. Dennoch präferiert.“

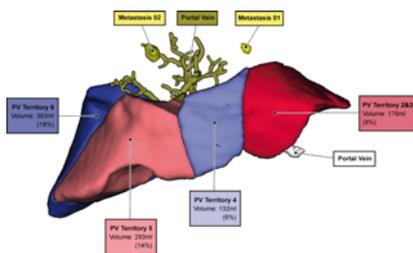


Bewertung der ...

- ... Darstellung:** 2,6 ($\sigma = 0,91$)
- ... Zuordnung:** 2,5 ($\sigma = 0,88$)

Kommentare:

- „Sieht eindeutig klarer aus. Schwierigkeiten bereitet mir teilweise die Darstellung der Linien - vor allem bei den Gefaessen. Bei den grosseren Strukturen ist dies einfacher“

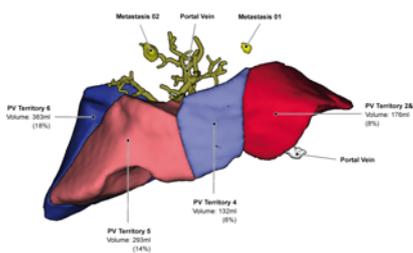


Bewertung der ...

- ... Darstellung:** 2,4 ($\sigma = 1,10$)
- ... Zuordnung:** 1,8 ($\sigma = 0,88$)

Kommentare:

- „wirkt zu bunt“
- „Trotz der Farben finde ich die Zuordnung nicht besser, da sich durch die gleichen Farben die Labels aktiv in die Szenen drängen.“
- „Schrift hier etwas zu klein.“



Bewertung der ...

- ... Darstellung:** 2,7 ($\sigma = 0,87$)
- ... Zuordnung:** 2,4 ($\sigma = 0,94$)

Kommentare:

- „An sich präferiere ich Labels ohne Box.“
- „eleganter“

Literaturverzeichnis

- [ADLER u. a. 2006] ADLER, M. ; HEGE, I. ; HOLZER, M. ; FISCHER, M.: CASUS - new implementations and challenges. In: *Slice of Life Conference*, 2006 (Zitiert auf Seite 31)
- [ADOBE SYSTEMS 2009] ADOBE SYSTEMS: *Adobe Photoshop CS4*. 2009 (Zitiert auf Seite 168)
- [ALI u. a. 2005] ALI, K. ; HARTMANN, K. ; STROTHOTTE, T.: Label Layout for Interactive 3D Illustrations. In: *Journal of the Winter School of Computer Graphics*, 13(1), 2005, S. 1–8 (Zitiert auf den Seiten 128, 129, 130, 132, 141 und 142)
- [ALPER u. a. 2001] ALPER, C. M. ; MYERS, E. N. ; EIBLING, D. E.: *Decision Making in Ear, Nose, and Throat Disorders*. W.B. Saunders Company, 2001 (Zitiert auf Seite 27)
- [APPLE 2009a] APPLE: *Apple Human Interface Guidelines*. 2009. – <http://developer.apple.com/mac/library/DOCUMENTATION/UserExperience/Conceptual/AppleHIGuidelines/XHIGIntro/XHIGIntro.html>. – besucht am 23.11.2009 (Zitiert auf Seite 230)
- [APPLE 2009b] APPLE: *Introduction to QuickTime VR*. 2009. – http://developer.apple.com/legacy/mac/library/documentation/QuickTime/InsideQT_QTVR/index.html. – besucht am 23.11.2009 (Zitiert auf Seite 40)
- [ARGELAGUET u. a. 2008] ARGELAGUET, F. ; ANDUJAR, C. ; TRUEBA, R.: Overcoming eye-hand visibility mismatch in 3D pointing selection. In: *Proc. of ACM symposium on Virtual reality software and technology (VRST)*, 2008, S. 43–46 (Zitiert auf Seite 200)
- [ASSMANN & HOEHNE 1983] ASSMANN, K. ; HOEHNE, K. H.: Investigation of Structures and Operations for Medical Image Data bases. In: *2nd Conference on Picture Archiving and Communication Systems*, 1983, S. 282–286 (Zitiert auf Seite 246)
- [AUTODESK 2009] AUTODESK: *Maya*. 2009. – <http://www.autodesk.com/maya>. – besucht am 1.3.2009 (Zitiert auf Seite 158)

LITERATURVERZEICHNIS

- [BADE u. a. 2005] BADE, R. ; RITTER, F. ; PREIM, B.: Usability Comparison of Mouse-Based Interaction Techniques for Predictable 3d Rotation. In: *Proc. of Smart Graphics: 5th International Symposium (SG)*, 2005, S. 138–150 (Zitiert auf den Seiten 51 und 55)
- [BADE u. a. 2008] BADE, R. ; CORDES, J. ; MEWES, M. ; PREIM, B.: Interaction Techniques for Case Selection in Medical Computer Based Training Systems. In: *Proc. of Mensch und Computer*, 2008, S. 247–256 (Zitiert auf Seite 218)
- [BADE 2008] BADE, R.: *Interaktive und dynamische Visualisierung fuer die chirurgische Ausbildung und Interventionsplanung*, Universität Magdeburg, Dissertation, 2008 (Zitiert auf Seite 19)
- [BAER u. a. 2009] BAER, A. ; ADLER, F. ; LENZ, D. ; PREIM, B.: Perception-based Evaluation of Emphasis Techniques Used in 3D Medical Visualization. In: *Proc. of Vision, Modeling, and Visualization Workshop (VMV)*, 2009, S. 295–304 (Zitiert auf den Seiten 87 und 247)
- [BARRAL u. a. 2000] BARRAL, P. ; DORME, G. ; PLEMENOS, D.: Scene understanding techniques using a virtual camera. In: *Proc. of Eurographics (EG)*, 2000. – short presentations (Zitiert auf Seite 108)
- [BARRAL u. a. 1999] BARRAL, P. ; DORME, G. ; PLEMENOS, D.: Visual Understanding of a Scene by Automatic Movement of a Camera. In: *Proc. of GraphiCon*, 1999 (Zitiert auf Seite 108)
- [BAVOIL u. a. 2005] BAVOIL, L. ; CALLAHAN, S. P. ; CROSSNO, P. J. ; FREIRE, J. ; SCHEIDEGGER, C. E. ; SILVA, C. T. ; VO, H. T.: VisTrails: enabling interactive multiple-view visualizations. In: *Proc. of IEEE Visualization (VIS)*, 2005, S. 135–142 (Zitiert auf Seite 185)
- [BECKHAUS u. a. 2001] BECKHAUS, S. ; RITTER, F. ; STROTHOTTE, T.: Guided Exploration with Dynamic Potential Fields: The CubicalPath System. In: *Computer Graphics Forum*, 20(4), 2001, S. 201–210 (Zitiert auf Seite 109)
- [BEKOS u. a. 2006] BEKOS, M. A. ; KAUFMANN, M. ; SYMVONIS, A. ; WOLFF, A.: Boundary Labeling: Models and Efficient Algorithms for Rectangular Maps. In: *Computational Geometry: Theory and Applications*, 36(3), 2006, S. 215–236 (Zitiert auf Seite 125)
- [BELL u. a. 2001] BELL, B. ; FEINER, S. ; HÖLLERER, T.: View management for virtual and augmented reality. In: *Proc. of ACM Symposium on User Interface Software and Technology (UIST)*, 2001, S. 101–110 (Zitiert auf Seite 126)
- [BIRTH u. a. 2003] BIRTH, M. ; HILDEBRAND, P. ; KLEEMANN, M. ; BRUCH, H.: Interventionelle Therapie von Lebermetastasen. In: *Chirurgische Gastroenterologie*, 19, 2003, S. 349–358 (Zitiert auf Seite 13)
- [BITTER u. a. 2007] BITTER, I. ; UITERT, R. V. ; WOLF, I. ; IBANEZ, L. ; KUHNIGK, J.-M.: Comparison of Four Freely Available Frameworks for Image Processing and Visualization That Use ITK. In: *IEEE Transactions on Visualization and*

- Computer Graphics (TVCG)*, 13(3), 2007, S. 483–493 (Zitiert auf den Seiten [190](#) und [207](#))
- [BITTNER & WONKA 2003] BITTNER, J. ; WONKA, P.: Visibility in Computer Graphics. In: *Environment and Planning B: Planning and Design*, 30(5), 2003, S. 729–756 (Zitiert auf Seite [61](#))
- [BLAKE u. a. 2008] BLAKE, K. ; RASHID, M. ; CURLEY, J. ; MORLEY, S. ; HOLMES, B.: Bridging the gap: using student-generated clinical case presentations in paediatrics. In: *The Clinical Teacher*, 5(4), 2008, S. 208–212 (Zitiert auf Seite [33](#))
- [BLANZ u. a. 1999] BLANZ, V. ; TARR, M. ; BÜLTHOFF, H.: What object attributes determine canonical views? In: *Perception*, 28, 1999, S. 575–599 (Zitiert auf Seite [59](#))
- [BORDOLOI & SHEN 2005] BORDOLOI, U. ; SHEN, H.-W.: View Selection for Volume Rendering. In: *Proc. of IEEE Visualization (VIS)*, 2005, S. 487–494 (Zitiert auf den Seiten [60](#) und [91](#))
- [BOTH A & POST 2008] BOTH A, C. P. ; POST, F. H.: Hybrid Scheduling in the DeVIDE Dataflow Visualisation Environment. In: *Proc. of Simulation and Visualization (SimVis)*, 2008, S. 309–322 (Zitiert auf den Seiten [187](#), [188](#) und [207](#))
- [BOUCHEIX & SCHNEIDER 2009] BOUCHEIX, J.-M. ; SCHNEIDER, E.: Static and animated presentations in learning dynamic mechanical systems. In: *Learning and Instruction*, 19(2), 2009, S. 112–127 (Zitiert auf Seite [3](#))
- [BOURQUAIN u. a. 2002] BOURQUAIN, H. ; SCHENK, A. ; LINK, F. ; PREIM, B. ; PRAUSE, G. ; PEITGEN, H.-O.: HepaVision2: A software assistant for preoperative planning in living-related liver transplantation and oncologic liver surgery. In: *Proc. of Computer Assisted Radiology and Surgery (CARS)*, 2002, S. 341–346 (Zitiert auf den Seiten [19](#), [23](#), [181](#) und [207](#))
- [BROELSCH u. a. 1990] BROELSCH, C. ; EMOND, J. ; WHITINGTON, P. ; THISTLETHWAITE, J. ; BAKER, A. ; LICHTOR, J.: Application of reduced-size liver transplants as split grafts, auxiliary orthotopic grafts, and living related segmental transplants. In: *Annals of Surgery*, 212(3), 1990, S. 368–377 (Zitiert auf Seite [11](#))
- [BRUCKNER & GRÖLLER 2005] BRUCKNER, S. ; GRÖLLER, M. E.: VolumeShop: An Interactive System for Direct Volume Illustration. In: *Proc. of IEEE Visualization (VIS)*, 2005, S. 671–678 (Zitiert auf den Seiten [181](#), [185](#), [186](#) und [190](#))
- [BRUCKNER 2008] BRUCKNER, S.: *Interactive Illustrative Volume Visualization*, Vienna University of Technology, Dissertation, 2008 (Zitiert auf Seite [1](#))
- [BUCKLEY 1994] BUCKLEY, C.: Bézier Curves for Camera Motion. In: *Proc. of 2nd Irish Computer Graphics Workshop*, 1994 (Zitiert auf Seite [111](#))
- [BUMM u. a. 2002] BUMM, R. ; SIESS, M. ; LANGE, M. ; SIEWERT, J.: Notwendige Voraussetzungen für die Funktion eines onkologischen Kompetenzzentrums – Infomationstechnologie, Befunddokumentation und Telekommunikation. In:

LITERATURVERZEICHNIS

- Deutsche Medizinische Wochenschrift*, 127(17), 2002, S. 907–912 (Zitiert auf Seite 29)
- [BÜRGER u. a. 2008] BÜRGER, K. ; KRÜGER, J. ; WESTERMANN, R.: Direct Volume Editing. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 14(6), 2008, S. 1388–1395 (Zitiert auf Seite 127)
- [CABAN u. a. 2007] CABAN, J. J. ; JOSHI, A. ; NAGY, P.: Rapid Development of Medical Imaging Tools with Open-Source Libraries. In: *Journal of Digital Imaging*, 20, 2007, S. 83–93 (Zitiert auf Seite 181)
- [CAI u. a. 2001] CAI, W. ; FENG, D. ; FULTON, R.: Web-Based Digital Medical Images. In: *IEEE Computer Graphics and Applications (CGA)*, 21(1), 2001, S. 44–47 (Zitiert auf Seite 131)
- [CHAN u. a. 2006] CHAN, M.-Y. ; QU, H. ; WU, Y. ; ZHOU, H.: Viewpoint Selection for Angiographic Volume. In: *Proc. of Advances in Visual Computing, Second International Symposium (ISVC)*, 2006, S. 528–537 (Zitiert auf Seite 61)
- [CHAN u. a. 2008] CHAN, M.-Y. ; QU, H. ; CHUNG, K.-K. ; MAK, W.-H. ; WU, Y.: Relation-Aware Volume Exploration Pipeline. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 14(6), 2008, S. 1683–1690 (Zitiert auf Seite 60)
- [CHARITON & CHOI 2002] CHARITON, C. ; CHOI, M.-H.: User interface guidelines for enhancing usability of airline travel agency e-commerce web sites. In: *Proc. of Conference on Human Factors in Computing Systems (CHI)*, 2002, S. 676–677 (Zitiert auf Seite 230)
- [CHAUDHURI u. a. 2004] CHAUDHURI, P. ; KHANDEKAR, R. ; SETHI, D. ; KALRA, P.: An Efficient Central Path Algorithm for Virtual Navigation. In: *Proc. of Computer Graphics International Conference*, 2004, S. 188–195 (Zitiert auf Seite 109)
- [CHRISTENSEN u. a. 1995] CHRISTENSEN, J. ; MARKS, J. ; SHIEBER, S.: An empirical study of algorithms for point-feature label placement. In: *ACM Transactions on Graphics*, 14, 1995, S. 203–232 (Zitiert auf Seite 125)
- [CHRISTIE u. a. 2005] CHRISTIE, M. ; MACHAP, R. ; NORMAND, J.-M. ; OLIVIER, P. ; PICKERING, J.: Virtual Camera Planning: A Survey. In: *Proc. of Smart Graphics (SG)*, 2005, S. 40–52 (Zitiert auf Seite 64)
- [CHRISTIE u. a. 2008] CHRISTIE, M. ; OLIVIER, P. ; NORMAND, J.-M.: Camera Control in Computer Graphics. In: *Computer Graphics Forum (CGF)*, 27(8), 2008, S. 2197–2218 (Zitiert auf den Seiten 55 und 108)
- [CHRONAKI u. a. 1997] CHRONAKI, C. ; ZABULIS, X. ; ORPHANOUDAKIS, S.: I2Cnet Medical Image Annotation Service. In: *Medical Informatics*, 22(4), 1997, S. 337–347 (Zitiert auf Seite 131)

- [CHU & CHUNG 2002] CHU, G. ; CHUNG, M.: Autonomous Selection and Modification of Camera Configurations Using Visibility and Manipulability Measures. In: *Journal of Robotic Systems*, 19(4), 2002, S. 219–230 (Zitiert auf Seite [62](#))
- [CLEYNENBREUGEL u. a. 1996] CLEYNENBREUGEL, J. V. ; BELLON, E. ; MARCHAL, G. ; SUETENS, P.: Annotating radiological images for computer assisted communication and teaching. In: *Computer Communications*, 19(6–7), 1996, S. 498–508 (Zitiert auf Seite [131](#))
- [COHEN 1992] COHEN, J.: A power primer. In: *Psychological Bulletin*, 112(1), 1992, S. 155–159 (Zitiert auf Seite [87](#))
- [COHEN u. a. 1988] COHEN, M. F. ; CHEN, S. E. ; WALLACE, J. R. ; GREENBERG, D. P.: A progressive refinement approach to fast radiosity image generation. In: *ACM SIGGRAPH Computer Graphics*, 22(4), 1988, S. 75–84 (Zitiert auf Seite [62](#))
- [COHEN-OR u. a. 2003] COHEN-OR, D. ; CHRYSANTHOU, Y. L. ; SILVA, C. T. ; DURAND, F.: A Survey of Visibility for Walkthrough Applications. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 9(3), 2003, S. 412–431 (Zitiert auf den Seiten [62](#), [63](#) und [108](#))
- [CORDES u. a. 2006a] CORDES, J. ; MÜHLER, K. ; PREIM, B.: Die Konzeption des LiverSurgeryTrainers. In: *Proc. of Workshop Softwareassistenten*, 2006 (Zitiert auf Seite [212](#))
- [CORDES u. a. 2006b] CORDES, J. ; DORNHEIM, J. ; PREIM, B. ; HERTEL, I. ; STRAUSS, G.: Preoperative Segmentation of Neck CT Datasets for the Planning of Neck Dissections. In: *Proc. of SPIE Medical Imaging*, 2006, S. 1447–1456 (Zitiert auf den Seiten [18](#), [24](#), [181](#) und [207](#))
- [CORDES u. a. 2007a] CORDES, J. ; MÜHLER, K. ; OLDHAFFER, K. ; STAVROU, G. ; HILLERT, C. ; PREIM, B.: Evaluation of a training system of the computer-based planning of liver surgery. In: *Proc. of Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, 2007, S. 151–154 (Zitiert auf den Seiten [201](#), [202](#), [212](#) und [229](#))
- [CORDES u. a. 2007b] CORDES, J. ; MÜHLER, K. ; OLDHAFFER, K. J. ; STAVROU, G. ; HILLERT, C. ; PREIM, B.: Szenariobasierte Entwicklung eines chirurgischen Trainingssystem. In: *Proc. of eLearning in der Medizin und Zahnmedizin (11. Workshop der GMDS AG)*, 2007, S. 17–30 (Zitiert auf den Seiten [212](#) und [215](#))
- [CORDES u. a. 2008] CORDES, J. ; HINTZ, K. ; FRANKE, J. ; BOCHWITZ, C. ; PREIM, B.: Conceptual Design and Prototyping Implementation of a Case-based Training System for Spine Surgery. In: *Proc. of 1st International eLearning Baltics Science Conference (eLBa)*, 2008, S. 169–178 (Zitiert auf den Seiten [205](#) und [229](#))
- [COUINAUD 1954] COUINAUD, C.: Lobes et segments hepatiques: notes sur l'architecture anatomique et chirurgicale du foie. In: *Presse Med.*, 62, 1954, S. 709–712 (Zitiert auf Seite [10](#))

LITERATURVERZEICHNIS

- [COVER & THOMAS 1991] COVER, T. M. ; THOMAS, J. A.: *Elements of Information Theory*. Wiley-Interscience, 1991 (Zitiert auf Seite 57)
- [CROSSINGHAM u. a. 2009] CROSSINGHAM, J. ; JENKINSON, J. ; WOOLRIDGE, N. ; GALLINGER, S. ; TAIT, G. ; MOULTON, C.-A.: Interpreting three-dimensional structures from two-dimensional images: A web-based interactive 3D teaching model of surgical liver anatomy. In: *Official Journal of The International Hepato-Pancreato-Biliary Association*, 11(6), 2009, S. 523–528 (Zitiert auf den Seiten 33 und 34)
- [DANCY & BEICHNER 2006] DANCY, M. H. ; BEICHNER, R. J.: Impact of animation on assessment of conceptual understanding in physics. In: *Physical Review Special Topics – Physics Education Research*, 2(1), 2006, S. 010104 (7 Seiten) (Zitiert auf Seite 247)
- [DEINZER u. a. 2003] DEINZER, F. ; DENZLER, J. ; NIEMANN, H.: Viewpoint Selection - Planning Optimal Sequences of Views for Object Recognition. In: *Proc. of International Conference on Computer Vision*, 2003, S. 65–73 (Zitiert auf Seite 63)
- [DELINGETTE & AYACHE 2005] DELINGETTE, H. ; AYACHE, N.: Hepatic surgery simulation. In: *Communications of the ACM*, 48(2), 2005, S. 31–36 (Zitiert auf den Seiten 31 und 212)
- [DERICHS u. a. 2005] DERICHS, C. ; DEINZER, F. ; NIEMANN, H.: Cost Integration in Multi-step Viewpoint Selection for Object Recognition. In: *Proc. of Machine Learning and Data Mining in Pattern Recognition*, 2005, S. 415–425 (Zitiert auf Seite 63)
- [DIRISAMER u. a. 2005] DIRISAMER, A. ; FRIEDRICH, K. ; SCHIMA, W.: Anatomie und Normvarianten der Segmente, Gefäße und Gallengänge der Leber. In: *Der Radiologe*, 45(1), 2005, S. 8–14 (Zitiert auf Seite 10)
- [DORNHEIM u. a. 2008] DORNHEIM, J. ; DORNHEIM, L. ; PREIM, B. ; STRAUSS, G.: Modellbasierte Segmentierung von Weichgewebestrukturen in CT-Datensätzen des Halses. In: *Proc. of Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, 2008, S. 197–200 (Zitiert auf Seite 18)
- [DUDENREDAKTION 2009] DUDENREDAKTION: *Duden 01. Die deutsche Rechtschreibung. Band 1*. Bibliographisches Institut & F. A. Brockhaus, 2009 (Zitiert auf Seite 181)
- [DURAND u. a. 1997] DURAND, F. ; DRETTAKIS, G. ; PUECH, C.: The Visibility Skeleton: a Powerful and Efficient Multi-Purpose Global Visibility Tool. In: *Computer Graphics (CG)*, 31, 1997, S. 89–100 (Zitiert auf den Seiten 62 und 63)
- [ECMA INTERNATIONAL 2007] ECMA INTERNATIONAL: *Standard ECMA - 363: Universal 3D File Format*. 4, 2007 (Zitiert auf Seite 246)

- [ELLIOTT u. a. 1996] ELLIOTT, P. J. ; DIEDRICHSEN, J. ; GOODSON, K. ; RISTE-SMITH, R. ; SIVEWRIGHT, G.: An object-oriented system for 3D medical image analysis. In: *IBM Systems Journal*, 35(1), 1996, S. 4–24 (Zitiert auf Seite 230)
- [ELMQVIST & FEKETE 2008] ELMQVIST, N. ; FEKETE, J.-D.: Semantic pointing for object picking in complex 3D environments. In: *Proc. of Graphics Interface (GI)*, 2008, S. 243–250 (Zitiert auf Seite 200)
- [ELMQVIST & TSIGAS 2008] ELMQVIST, N. ; TSIGAS, P.: A Taxonomy of 3D Occlusion Management for Visualization. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 14, 2008, S. 1095–1109 (Zitiert auf Seite 55)
- [ENQUOBAHRIE u. a. 2007] ENQUOBAHRIE, A. ; CHENG, P. ; GARY, K. ; IBANEZ, L. ; GOBBI, D. ; LINDSETH, F. ; YANIV, Z. ; AYLWARD, S. ; JOMIER, J. ; CLEARY, K.: The Image-Guided Surgery Toolkit IGSTK: An Open Source C++ Software Toolkit. In: *Journal of Digital Imaging*, 20, 2007, S. 21–33 (Zitiert auf Seite 187)
- [FISCHER u. a. 2005] FISCHER, L. ; THORN, M. ; NEUMANN, J. ; HEIMANN, T. ; GRENACHER, L. ; MEINZER, H. ; FRIESS, H. ; BÜCHLER, M.: Die computergestützte Analyse von Größe, Position und Form venöser Lebersegmenten und deren Lagebeziehung zu dem Couinaud und portalen Lebersegmenten. In: *Proc. of Bildverarbeitung für die Medizin (BVM)*, 2005, S. 21–25 (Zitiert auf Seite 17)
- [FISCHER u. a. 1996] FISCHER, M. ; SCHAUER, S. ; GRÄSEL, C. ; BAEHRING, T. ; H. MANDL, H. ; GÄRTNER, R. ; SCHERBAUM, W. ; SCRIBA, P.: Modellversuch CA-SUS: Ein computergestütztes Autorensystem für die problemorientierte Lehre in der Medizin. In: *Zeitschrift für ärztliche Fortbildung*, 90, 1996, S. 385–389 (Zitiert auf Seite 31)
- [FISCHER u. a. 2008] FISCHER, M. R. ; HEGE, I. ; HÖRNLEIN, A. ; PUPPE, F. ; TÖNSHOFF, B. ; HUWENDIEK, S. S.: Virtuelle Patienten in der medizinischen Ausbildung: Vergleich verschiedener Strategien zur curricularen Integration. In: *Zeitschrift für Evidenz, Fortbildung und Qualität im Gesundheitswesen*, 102(10), 2008, S. 648–653 (Zitiert auf Seite 33)
- [FRASER u. a. 2009] FRASER, J. F. ; SCHWARTZ, T. H. ; KAPLITT, M. G.: *Textbook of Stereotactic and Functional Neurosurgery*. Kap. BrainLab Image Guided System, S. 567–581, Springer Berlin Heidelberg, 2009 (Zitiert auf Seite 20)
- [FU u. a. 2008] FU, H. ; COHEN-OR, D. ; DROR, G. ; SHEFFER, A.: Upright orientation of man-made objects. In: *ACM Transactions on Graphics*, 27(3), 2008, S. 1–7 (Zitiert auf Seite 58)
- [GATCLIFFE & COLEMAN 2008] GATCLIFFE, T. ; COLEMAN, R.: Tumor board: More than treatment planning – A 1-year prospective survey. In: *Journal of Cancer Education*, 23(4), 2008, S. 235–237 (Zitiert auf Seite 29)
- [GELB & GARDINER 1997] GELB, J. ; GARDINER, J. J.: Developing a Company Style Guide. In: *Proc. of 44th International Technical Communication Conference*, 1997, S. 469–472 (Zitiert auf Seite 230)

LITERATURVERZEICHNIS

- [GERING u. a. 1999] GERING, D. ; NABAVI, A. ; KIKINIS, R. ; GRIMSON, W. E. L. ; HATA, N. ; EVERETT, P. ; JOLESZ, F. ; III., W. W.: An Integrated Visualization System for Surgical Planning and Guidance using Image Fusion and Interventional Imaging. In: *Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 1999, S. 809–819 (Zitiert auf den Seiten 181, 186 und 190)
- [GLASSER u. a. 2010] GLASSER, S. ; OELTZE, S. ; HENNEMUTH, A. ; KUBISCH, C. ; MAHNKEN, A. ; WILHELMSSEN, S. ; PREIM, B.: Automatic Transfer Function Specification for Visual Emphasis of Coronary Artery Plaque. In: *Computer Graphics Forum (CGF)*, erscheint, 2010 (Zitiert auf den Seiten 150 und 151)
- [GLITTENBERG & BINDER 2008] GLITTENBERG, C. ; BINDER, S.: Results of the ICAPE pilot study: Interactive computer aided patient education pilot study. In: *Spektrum der Augenheilkunde*, 22(6), 2008, S. 376–383 (Zitiert auf Seite 38)
- [GÜNTHER u. a. 2008] GÜNTHER, P. ; GÖPPL, M. ; TRÖGER, J. ; HOLLAND-CUNZ, S. ; WAAG, K. L. ; SCHENK, J. P.: Operationsplanung solider, abdominaler Tumoren im Kindesalter. In: *Monatsschrift Kinderheilkunde*, 156(6), 2008, S. 569–573 (Zitiert auf Seite 17)
- [GOEDE u. a. 2004] GOEDE, P. A. ; LAUMAN, J. R. ; COHELLA, C. ; KATZMAN, G. L. ; MORTON, D. A. ; ALBERTINE, K. H.: A Methodology and Implementation for Annotating Digital Images for Context-appropriate Use in an Academic Health Care Environment. In: *Journal of the American Medical Informatics Association*, 11(1), 2004, S. 29–41 (Zitiert auf Seite 131)
- [GOOGLE 2010] GOOGLE: *O3D API*. 2010. – <http://code.google.com/apis/o3d/>. – besucht am 02.02.2010 (Zitiert auf Seite 246)
- [GREVERA u. a. 2007] GREVERA, G. ; UDUPA, J. ; ODHNER, D. ; ZHUGE, Y. ; SOUZA, A. ; IWANAGA, T. ; MISHRA, S.: CAVASS: A Computer-Assisted Visualization and Analysis Software System. In: *Journal of Digital Imaging*, 20, 2007, S. 101–118 (Zitiert auf Seite 187)
- [GROTH & STREEFKERK 2006] GROTH, D. P. ; STREEFKERK, K.: Provenance and Annotation for Visual Exploration Systems. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 12(6), 2006, S. 1500–1510 (Zitiert auf den Seiten 158 und 168)
- [GRUENBERGER 2004] GRUENBERGER, T.: Anforderungen der Chirurgie an die bildgebende Diagnostik. In: *Der Radiologe*, 44(12), 2004, S. 1154–1159 (Zitiert auf Seite 27)
- [GÖTZELMANN u. a. 2006] GÖTZELMANN, T. ; HARTMANN, K. ; STROTHOTTE, T.: Contextual Grouping of Labels. In: *Proc. of Simulation and Visualization (SimVis)*, 2006, S. 245–258 (Zitiert auf Seite 127)
- [GÖTZELMANN u. a. 2005] GÖTZELMANN, T. ; ALI, K. ; HARTMANN, K. ; STROTHOTTE, T.: Form Follows Function: Aesthetic Interactive Labels. In: *Proc. of*

- Computational Aesthetics*, 2005, S. 193–200 (Zitiert auf den Seiten [119](#), [121](#), [127](#) und [128](#))
- [GÖTZELMANN u. a. 2007a] GÖTZELMANN, T. ; HARTMANN, K. ; STROTHOTTE, T.: Annotation of Animated 3D Objects. In: *Proc. of Simulation and Visualization (SimVis)*, 2007, S. 209–222 (Zitiert auf Seite [150](#))
- [GÖTZELMANN u. a. 2007b] GÖTZELMANN, T. ; VÁZQUEZ, P.-P. ; HARTMANN, K. ; NÜRNBERGER, A. ; STROTHOTTE, T.: Correlating Text and Images: Concept and Evaluation. In: *Proc. of Smart Graphics (SG)*, 2007, S. 97–109 (Zitiert auf Seite [64](#))
- [GULLIKSEN & SANDBLAD 1995] GULLIKSEN, J. ; SANDBLAD, B.: Domain-specific design of user interfaces. In: *International Journal of Human-Computer Interaction*, 7(2), 1995, S. 135–151 (Zitiert auf Seite [230](#))
- [HAAG 1998] HAAG, M.: *Plattformunabhängige, adaptive Lehr-/Lernsysteme für die medizinische Aus- und Weiterbildung*, Universität Heidelberg, Dissertation, 1998 (Zitiert auf Seite [31](#))
- [HALPER u. a. 2001] HALPER, N. ; HELBING, R. ; STROTHOTTE, T.: A Camera Engine for Computer Games: Managing the Trade-Off Between Constraint Satisfaction and Frame Coherence. In: *Proc. of Eurographics (EG)*, 2001, S. 174–183 (Zitiert auf den Seiten [62](#), [64](#) und [108](#))
- [HAMEL & STROTHOTTE 1999] HAMEL, J. ; STROTHOTTE, T.: Capturing and Re-Using Rendition Styles for Non-Photorealistic Rendering. In: *Proc. of Eurographics (EG)*, 1999, S. 173–182 (Zitiert auf Seite [157](#))
- [HANRAHAN u. a. 1991] HANRAHAN, P. ; SALZMAN, D. ; AUPPERLE, L.: A rapid hierarchical radiosity algorithm. In: *Proc. of ACM SIGGRAPH Computer Graphics*, 1991, S. 197–206 (Zitiert auf Seite [62](#))
- [HANSEN u. a. 2007a] HANSEN, C. ; SCHLICHTING, S. ; MARKERT, M. ; HINDENNACH, M. ; ZIDOWITZ, S. ; PEITGEN, H.-O.: Intraoperative Adaptation of Preoperative Risk Analyses for Oncologic Liver Surgery using Navigated 2D-Ultrasound. In: *Proc. of Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, 2007, S. 155–158 (Zitiert auf Seite [99](#))
- [HANSEN u. a. 2008] HANSEN, C. ; KOEHN, A. ; SCHLICHTING, S. ; FWEILER ; ZIDOWITZ, S. ; KLEEMANN, M. ; PEITGEN, H.-O.: Intraoperative Modification of Resection Plans for Liver Surgery. In: *International Journal of Computer Assisted Radiology and Surgery*, 3(3-4), 2008, S. 291–297 (Zitiert auf Seite [30](#))
- [HANSEN u. a. 2010] HANSEN, C. ; ZIDOWITZ, S. ; SCHENK, A. ; K.-J.OLDHAFFER ; LANG, H. ; PEITGEN, H.-O.: Risk Maps for Navigation in Liver Surgery. In: *Proc. of SPIE Medical Imaging* (erscheint), 2010 (Zitiert auf Seite [223](#))
- [HANSEN u. a. 2007b] HANSEN, C. ; KÖHN, A. ; RITTER, F. ; ZIDOWITZ, S. ; PEITGEN, H.-O.: Simultaneous Visualization of Preoperative Planning Models and

LITERATURVERZEICHNIS

- Intraoperative 2D Ultrasound for Liver Surgery. In: *Proc. of Eurographics (EG)*, 2007, S. 117–120. – short paper (Zitiert auf Seite [100](#))
- [HASTREITER u. a. 1998] HASTREITER, P. ; REZK-SALAMA, C. ; TOMANDL, B. ; EBERHARDT, K. ; ERTL, T.: Fast Analysis of Intracranial Aneurysms Based on Interactive Direct Volume Rendering and CTA. In: *Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 1998, S. 660–669 (Zitiert auf Seite [60](#))
- [HELMBERGER u. a. 2005] HELMBERGER, T. ; HOFFMANN, R.-T. ; JAKOBS, T. ; LEIBECKE, T. ; LUBIENSKI, A. ; REISER, M.: Tumorablation der Leber. In: *Der Radiologe*, 45, 2005, S. 55–62 (Zitiert auf Seite [13](#))
- [HERRMANN 2002] HERRMANN, M.: Dreidimensionale Computeranimation – neues Medium zur Unterstützung des Aufklärungsgesprächs vor Operationen. In: *Der Chirurg*, 73(5), 2002, S. 500–507 (Zitiert auf Seite [38](#))
- [HÖHNE u. a. 2003] HÖHNE, K. H. ; PFLESSER, B. ; POMMERT, A. ; PRIESMEYER, K. ; RIEMER, M. ; SCHIEMANN, T. ; SCHUBERT, R. ; TIEDE, U. ; FREDERKING, H. ; GEHRMANN, S. ; NOSTER, S. ; SCHUMACHER, U.: *VOXEL-MAN 3D Navigator: Inner Organs. Regional, Systemic and Radiological Anatomy - 3 CD-ROMs*. Springer-Verlag Electronic Media, 2003 (Zitiert auf Seite [31](#))
- [HÖHNE u. a. 2008] HÖHNE, K. H. ; GEHRMANN, S. ; NAZAR, T. ; PETERSIK, A. ; PFLESSER, B. ; POMMERT, A. ; SCHUMACHER, U. ; TIEDE, U.: *VOXEL-MAN 3D Navigator: Upper Limb*. Springer Electronic Media, 2008 (Zitiert auf den Seiten [31](#), [32](#), [34](#), [131](#) und [132](#))
- [IMHOF 1975] IMHOF, E.: Positioning Names on Maps. In: *The American Cartographer*, 2(2), 1975, S. 128–144 (Zitiert auf Seite [121](#))
- [INTERRANTE u. a. 1997] INTERRANTE, V. ; FUCHS, H. ; PIZER, S. M.: Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 3(2), 1997, S. 98–117 (Zitiert auf Seite [198](#))
- [ISERHARDT-BAUER u. a. 2002] ISERHARDT-BAUER, S. ; HASTREITER, P. ; TOMANDL, B. ; KÖSTNER, N. ; SCHEMPERSHOFE, M. ; NISSEN, U. ; ERTL, T.: Standardized Analysis of Intracranial Aneurysms Using Digital Video Sequences. In: *Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2002, S. 411–418 (Zitiert auf den Seiten [39](#), [109](#) und [158](#))
- [ISERHARDT-BAUER u. a. 2001] ISERHARDT-BAUER, S. ; HASTREITER, P. ; ERTL, T. ; EBERHARDT, K. ; TOMANDL, B.: Case Study: Medical Web Service for the Automatic 3D Documentation for Neuroradiological Diagnosis. In: *Proc. of IEEE Visualization (VIS)*, 2001, S. 425–428 (Zitiert auf Seite [39](#))

- [JAECK u. a. 2003] JAECK, D. ; BACHELLIER, P. ; WEBER, J. C. ; OUSSOULTZOGLOU, E. ; GREGET, M.: Process in surgical exeresis of colorectal cancer liver metastases. In: *Bulletin de l'Académie nationale de médecine*, 187(5), 2003, S. 863–876 (Zitiert auf Seite 29)
- [JANKE u. a. 2006] JANKE, C. ; TIETJEN, C. ; BAER, A. ; ZWICK, C. ; PREIM, B. ; HERTEL, I. ; STRAUSS, G.: Design und Realisierung eines Softwareassistenten zur Planung von Halsoperationen. In: *Proc. of Mensch und Computer*, 2006, S. 373–378 (Zitiert auf Seite 229)
- [JANKUN-KELLY u. a. 2007] JANKUN-KELLY, T. ; MA, K.-L. ; GERTZ, M.: A Model and Framework for Visualization Exploration. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 3(2), 2007, S. 357–369 (Zitiert auf Seite 156)
- [JANSEN u. a. 2001] JANSEN, T. ; RYMON-LIPINSKI, B. von ; KROL, Z. ; RITTER, L. ; KEEVE, E.: An Extendable Application Framework for Medical Visualization and Surgical Planning. In: *Proc. of SPIE Medical Imaging*, 2001, S. 349–357 (Zitiert auf Seite 186)
- [JHA u. a. 2002] JHA, V. ; WIDDOWSON, S. ; DUFFY, S.: Development and evaluation of an interactive computer-assisted learning program – a novel approach to teaching gynaecological surgery. In: *British Journal of Educational Technology*, 33(3), 2002, S. 323–331 (Zitiert auf Seite 33)
- [KAINMUELLER u. a. 2007] KAINMUELLER, D. ; LANGE, T. ; LAMECKER, H.: Shape Constrained Automatic Segmentation of the Liver based on a Heuristic Intensity Model. In: *MICCAI Workshop on 3D Segmentation in the Clinic: A Grand Challenge*, 2007, S. 109–116 (Zitiert auf den Seiten 19 und 27)
- [KAPPEL u. a. 2006] KAPPEL, K. ; TOMITSCH, M. ; KÖLTRINGER, T. ; GRECHENIG, T.: Developing user interface guidelines for DVD menus. In: *Proc. of Conference on Human Factors in Computing Systems (CHI)*, 2006, S. 177–182 (Zitiert auf Seite 230)
- [KHALIL u. a. 2005] KHALIL, M. K. ; PAAS, F. ; JOHNSON, T. E. ; PAYER, A. F.: Design of interactive and dynamic anatomical visualizations: The implication of cognitive load theory. In: *The Anatomical Record Part B: The New Anatomist*, 286B, 2005, S. 15–20 (Zitiert auf Seite 123)
- [KHAWAJA u. a. 1996] KHAWAJA, K. ; MACIEJEWSKI, A. ; TRETTER, D. ; BOUMAN, C.: Camera and light placement for automated assembly inspection. In: *Proc. of IEEE International Conference on Robotics and Automation*, 1996, S. 3246–3252 (Zitiert auf Seite 56)
- [KINDSMÜLLER u. a. 2009] KINDSMÜLLER, M. C. ; SCHULZ, H. ; HAAR, M.: Usability Engineering für medizinische Unterstützungssysteme im Bereich Smart-Applications. In: *i-com, Zeitschrift für interaktive und kooperative Medien*, 8(1), 2009, S. 45–51 (Zitiert auf Seite 232)

LITERATURVERZEICHNIS

- [KLAR u. a. 2003] KLAR, E. ; BIRTH, M. ; BECHSTEIN, W. ; GÜNTHER, R. ; TACKE, J. ; HELMBERGER, T. ; FREUND, M.: Multimodale Therapie von Lebermetastasen. In: *Chirurgische Gastroenterologie*, 19, 2003, S. 384–390 (Zitiert auf Seite 13)
- [KOHLMANN u. a. 2007] KOHLMANN, P. ; BRUCKNER, S. ; KANITSAR, A. ; GRÖLLER, M. E.: LiveSync: Deformed Viewing Spheres for Knowledge-Based Navigation. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 13(5), 2007, S. 1544–1551 (Zitiert auf Seite 91)
- [KOHLMANN u. a. 2008] KOHLMANN, P. ; BRUCKNER, S. ; KANITSAR, A. ; GRÖLLER, M. E.: LiveSync++: Enhancements of an Interaction Metaphor. In: *Proc. of Graphics Interface (GI)*, 2008, S. 81–88 (Zitiert auf den Seiten 91 und 247)
- [KONRAD-VERSE u. a. 2004] KONRAD-VERSE, O. ; PREIM, B. ; LITTMANN, A.: Virtual Resection with a Deformable Cutting plane. In: *Proc. of Simulation and Visualization (SimVis)*, 2004, S. 203–214 (Zitiert auf den Seiten 23, 221 und 228)
- [KOOP u. a. 2008] KOOP, D. ; SCHEIDEGGER, C. E. ; CALLAHAN, S. P. ; FREIRE, J. ; SILVA, C. T.: VisComplete: Automating Suggestions for Visualization Pipelines. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 14(6), 2008, S. 1691–1698 (Zitiert auf Seite 157)
- [KOVACIC u. a. 1998] KOVACIC, S. ; LEONARDIS, A. ; PERNUS, F.: Planning Sequences of Views for 3-D Object Recognition and Pose Determination. In: *Pattern Recognition*, 31, 1998, S. 1407–1417 (Zitiert auf Seite 63)
- [KREKEL u. a. 2009] KREKEL, P. R. ; BRUIN, P. W. de ; VALSTAR, E. R. ; POST, F. H. ; ROZING, P. M. ; BOTHA, C. P.: Evaluation of Bone Impingement Prediction in Pre-operative Planning for Shoulder Arthroplasty. In: *Journal of Engineering in Medicine*, 223, 2009, S. 813–822 (Zitiert auf Seite 20)
- [KRÜGER u. a. 2008] KRÜGER, A. ; STAMPE, K. ; IRRGANG, S. ; RICHTER, I. ; STRAUSS, G. ; PREIM, B.: Eingabegeräte und Interaktionstechniken für die virtuelle Endoskopie. In: *Proc. of Mensch und Computer*, 2008, S. 237–246 (Zitiert auf Seite 55)
- [KRÖGER u. a. 2009] KRÖGER, T. ; PANNIER, S. ; KALISKE, M. ; ALTROGGE, I. ; W. GRAF, T. P.: Optimal Applicator Placement in Hepatic Radiofrequency Ablation on the Basis of Rare Data. In: *Computer Methods in Biomechanics and Biomedical Engineering*, 13(3), 2009, S. 431–440 (Zitiert auf den Seiten 13 und 223)
- [KURLANDER & FEINER 1992] KURLANDER, D. ; FEINER, S.: A history-based macro by example system. In: *Proc. of ACM Symposium on User Interface Software and Technology (UIST)*, 1992, S. 99–106 (Zitiert auf Seite 158)
- [LAMADÉ u. a. 2000] LAMADÉ, W. ; GLOMBITZA, G. ; FISCHER, L. ; CHIU, P.: The Impact of 3-Dimensional Reconstructions on Operation Planning in Liver Surgery. In: *Archives of Surgery*, 135, 2000, S. 1256–1261 (Zitiert auf Seite 17)
- [LAMADÉ u. a. 2005] LAMADÉ, W. ; ULMER, C. ; RITTER, F. ; PEITGEN, H.-O. ; OTTO, G. ; THON, K.-P.: Virtuelle Operationsplanung: Aktueller Stand und

- Perspektiven. In: *Chirurgische Gastroenterologie*, 21, 2005, S. 7–13 (Zitiert auf Seite 27)
- [LANG u. a. 2001] LANG, H. ; MALAGO, M. ; TESTA, G. ; HERTL, M. ; NADALIN, S. ; BROELSCH, C. E.: Leberlebenspende und Lebertransplantation. In: *Lebenslinien*, (1), 2001 (Zitiert auf den Seiten 11 und 14)
- [LANG u. a. 2005a] LANG, H. ; RADTKE, A. ; HINDENNACH, M. ; SCHROEDER, T. ; FRÜHAUF, N. R. ; MALAGÓ, M. ; BOURQUAIN, H. ; PEITGEN, H.-O. ; OLDHAFFER, K. J. ; BROELSCH, C. E.: Impact of Virtual Tumor Resection and Computer-Assisted Risk Analysis on Operation Planning and Intraoperative Strategy in Major Hepatic Resection. In: *Archives of Surgery*, 140, 2005, S. 629–638 (Zitiert auf Seite 17)
- [LANG u. a. 2005b] LANG, H. ; RADTKE, A. ; LIE, C. ; SOTIROPOULOS, G. C. ; HINDENNACH, M. ; SCHROEDER, T. ; PEITGEN, H.-O. ; BROELSCH, C. E.: Improved Assessment of Functional Resectability in Repeated Hepatectomy by Computer-assisted Operation Planning. In: *Hepato-Gastroenterology*, 52, 2005, S. 1645–1648 (Zitiert auf Seite 10)
- [LANGE u. a. 2004] LANGE, T. ; EULENSTEIN, S. ; HUNERBEIN, M. ; LAMECKER, H. ; SCHLAG, P. M.: Augmenting Intraoperative 3D Ultrasound with Preoperative Models for Navigation in Liver Surgery. In: *Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2004, S. 534–541 (Zitiert auf Seite 99)
- [LEVENSHTEIN 1966] LEVENSHTEIN, V. I.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. In: *Soviet Physics Doklady*, 10(8), 1966, S. 707–710 (Zitiert auf Seite 150)
- [LI u. a. 2007] LI, W. ; RITTER, L. ; AGRAWALA, M. ; CURLLESS, B. ; SALESIN, D.: Interactive Cutaway Illustrations of Complex 3D Models. In: *Proc. of ACM SIGGRAPH Computer Graphics*, 2007 (Zitiert auf den Seiten 59 und 129)
- [LI u. a. 2008] LI, W. ; AGRAWALA, M. ; CURLLESS, B. ; SALESIN, D.: Automated generation of interactive 3D exploded view diagrams. In: *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2008)*, 27(3), 2008, S. 1–7 (Zitiert auf Seite 40)
- [LIN 1991] LIN, J.: Divergence measures based on the Shannon entropy. In: *IEEE Transactions on Information Theory*, 37(1), 1991, S. 145–151 (Zitiert auf Seite 60)
- [LIND & KJELLIN 2005] LIND, M. ; KJELLIN, A.: Faster is Better: Optimal Speed of Animated Visualizations for Decision Makers. In: *Proc. of IEEE Information Visualization (InfoVis)*, 2005 (Zitiert auf Seite 247)
- [LINK u. a. 2006] LINK, F. ; KOENIG, M. ; PEITGEN, H.-O.: Multi-Resolution Volume Rendering with per Object Shading. In: *Proc. of Vision, Modeling, and*

LITERATURVERZEICHNIS

- Visualization Workshop (VMV)*, 2006, S. 185–191 (Zitiert auf den Seiten [19](#), [190](#) und [196](#))
- [LIU u. a. 2003] LIU, A. ; TENDICK, F. ; CLEARY, K. ; KAUFMANN, C.: A survey of surgical simulation: applications, technology, and education. In: *Presence: Teleoperators and Virtual Environments*, 12(6), 2003, S. 599–614 (Zitiert auf Seite [212](#))
- [LO u. a. 1997] LO, C.-M. ; FAN, S.-T. ; UU, C.-L. ; WEI, W. ; LO, R. ; LAI, C.-L. ; CHAN, J. ; NG, I. ; FUNG, A. ; WONG, J.: Adult-to-adult living donor liver transplantation using extended right lobe grafts. In: *Annals of Surgery*, 226(3), 1997, S. 261–270 (Zitiert auf Seite [14](#))
- [LOBER u. a. 2001] LOBER, W. B. ; TRIGG, L. J. ; BLISS, D. ; BRINKLEY, J. F.: IML: An Image Markup Language. In: *Proc. of American Medical Informatics Association Fall Symposium*, 2001, S. 403–407 (Zitiert auf Seite [131](#))
- [LOGGE u. a. 2007] LOGGE, C. ; CORDES, J. ; MÜHLER, K. ; PREIM, B. ; HILLERT, C.: Training in Living Donor Liver Transplantation - The LiverSurgeryTrainer. In: *Proc. of ILTS 13th Annual International Congress*, 2007 (Zitiert auf Seite [212](#))
- [LU u. a. 2005] LU, J. ; PAN, Z. ; LIN, H. ; ZHANG, M. ; SHI, J.: Virtual learning environment for medical education based on VRML and VTK. In: *Computers and Graphics*, 29(2), 2005, S. 283–288 (Zitiert auf Seite [33](#))
- [MA 1999] MA, K.-L.: Image graphs - a novel approach to visual data exploration. In: *Proc. of IEEE Visualization (VIS)*, 1999, S. 81–88 (Zitiert auf Seite [156](#))
- [MANSOUR u. a. 2001] MANSOUR, I. H. ; FURUIE, S. S. ; NEDEL, L. P. ; FREITAS, C. M. D. S.: A Framework to Visualize and Interact with Multimodal Medical Images. In: *Proc. of Joint IEEE/TVCG and Eurographics Workshop on Volume Graphics (VG)*, 2001, S. 385–398 (Zitiert auf Seite [187](#))
- [MARKERT u. a. 2006] MARKERT, M. ; WEBER, S. ; KLEEMANN, M. ; BRUCH, H. ; LUETH, T.: Comparison of fundamental requirements for soft tissue navigation with a novel assistance system for open liver surgery. In: *International Journal of Computer Assisted Radiology and Surgery*, 2006, S. 498–499 (Zitiert auf Seite [99](#))
- [MARKS u. a. 1997] MARKS, J. ; ANDALMAN, B. ; BEARDSLEY, P. A. ; FREEMAN, W. ; GIBSON, S. ; HODGINS, J. ; KANG, T. ; MIRTICH, B. ; PFISTER, H. ; RUMML, W. ; RYALL, K. ; SEIMS, J. ; SHIEBER, S.: Design galleries: a general approach to setting parameters for computer graphics and animation. In: *Proc. of International Conference on Computer Graphics and Interactive Techniques*, 1997, S. 389–400 (Zitiert auf Seite [156](#))
- [MEINZER u. a. 2002] MEINZER, H.-P. ; THORN, M. ; CÁRDENAS, C. E.: Computerized planning of liver surgery—an overview. In: *Computers and Graphics*, 26(4), 2002, S. 569–576 (Zitiert auf Seite [17](#))

- [VAN MERRIËNBOER u. a. 2002] MERRIËNBOER, J. van ; CLARK, R. ; CROOCK, M. de: Blueprints for Complex Learning: The 4C/ID-Model. In: *Educational Technology Research and Development*, 50(2), 2002, S. 39–64 (Zitiert auf Seite 215)
- [METCALF 1920] METCALF, Z. P.: Labeling Illustrations. In: *Transactions of the American Microscopical Society*, 39(2), 1920, S. 149–162 (Zitiert auf Seite 123)
- [MEVIS MEDICAL SOLUTIONS 2005] MEVIS MEDICAL SOLUTIONS: *MeVis - Die ersten zehn Jahre 1995-2005*. 2005 (Zitiert auf Seite 23)
- [MEVIS MEDICAL SOLUTIONS - DISTANT SERVICES 2010] MEVIS MEDICAL SOLUTIONS - DISTANT SERVICES: 2010. – http://mms.mevis.de/Distant_Services.html. – besucht am 5.2.2010 (Zitiert auf den Seiten 15, 38 und 118)
- [MÜHLER 2005] MÜHLER, K.: *Skriptbasierte Generierung von Animationen für die medizinische Ausbildung und Therapieplanung*, Universität Magdeburg, Diplomarbeit, 2005 (Zitiert auf Seite 39)
- [MÜHLER u. a. 2006a] MÜHLER, K. ; BADE, R. ; PREIM, B.: Adaptive script based animations for intervention planning. In: *Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2006, S. 478–485 (Zitiert auf den Seiten 35 und 153)
- [MÜHLER u. a. 2006b] MÜHLER, K. ; BADE, R. ; PREIM, B.: Skriptbasierte Animationen fuer die Operationsplanung und Ausbildung. In: *Proc. of Bildverarbeitung für die Medizin (BVM)*, 2006, S. 296–300 (Zitiert auf Seite 35)
- [MÜHLER u. a. 2007a] MÜHLER, K. ; CORDES, J. ; PREIM, B. ; HILLERT, C. ; OLDHAFER, K.: *LiverSurgeryTrainer 1.0 (DVD)*. Universität Magdeburg, 2007 (Zitiert auf Seite 212)
- [MÜHLER u. a. 2007b] MÜHLER, K. ; NEUGEBAUER, M. ; TIETJEN, C. ; PREIM, B.: Viewpoint Selection for Intervention Planning. In: *Proc. of IEEE/Eurographics Symposium on Visualization (EuroVis)*, 2007, S. 267–274 (Zitiert auf den Seiten 50, 55 und 94)
- [MÜHLER u. a. 2008a] MÜHLER, K. ; HANSEN, C. ; NEUGEBAUER, M. ; PREIM, B.: Automatische Kamerapositionierung für intra-operative Visualisierungen in der onkologischen Leberchirurgie. In: *Proc. of Bildverarbeitung für die Medizin (BVM)*, 2008, S. 143–147 (Zitiert auf den Seiten 50, 94 und 99)
- [MÜHLER u. a. 2008b] MÜHLER, K. ; CORDES, J. ; TIETJEN, C. ; PREIM, B.: Richtlinien für die Gestaltung chirurgischer Planungs- und Trainings-Software. In: *Proc. of Jahrestagung der Deutschen Gesellschaft für Computer- und Robotergestützte Chirurgie (CURAC)*, 2008, S. 241–242 (Zitiert auf Seite 212)
- [MÜHLER & PREIM 2009a] MÜHLER, K. ; PREIM, B.: Automatic Textual Annotation for Surgical Planning. In: *Proc. of Vision, Modeling, and Visualization Workshop (VMV)*, 2009, S. 277–284 (Zitiert auf Seite 118)

LITERATURVERZEICHNIS

- [MÜHLER & PREIM 2009b] MÜHLER, K. ; PREIM, B.: Automatische Annotation medizinischer 2D- und 3D-Visualisierungen. In: *Proc. of Bildverarbeitung für die Medizin (BVM)*, 2009, S. 11–15 (Zitiert auf Seite 118)
- [MÜHLER u. a. 2009] MÜHLER, K. ; TIETJEN, C. ; PREIM, B.: The Medical Exploration Toolkit. In: *Proc. of IEEE/Eurographics Symposium on Visualization (EuroVis) (Poster)*, 2009 (Zitiert auf Seite 179)
- [MÜHLER & PREIM 2010a] MÜHLER, K. ; PREIM, B.: Günstige Kamerapfade für medizinische Animationen. In: *Proc. of Bildverarbeitung für die Medizin (BVM)* (erscheint), 2010 (Zitiert auf den Seiten 94 und 106)
- [MÜHLER u. a. 2010] MÜHLER, K. ; TIETJEN, C. ; RITTER, F. ; PREIM, B.: The Medical Exploration Toolkit: An Efficient Support for Visual Computing in Surgical Planning and Training. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 16(1), 2010, S. 133–146 (Zitiert auf den Seiten 179, 197, 199, 201 und 203)
- [MÜHLER & PREIM 2010b] MÜHLER, K. ; PREIM, B.: Reusable Visualizations and Animations for Surgery Planning. In: *Computer Graphics Forum*, 29(2), 2010, S. 1103–1112 (Zitiert auf Seite 154)
- [MICROSOFT CORPORATION 1999] MICROSOFT CORPORATION: *Microsoft Windows User Experience*. Microsoft Press, 1999 (Zitiert auf Seite 230)
- [MÖNCH In Vorbereitung] MÖNCH, J.: *Erstellung von Richtlinien für die Konzeption und Entwicklung chirurgischer Ausbildungs- und Trainingssysteme*, Universität Magdeburg, Dissertation, In Vorbereitung (Zitiert auf Seite 215)
- [MONTGOMERY u. a. 2004] MONTGOMERY, K. ; STEPHANIDES, M. ; SCHENDEL, S. ; ROSS, M.: User interface paradigms for patient-specific surgical planning: lessons learned over a decade of research. In: *Computerized Medical Imaging and Graphics*, 29(2), 2004, S. 203–222 (Zitiert auf Seite 230)
- [MUTTER u. a. 2005] MUTTER, D. ; RUBINO, F. ; GUY TEMPORAL, M. ; MARESCAUX, J.: Surgical education and internet-based simulation: The World Virtual University. In: *Minimally Invasive Therapy and Allied Technologies*, 14(4–5), 2005, S. 267–274 (Zitiert auf Seite 31)
- [NAUTA 1998] NAUTA, R. J.: *Hepatobiliary and Pancreatic Surgery: Imaging Strategies and Surgical Decision Making*. Kap. Surgical Summary: Liver Imaging, S. 413–427, Wiley, John & Sons, Incorporated, 1998 (Zitiert auf Seite 27)
- [NEUGEBAUER u. a. 2007] NEUGEBAUER, M. ; MÜHLER, K. ; TIETJEN, C. ; PREIM., B.: Automatische Kamerapositionierung in komplexen medizinischen 3D-Visualisierungen. In: *Proc. of Bildverarbeitung für die Medizin (BVM)*, 2007, S. 318–322 (Zitiert auf Seite 50)

- [NEUGEBAUER u. a. 2009] NEUGEBAUER, M. ; GASTEIGER, R. ; BEUING, O. ; DIEHL, V. ; SKALEJ, M. ; PREIM, B.: Combining Map Displays and 3D Visualizations for the Analysis of Scalar Data on Cerebral Aneurysm Surfaces. In: *Computer Graphics Forum (CGF)* 28, 2009, S. 895–902 (Zitiert auf Seite 56)
- [NIELSEN 1993] NIELSEN, J.: *Usability Engineering*. Morgan Kaufmann, 1993 (Zitiert auf Seite 240)
- [NIMSKY u. a. 2006] NIMSKY, C. ; GANSLANDT, O. ; ENDERS, F. ; MERHOF, D. ; HAMMEN, T. ; BUCHFELDER, M.: Visualization Strategies for Major White Matter Tracts for Intraoperative Use. In: *International Journal of Computer Assisted Radiology and Surgery*, 1(1), 2006, S. 13–22 (Zitiert auf Seite 100)
- [OELTZE & PREIM 2004] OELTZE, S. ; PREIM, B.: Visualization of Anatomic Tree Structures with Convolution Surfaces. In: *Proc. of IEEE/Eurographics Symposium on Visualization (EuroVis)*, 2004, S. 311–320 (Zitiert auf Seite 19)
- [OKAMOTO u. a. 1984] OKAMOTO, E. ; KYO, A. ; YAMANAKA, N. ; TANAKA, N. ; KUWATA, K.: Prediction of the safe limits of hepatectomy by combined volumetric and functional measurements in patients with impaired hepatic function. In: *Surgery*, 95(5), 1984, S. 586–592 (Zitiert auf Seite 17)
- [OLBRICH 2006] OLBRICH, M.: *Beschriftung von 3D-Modellen*, Technische Universität Dresden, Diplomarbeit, 2006 (Zitiert auf Seite 126)
- [OLDHAFFER u. a. 2001] OLDHAFFER, K. J. ; PREIM, B. ; DÖRGE, C. ; PEITGEN, H.-O. ; BROELSCH, C. E.: Akzeptanz einer computergestützten Operationsplanung in der Viszeralchirurgie - Ergebnisse einer bundesweiten Umfrage. In: *Zentralblatt für Chirurgie*, 127(2), 2001, S. 128–133 (Zitiert auf Seite 17)
- [OLDHAFFER u. a. 2006] OLDHAFFER, K. J. ; BOURQUAIN, H. ; STAVROU, G. A.: Die Leberresektion (Teil I) - Anatomische Grundlagen, Indikationen und präoperative Planung. In: *Zentralblatt für Chirurgie*, 131(2), 2006, S. W22–W35 (Zitiert auf Seite 12)
- [OLDHAFFER & STAVROU 2006] OLDHAFFER, K. J. ; STAVROU, G. A.: Die Leberresektion (Teil II) - Technik, Durchführung und Chirurgisch-Onkologische Aspekte. In: *Zentralblatt für Chirurgie*, 131, 2006, S. W54–W66 (Zitiert auf den Seiten 11 und 12)
- [OPEN SOFTWARE FOUNDATION 1992] OPEN SOFTWARE FOUNDATION: *Osf/Motif Style Guide: Revision 1.2*. Prentice Hall, 1992 (Zitiert auf Seite 230)
- [OUD 2009] OUD, J.: Guidelines for effective online instruction using multimedia screencasts. In: *Reference Services Review*, 37(2), 2009, S. 164–177 (Zitiert auf Seite 242)
- [PFLESSER u. a. 2002] PFLESSER, B. ; PETERSIK, A. ; TIEDE, U. ; HÖHNE, K. H. ; LEUWER, R.: Volume cutting for virtual petrous bone surgery. In: *Computer Aided Surgery*, 7(2), 2002, S. 74–83 (Zitiert auf Seite 212)

LITERATURVERZEICHNIS

- [PHILIP u. a. 2008] PHILIP, C. T. ; UNRUH, K. P. ; LACHMAN, N. ; PAWLINA, W.: An explorative learning approach to teaching clinical anatomy using student generated content. In: *Anatomical Sciences Education*, 1(3), 2008, S. 106–110 (Zitiert auf Seite 33)
- [PREIM u. a. 1995] PREIM, B. ; RITTER, A. ; STROTHOTTE, T. ; POHLE, T. ; FORSEY, D. R. ; BARTRAM, L.: Consistency of Rendered Images and Their Textual Labels. In: *Proc. of Computer Graphics*, 1995, S. 201–210 (Zitiert auf Seite 130)
- [PREIM & STROTHOTTE 1996] PREIM, B. ; STROTHOTTE, T.: Annotating and Interacting with 3D-Models for Explanation Purposes: System Architecture. In: *Proc. of European Conference on Artificial Intelligence*, 1996 (Zitiert auf Seite 126)
- [PREIM u. a. 1996] PREIM, B. ; RITTER, A. ; STROTHOTTE, T.: Illustrating Anatomic Models: A Semi-Interactive Approach. In: *Proc. of International Conference on Visualisation in Biomedical Computing*, 1996, S. 23–32 (Zitiert auf Seite 40)
- [PREIM & RAAB 1998] PREIM, B. ; RAAB, A.: Annotation von topographisch komplizierten 3D-Modellen. In: *Proc. of Simulation and Visualization (SimVis)*, 1998, S. 128–140 (Zitiert auf den Seiten 126 und 127)
- [PREIM u. a. 2002] PREIM, B. ; TIETJEN, C. ; SPINDLER, W. ; PEITGEN, H.-O.: Integration of Measurement Tools in Medical Visualizations. In: *Proc. of IEEE Visualization (VIS)*, 2002, S. 21–28 (Zitiert auf Seite 95)
- [QUALTER u. a. 2004] QUALTER, J. ; TRIOLA, M. M. ; WEINER, M. J. ; HOPKINS, M. A. ; KIROV, M. ; NACHBAR, M. S.: The Virtual Surgery Patient: Development of a digital, three-dimensional model of human anatomy designed for surgical. In: *Proc. of IEEE Symposium on Computer-Based Medical Systems*, 2004, S. 38 (Zitiert auf Seite 33)
- [RAAB u. a. 1998] RAAB, R. ; SCHLITT, H. J. ; OLDHAFFER, K.: Technik der Leberresektion. In: *Chirurgische Gastroenterologie*, 14, 1998, S. 213–218 (Zitiert auf Seite 12)
- [RADTKE u. a. 2006] RADTKE, A. ; BOCKHORN, M. ; LANG, T. S. H. ; PAUL, A. ; NADALIN, S. ; SANER, F. ; SCHENK, A. ; BROELSCH, C. ; MALAGO, M.: Computer-assisted planning in living donor liver operation. In: *Zentralblatt für Chirurgie*, 131(1), 2006, S. 69–74 (Zitiert auf Seite 17)
- [RAIA u. a. 1989] RAIA, S. ; NERY, J. R. ; MIES, S.: Liver transplantation from live donors. In: *Lancet*, 2, 1989, S. 1042–1043 (Zitiert auf Seite 14)
- [RAU u. a. 2003] RAU, H. ; ZIMMERMANN, A. ; WARDEMANN, C. ; SCHILDBERG, F.: Standards in der Operationstechnik bei Lebermetastasen. In: *Chirurgische Gastroenterologie*, 19(4), 2003, S. 333–339 (Zitiert auf Seite 15)
- [RAUTEK u. a. 2008] RAUTEK, P. ; BRUCKNER, S. ; GRÖLLER, M. E.: Interaction-Dependent Semantics for Illustrative Volume Rendering. In: *Computer Graphics Forum*, 27(3), 2008, S. 847–854 (Zitiert auf Seite 157)

- [REITINGER u. a. 2006] REITINGER, B. ; BORNIG, A. ; BEICHEL, R. ; SCHMALSTIEG, D.: Liver Surgery Planning Using Virtual Reality. In: *IEEE Computer Graphics and Applications (CGA)*, 26(6), 2006, S. 36–47 (Zitiert auf Seite 23)
- [REXILIUS u. a. 2006] REXILIUS, J. ; KUHNIGK, J.-M. ; HAHN, H. K. ; PEITGEN, H.-O.: An Application Framework for Rapid Prototyping of Clinically Applicable Software Assistants. In: *Proc. of Informatik für Menschen - Band 1*, 2006, S. 522–528 (Zitiert auf den Seiten 181, 188 und 190)
- [REZK-SALAMA u. a. 2000] REZK-SALAMA, C. ; HASTREITER, P. ; SCHERER, J. ; GREINER, G.: Automatic Adjustment of Transfer Functions for 3D Volume Visualization. In: *Proc. of Vision, Modeling, and Visualization Workshop (VMV)*, 2000, S. 357–364 (Zitiert auf Seite 39)
- [REZK-SALAMA u. a. 2006] REZK-SALAMA, C. ; KELLER, M. ; KOHLMANN, P.: High-Level User Interfaces for Transfer Function Design with Semantics. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 11(5), 2006, S. 1021–1028 (Zitiert auf Seite 157)
- [RITTER u. a. 2001] RITTER, F. ; DEUSSEN, O. ; PREIM, B. ; STROTHOTTE, T.: Virtual 3D Puzzles: A New Method for Exploring Geometric Models in VR. In: *IEEE Computer Graphics and Applications (CGA)*, 21(5), 2001, S. 11–13 (Zitiert auf Seite 109)
- [RITTER u. a. 2009] RITTER, F. ; HANSEN, C. ; WILKENS, K. ; KOEHN, A. ; PEITGEN, H.-O.: User Interfaces for Direct Interaction with 3D Planning Data in the Operating Room. In: *i-com, Zeitschrift für interaktive und kooperative Medien*, 8(1), 2009, S. 24–31 (Zitiert auf den Seiten 30 und 231)
- [RÖSSLER u. a. 2007] RÖSSLER, F. ; NENOV, M. ; ISEHARDT-BAUER, S. ; HASTREITER, P. ; ERTL, T.: Investigating 3D Object Movies for Web-Based Medical Visualization. In: *Proc. of Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, 2007, S. 209–212 (Zitiert auf Seite 39)
- [RÖSSLER u. a. 2008] RÖSSLER, F. ; BOTCHEN, R. P. ; ERTL, T.: Dynamic Shader Generation for Flexible Multi-Volume Visualization. In: *Proc. of IEEE Pacific Visualization Symposium (PacificVis)*, 2008, S. 17–24 (Zitiert auf den Seiten 185, 186 und 187)
- [ROBERTSON u. a. 2006] ROBERTSON, G. G. ; CZERWINSKI, M. P. ; TAN, D.: *3D navigation techniques*. United States Patent 7148892. 2006 (Zitiert auf Seite 64)
- [ROPINSKI u. a. 2007] ROPINSKI, T. ; PRASSNI, J.-S. ; ROTERS, J. ; HINRICHS, K. H.: Internal Labels as Shape Cues for Medical Illustration. In: *Proc. of Vision, Modeling, and Visualization Workshop (VMV)*, 2007, S. 203–212 (Zitiert auf den Seiten 119, 127 und 128)

LITERATURVERZEICHNIS

- [RÖSSLING u. a. 2009] RÖSSLING, I. ; CYRUS, C. ; DORNHEIM, L. ; HAHN, P. ; PREIM, B. ; BOEHM, A.: Interaktive Visualisierung von Abständen und Ausdehnungen anatomischer Strukturen für die Interventionsplanung. In: *Proc. of Bildverarbeitung für die Medizin (BVM)*, 2009, S. 381–385 (Zitiert auf Seite 29)
- [RYAN u. a. 2004] RYAN, J. ; O’SULLIVAN, C. ; BELL, C.: Real-time interactive volumetric animation of the heart’s electrical cycle from automatically synchronized ECG: Research Articles. In: *Computer Animation and Virtual Worlds*, 15(3–4), 2004, S. 353–360 (Zitiert auf Seite 40)
- [SAP 2001] SAP: *SAP Style Guide*. 2001. – <http://help.sap.com/printdocu/core/Print46c/en/data/pdf/BCDWBERG/BCDWBERG.pdf>. – besucht am 2.2.2010 (Zitiert auf Seite 230)
- [SBERT u. a. 2005] SBERT, M. ; PLEMENOS, D. ; FEIXAS, M. ; GONZÁLEZ, F.: Viewpoint Quality: Measures and Applications. In: *Proc. of Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging*, 2005, S. 185–192 (Zitiert auf den Seiten 57, 58 und 87)
- [SCHEIDEGGER u. a. 2007] SCHEIDEGGER, C. ; VO, H. ; KOOP, D. ; FREIRE, J. ; SILVA, C.: Querying and Creating Visualizations by Analogy. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 13(6), 2007, S. 1560–1567 (Zitiert auf den Seiten 157 und 185)
- [SCHEMENAUR & PAWLICK 2007] SCHEMENAUR, P. J. ; PAWLICK, C.: Evaluating guidelines for writing user interface text. In: *Proc. of ACM International Conference on Design of Communication (SIGDOC)*, 2007, S. 216–220 (Zitiert auf Seite 230)
- [SCHENK u. a. 2008] SCHENK, A. ; ZIDOWITZ, S. ; BOURQUAIN, H. ; HINDEN-NACH, M. ; HANSEN, C. ; HAHN, H. ; PEITGEN, H.-O.: Clinical relevance of model based computer-assisted diagnosis and therapy. In: *Proc. of SPIE Medical Imaging: Computer-Aided Diagnosis* 6915, 2008, S. 691502_1–19 (Zitiert auf den Seiten 23 und 29)
- [SCHROEDER & RUEHM 2005] SCHROEDER, T. ; RUEHM, S. G.: Bildgebende Diagnostik bei Lebertransplantationen - Präoperative Evaluierung und postoperative Komplikationen. In: *Der Radiologe*, 45(1), 2005, S. 34–43 (Zitiert auf Seite 14)
- [SCHUMANN u. a. 2007] SCHUMANN, C. ; OELTZE, S. ; BADE, R. ; PREIM, B.: Model-free Surface Visualization of Vascular Trees. In: *Proc. of IEEE/Eurographics Symposium on Visualization (EuroVis)*, 2007, S. 283–290 (Zitiert auf Seite 193)
- [SEIFFERT u. a. 2000] SEIFFERT, C. ; CLASBRUMMEL, B. ; GEMÜNDEN, H. G. ; BOLZ, A.: Strategieentwicklung für ein telemedizinisches Dienstleistungsunternehmen am Beispiel der TRAUMA GmbH. In: *Biomedizinische Technik*, 45, 2000, S. 525–526 (Zitiert auf Seite 32)

- [SELLE u. a. 2002] SELLE, D. ; PREIM, B. ; SCHENK, A. ; PEITGEN, H.-O.: Analysis of Vasculature for Liver Surgical Planning. In: *IEEE Transactions on Medical Imaging*, 21, 2002, S. 1344–1357 (Zitiert auf Seite 17)
- [SHNEIDERMAN & PLAISANT 2004] SHNEIDERMAN, B. ; PLAISANT, C.: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson Addison Wesley, 2004 (Zitiert auf Seite 226)
- [SHOEMAKE 1992] SHOEMAKE, K.: ARCBALL: a user interface for specifying three-dimensional orientation using a mouse. In: *Proc. of Graphics Interface (GI)*, 1992, S. 151–156 (Zitiert auf Seite 55)
- [SIMBIONIX USA CORP.] SIMBIONIX USA CORP.: *Symbionix, medical training simulators and clinical devices for MIS (minimally invasive surgery)*. – <http://www.symbionix.com/>. – besucht am 25. 11. 2009 (Zitiert auf den Seiten 31 und 212)
- [SOKOLOV & PLEMENOS 2005] SOKOLOV, D. ; PLEMENOS, D.: Viewpoint quality and scene understanding. In: *Proc. of Eurographics Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*, 2005, S. 67–73 (Zitiert auf Seite 58)
- [SOKOLOV u. a. 2006a] SOKOLOV, D. ; PLEMENOS, D. ; TAMINE, K.: Methods and data structures for virtual world exploration. In: *The Visual Computer*, 22(7), 2006, S. 506–516 (Zitiert auf Seite 108)
- [SOKOLOV u. a. 2006b] SOKOLOV, D. ; PLEMENOS, D. ; TAMINE, K.: Viewpoint quality and global scene exploration strategies. In: *Proc. of International Conference on Computer Graphics Theory and Applications (GRAPP)*, 2006, S. 184–191 (Zitiert auf Seite 58)
- [SOKOLOV & PLEMENOS 2008] SOKOLOV, D. ; PLEMENOS, D.: Virtual world explorations by using topological and semantic knowledge. In: *The Visual Computer*, 24(3), 2008, S. 173–185 (Zitiert auf Seite 109)
- [SONNET u. a. 2004] SONNET, H. ; CARPENDALE, S. ; STROTHOTTE, T.: Integrating expanding annotations with a 3D explosion probe. In: *Proc. of Working Conference on Advanced Visual Interfaces (AVI)*, 2004, S. 63–70 (Zitiert auf Seite 127)
- [SOURIN u. a. 2000] SOURIN, A. ; SOURINA, O. ; SEN, H. T.: Virtual Orthopedic Surgery Training. In: *IEEE Computer Graphics and Applications (CGA)*, 20(3), 2000, S. 6–9 (Zitiert auf Seite 32)
- [SPITZER u. a. 1996] SPITZER, V. ; ACKERMAN, M. ; SCHERZINGER, A. ; WHITLOCK, D.: The visible human male: a technical report. In: *Journal of the American Medical Informatics Association*, 3(2), 1996, S. 118–130 (Zitiert auf den Seiten 31, 32 und 33)

LITERATURVERZEICHNIS

- [STALLING u. a. 2005] STALLING, D. ; WESTERHOFF, M. ; HEGE, H.-C.: Amira: A Highly Interactive System for Visual Data Analysis. In: *The Visualization Handbook*. Elsevier, 2005, Kap. 38, S. 749–767 (Zitiert auf Seite 188)
- [STEIN & DÉCORET 2008] STEIN, T. ; DÉCORET, X.: Dynamic Label Placement for Improved Interactive Exploration. In: *Proc. of International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, 2008, S. 15–21 (Zitiert auf Seite 130)
- [STOEV & STRASSER 2002] STOEV, S. L. ; STRASSER, W.: A case study on automatic camera placement and motion for visualizing historical data. In: *Proc. of IEEE Visualization (VIS)*, 2002, S. 545–548 (Zitiert auf Seite 59)
- [STRONG u. a. 1990] STRONG, R. W. ; LYNCH, S. V. ; ONG, T. H. ; MATSUNAMI, H. ; KOIDO, Y. ; BALDERSON, G. A.: Successful liver transplantation from a living donor to her son. In: *The New England Journal of Medicine*, 322, 1990, S. 1505–1507 (Zitiert auf Seite 14)
- [SUN MICROSYSTEMS 2001] SUN MICROSYSTEMS: *Java Look and Feel Design Guidelines*. Addison-Wesley Longman, 2001 (Zitiert auf Seite 230)
- [SVAKHINE u. a. 2005] SVAKHINE, N. ; EBERT, D. ; STREDNEY, D.: Illustration motifs for effective medical volume illustration. In: *IEEE Computer Graphics and Applications (CGA)*, 25(3), 2005, S. 31–39 (Zitiert auf Seite 157)
- [TAKAHASHI u. a. 2005] TAKAHASHI, S. ; FUJISHIRO, I. ; TAKESHIMA, Y. ; NISHITA, T.: A Feature-Driven Approach to Locating Optimal Viewpoints for Volume Visualization. In: *Proc. of IEEE Visualization (VIS)*, 2005, S. 495–502 (Zitiert auf Seite 60)
- [TAKAHASHI u. a. 2006] TAKAHASHI, S. ; YOSHIDA, K. ; SHIMADA, K. ; NISHITA, T.: Occlusion-Free Animation of Driving Routes for Car Navigation Systems. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 12(5), 2006, S. 1141–1148 (Zitiert auf Seite 63)
- [TAN u. a. 2001] TAN, D. S. ; ROBERTSON, G. G. ; CZERWINSKI, M.: Exploring 3D Navigation: Combining Speed-coupled Flying with Orbiting. In: *Proc. of Conference on Human Factors in Computing Systems (CHI)*, 2001, S. 418–425 (Zitiert auf Seite 64)
- [TEMPELMAN-KLUIT 2006] TEMPELMAN-KLUIT, N.: Multimedia learning theories and online instruction. In: *College and Research Libraries*, 64(4), 2006, S. 364–369 (Zitiert auf Seite 242)
- [THE BLENDER FOUNDATION 2009] THE BLENDER FOUNDATION: *Blender*. 2009. – <http://www.blender.org>. – besucht am 25.11.2009 (Zitiert auf Seite 158)
- [THEISEN-KUNDE u. a. 2007] THEISEN-KUNDE, D. ; HERRMANN, K. ; KLEEMANN, M. ; TEDSEN, S. ; DERFUSS, T. ; SALAMON, H. ; FALKENSTEIN, W. ; BRINKMANN, R. ; BRUCH, H. ; BIRNGRUBER, R.: Smartes Infrarot Laser-Skalpell (SILAS) für

- die minimal invasive Chirurgie - Ein Beispiel für erfolgreiche Industriekooperation. In: *Focus MUL*, 24(2), 2007, S. 104–109 (Zitiert auf Seite [213](#))
- [THORN u. a. 1999] THORN, M. ; SONNTAG, S. ; GLOMBITZA, G.-P. ; LAMADÉ, W. ; MEINZER, H.-P.: Ein interaktives Tool für die Segmenteinteilung der Leber in der chirurgischen Operationsplanung. In: *Proc. of Bildverarbeitung für die Medizin (BVM)*, 1999, S. 155–159 (Zitiert auf Seite [23](#))
- [THORN u. a. 2004a] THORN, M. ; YALCIN, B. ; SCHEMMER, P. ; GRENACHER, L. ; KRAUS, T. ; BÜCHLER, M. W. ; MEINZER, H.-P.: LiverLine - ein webbasiertes Informationssystem für die computergestützte Leberoperationsplanung. In: *Proc. of Bildverarbeitung für die Medizin (BVM)*, 2004, S. 333–337 (Zitiert auf Seite [23](#))
- [THORN u. a. 2004b] THORN, M. ; HEIMANN, T. ; SONEK, C. ; SCHÖBINGER, M. ; SCHEMMER, P. ; GRENACHER, L. ; RICHTER, G. M. ; KRAUS, T. ; BÜCHLER, M. ; MEINZER, H. P.: OrgaNicer - Eine Visualisierungssoftware für die Operationsplanung in der Leberchirurgie. In: *Proc. of Aktuelle Trends in der Softwareforschung - Tagungsband zum doIT Software-Forschungstag 2003*, 2004, S. 175–185 (Zitiert auf Seite [23](#))
- [TIETJEN u. a. 2006a] TIETJEN, C. ; PREIM, B. ; HERTEL, I. ; STRAUSS, G.: A Software-Assistant for Pre-operative Planning and Visualization of Neck Dissections. In: *Proc. of Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, 2006, S. 176–177 (Zitiert auf den Seiten [24](#), [45](#) und [202](#))
- [TIETJEN u. a. 2006b] TIETJEN, C. ; MEYER, B. ; SCHLECHTWEG, S. ; PREIM, B. ; HERTEL, I. ; STRAUSS, G.: Enhancing Slice-based Visualizations of Medical Volume Data . In: *Proc. of IEEE/Eurographics Symposium on Visualization (EuroVis)*, 2006, S. 123–130 (Zitiert auf Seite [196](#))
- [TIETJEN u. a. 2008] TIETJEN, C. ; MÜHLER, K. ; RITTER, F. ; KONRAD, O. ; HINDENNACH, M. ; PREIM, B.: METK - The Medical Exploration Toolkit. In: *Proc. of Bildverarbeitung für die Medizin (BVM)*, 2008, S. 407–411 (Zitiert auf Seite [179](#))
- [TIETJEN u. a. 2009] TIETJEN, C. ; KUBISCH, C. ; HILLER, S. ; PREIM, B.: GPU-basierte Smart Visibility Techniken für die Planung von Tumor-Operationen. In: *Proc. of Bildverarbeitung für die Medizin (BVM)*, 2009, S. 272–276 (Zitiert auf den Seiten [56](#) und [76](#))
- [TIETJEN 2009] TIETJEN, C.: *Illustrative Visualisierungstechniken zur Unterstützung der präoperativen Planung von chirurgischen Eingriffen*, Universität Magdeburg, Dissertation, 2009 (Zitiert auf den Seiten [1](#), [24](#), [26](#), [29](#), [38](#) und [180](#))
- [TUNUGUNTLA u. a. 2008] TUNUGUNTLA, R. ; RODRIGUEZ, O. ; RUIZ, J. ; QADRI, S. ; MINTZER, M. ; VAN ZUILEN, M. ; ROOS, B.: Computer-based animations and static graphics as medical student aids in learning home safety assessment: a randomized controlled trial. In: *Medical Teacher*, 30(8), 2008, S. 815–817 (Zitiert auf Seite [247](#))

LITERATURVERZEICHNIS

- [VANACKEN u. a. 2009] VANACKEN, L. ; GROSSMAN, T. ; CONINX, K.: Multi-modal selection techniques for dense and occluded 3D virtual environments. In: *International Journal of Human-Computer Studies*, 67(3), 2009, S. 237–255 (Zitiert auf Seite 200)
- [VÁZQUEZ u. a. 2001] VÁZQUEZ, P.-P. ; FEIXAS, M. ; SBERT, M. ; HEIDRICH, W.: Viewpoint Selection using Viewpoint Entropy. In: *Proc. of Vision, Modeling, and Visualization Workshop (VMV)*, 2001, S. 273–280 (Zitiert auf den Seiten 57, 60 und 89)
- [VÁZQUEZ & SBERT 2003] VÁZQUEZ, P.-P. ; SBERT, M.: Fast Adaptive Selection of Best Views. In: *Proc. of Computational Science and its Applications*, 2003, S. 295–305 (Zitiert auf Seite 59)
- [VÁZQUEZ u. a. 2006] VÁZQUEZ, P.-P. ; FEIXAS, M. ; SBERT, M. ; LLOBET, A.: Realtime automatic selection of good molecular views. In: *Computers and Graphics* 30, 2006, S. 98–110 (Zitiert auf Seite 58)
- [VÁZQUEZ & ANDÚJAR 2007] VÁZQUEZ, P.-P. ; ANDÚJAR, C.: Tessellation-Independent Best View Selection. In: *Proc. of International Conference on Computer Graphics and Artificial Intelligence*, 2007 (Zitiert auf Seite 58)
- [VÁZQUEZ u. a. 2008] VÁZQUEZ, P.-P. ; MONCLUS, E. ; NAVAZO, I.: Representative Views and Paths for Volume Models. In: *Proc. of Smart Graphics (SG)*, 2008, S. 106–117 (Zitiert auf den Seiten 61 und 108)
- [VERNON & PECKHAM 2002] VERNON, T. ; PECKHAM, D.: The benefits of 3D modelling and animation in medical teaching. In: *Journal of Visual Communication in Medicine*, 25(4), 2002, S. 142–148 (Zitiert auf Seite 40)
- [VIOLA u. a. 2006] VIOLA, I. ; FEIXAS, M. ; SBERT, M. ; GRÖLLER, M. E.: Importance-Driven Focus of Attention. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 12(5), 2006, S. 933–940 (Zitiert auf den Seiten 60 und 91)
- [VOLL & WESKER 2009] VOLL, M. ; WESKER, K. ; SCHÜNKE, M. (Hrsg.) ; SCHULTE, E. (Hrsg.) ; SCHUMACHER, U. (Hrsg.): *Prometheus Innere Organe: LernAtlas der Anatomie*. Thieme, 2009 (Zitiert auf Seite 121)
- [VOLLIK u. a. 2007] VOLLIK, I. ; VOGEL, D. ; AGRAWALA, M. ; HERTZMANN, A.: Specifying Label Layout Styles by Example. In: *Proc. of ACM Symposium on User Interface Software and Technology (UIST)*, 2007, S. 221–230 (Zitiert auf den Seiten 126 und 158)
- [WALTER u. a. 2008] WALTER, J. ; BURDELSKI, M. ; BRÖRING, D. C.: Chancen und Risiken der Leber-Lebendspende-Transplantation. In: *Deutsches Ärzteblatt*, 105(6), 2008, S. 101–107 (Zitiert auf Seite 14)
- [WANG & LEE 2008] WANG, Y.-S. ; LEE, T.-Y.: Example-driven animation synthesis. In: *Visual Computer*, 24(7), 2008, S. 765–773 (Zitiert auf Seite 158)

- [WARE & OSBORNE 1990] WARE, C. ; OSBORNE, S.: Exploration and virtual camera control in virtual three dimensional environments. In: *Proc. of ACM SIGGRAPH Computer Graphics*, 24(2), 1990, S. 175–183 (Zitiert auf Seite 54)
- [WEB3D CONSORTIUM 2003] WEB3D CONSORTIUM: *ISO/IEC 14772-1:1997 and ISO/IEC 14772-2:2004 – Virtual Reality Modeling Language (VRML)*, 2003 (Zitiert auf Seite 246)
- [WEBOP GMBH 2009] WEBOP GMBH: *webop - All about surgical operations*. 2009. – <http://www.webop.de>. – besucht am 12.12.2009 (Zitiert auf Seite 38)
- [WEINSTEIN u. a. 2005] WEINSTEIN, D. ; PARKER, S. ; SIMPSON, J. ; ZIMMERMAN, K. ; JONES, G.: Visualization in the SCIRun Problem-Solving Environment. In: *The Visualization Handbook*. Elsevier, 2005, S. 615–632 (Zitiert auf den Seiten 187, 189, 190 und 207)
- [WELP u. a. 2006] WELP, C. ; SIEBERS, S. ; ERMERT, H. ; WERNER, J.: Investigation of the influence of blood flow rate on large vessel cooling in hepatic radiofrequency ablation. In: *Biomedizinische Technik*, 51(5/6), 2006, S. 337–346 (Zitiert auf den Seiten 13 und 223)
- [WESTERMARK u. a. 2005] WESTERMARK, A. ; ZACHOW, S. ; EPPLEY, B.: 3-D osteotomy planning in maxillofacial surgery, including soft tissue prediction. In: *Journal of Craniofacial Surgery*, 16(1), 2005, S. 100–104 (Zitiert auf Seite 20)
- [VAN WIJK & NUIJ 2003] WIJK, J. J. van ; NUIJ, W. A. A.: Smooth and efficient zooming and panning. In: *Proc. of IEEE Information Visualization (InfoVis)*, 2003, S. 15–22 (Zitiert auf Seite 156)
- [VAN WIJK & NUIJ 2004] WIJK, J. J. van ; NUIJ, W. A.: A Model for Smooth Viewing and Navigation of Large 2D Information Spaces. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 10(4), 2004, S. 447–458 (Zitiert auf den Seiten 109 und 248)
- [WOHLFART & HAUSER 2007] WOHLFART, M. ; HAUSER, H.: Story Telling for Presentation in Volume Visualization. In: *Proc. of IEEE/Eurographics Symposium on Visualization (EuroVis)*, 2007, S. 91–98 (Zitiert auf den Seiten 40, 150, 151 und 158)
- [WOLF u. a. 2005] WOLF, I. ; VETTER, M. ; WEGNER, I. ; BÖTTGER, T. ; NOLDEN, M. ; SCHÖBINGER, M. ; HASTENTEUFEL, M. ; KUNERT, T. ; MEINZER, H.-P.: The Medical Imaging Interaction Toolkit. In: *Medical Image Analysis*, 9, 2005, S. 594–604 (Zitiert auf den Seiten 181, 186 und 190)
- [YAMANAKA u. a. 2007] YAMANAKA, J. ; SAITO, S. ; FUJIMOTO, J.: Impact of Preoperative Planning Using Virtual Segmental Volumetry on Liver Resection for Hepatocellular Carcinoma. In: *World Journal of Surgery*, 31(6), 2007, S. 1251–1257 (Zitiert auf Seite 17)
- [YODER u. a. 1998] YODER, J. W. ; SCHULTZ, D. F. ; WILLIAMS, B. T.: The MEDI-GATE graphical user interface for entry of physical findings: Design principles

LITERATURVERZEICHNIS

and implementation. In: *Journal of Medical Systems*, 22(5), 1998, S. 325–337
(Zitiert auf Seite [230](#))

[ZHANG u. a. 2008] ZHANG, J. ; LIN, Y. ; WANG, C. ; LIU, Z. ; YANG, Q.: Criteria of human-computer interface design for computer assisted surgery systems. In: *Journal of Shanghai Jiaotong University (Science)*, 13(5), 2008, S. 538–541 (Zitiert auf Seite [231](#))

[ZUMBACH u. a. 2000] ZUMBACH, J. ; MEHRABI, A. ; SCHWARZER, H. ; RENTZ, C. ; HERFARTH, P. R. C. ; KALLINOWSKI, F.: Wie beurteilen Studierende CBT-Module? Evaluation von Trainingsprogrammen in der Chirurgie. In: *Proc. of CBT in der Medizin*, 2000, S. 113–125 (Zitiert auf Seite [34](#))