

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik



Diplomarbeit

Entwicklung eines Verfahrens zur
parametrisierbaren Kamerapositionierung
in medizinischen Visualisierungen

Mathias Neugebauer

Institut für Simulation und Graphik

**Entwicklung eines Verfahrens zur
parametrisierbaren Kamerapositionierung
in medizinischen Visualisierungen**

Diplomarbeit

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

von: MATHIAS NEUGEBAUER
geb. am: 30. Januar 1982
in: Wippra
Matrikelnummer: 16 25 49

1. Gutachter: Prof. Dr.-Ing. BERNHARD PREIM
2. Gutachter: Dr.-Ing. KNUT HARTMANN

Betreuer: Prof. Dr.-Ing. BERNHARD PREIM
Dipl.-Inf. KONRAD MÜHLER
Dipl.-Ing. CHRISTIAN TIETJEN

Zeit der Diplomarbeit: 12.06.2006 - 24.12.2006

Selbstständigkeitserklärung

Hiermit versichere ich, Mathias Neugebauer (Matrikel-Nr. 162549), die vorliegende Arbeit allein und nur unter Verwendung der angegebenen Quellen angefertigt zu haben.

Mathias Neugebauer

Magdeburg, 22.12.2006

Danksagung

In erster Linie möchte ich meinen Betreuern Prof. Dr. Bernhard Preim, Dipl.- Ing. Christian Tietjen und Dipl.-Inf. Konrad Mühler danken. Sie hatten immer ein offenes Ohr für meine Probleme und (häufig leicht überzogenen) Sorgen. Sie halfen mir dabei, MEVISLAB zu bezwingen und korrigierten unermüdlich so manche verworrene Textzeile. Ich danke Manuela Kuhn, die sich ebenfalls dieser Aufgabe gestellt hat.

Besonders danke ich meiner Freundin Karin. Nicht nur weil sie einen kontrollierenden Blick auf die vielen Seiten dieser Arbeit geworfen hat, sondern auch weil sie immer noch meine Freundin ist, obwohl ich in den letzten Wochen kaum Zeit für sie hatte. Statt sich zu beklagen hat sie mich aufgemuntert und mir Tee gekocht. Danke.

Ich danke meinen MitdiplomandInnen, weil sie alle so nett waren und es Spaß gemacht hat, ins Labor zu kommen. Besonders sei mein Labornachbar Rocco Gasteiger erwähnt, der sich nicht umgesetzt hat, obwohl ich ihn immer wieder mit Gesprächen, Fragen, Tee- und Kakaopausen von der Arbeit abgehalten habe.

Meinen Eltern danke ich dafür, dass sie mir dieses Studium ermöglicht und mich immer unterstützt haben. Dank ihnen kann ich heute diese Danksagung schreiben. Und *last but not least* danke ich meinem Herrn Jesus Christus, der immer an meiner Seite war und mir geholfen hat.

Mathias Neugebauer, 22. Dezember 2006

Inhaltsverzeichnis

- 1 Einleitung** **1**
 - 1.1 Zielsetzung 2
 - 1.2 Gliederung 2

- 2 Grundlagen und verwandte Arbeiten** **5**
 - 2.1 Medizinische Volumenvisualisierung 5
 - 2.1.1 Gewinnung der Daten 6
 - 2.1.2 Visualisierungsverfahren 6
 - 2.1.3 Anwendungsbeispiele 8
 - 2.1.4 Zusammenfassung 11
 - 2.2 Interaktive Exploration medizinischer Visualisierungen 12
 - 2.2.1 Zielsetzung von Visualisierungen 12
 - 2.2.2 Statische Informationsvermittlung 13
 - 2.2.3 Probleme interaktiver Exploration 13
 - 2.2.4 Techniken zur Hervorhebung und Kontextverdeutlichung 16
 - 2.2.5 Zusammenfassung 17
 - 2.3 Grundlagen zur virtuellen Kamera 18
 - 2.3.1 Kameramodell 18
 - 2.3.2 Projektionsarten 19
 - 2.3.3 Kamerasteuerung 20
 - 2.3.4 Zusammenfassung 22
 - 2.4 Verfahren zur Kamerapositionierung 24
 - 2.4.1 Sichtbarkeitsbestimmung 24
 - 2.4.2 Viewpoint Quality 32
 - 2.4.3 Systeme zur Kameraplanung 35
 - 2.5 Zusammenfassung 38

- 3 Automatische Kamerapositionierung** **41**
 - 3.1 Anforderungsanalyse 41
 - 3.1.1 Analyse der Szenen 42
 - 3.1.2 Analyse der Anwendung 46
 - 3.1.3 Zusammenfassung und Schlussfolgerung 47
 - 3.2 Generierung der Datenbasis 50

3.2.1	Viewport-Auflösung	50
3.2.2	Verteilung der Kamerapositionen	51
3.2.3	Sichtbarkeitsanalyse	53
3.2.4	Skalierbarkeit und Aufwandsbetrachtung	57
3.3	Berechnung der Kamerapositionen	59
3.3.1	Parameter	61
3.3.2	Ergebnisbildung und weiterführende Anwendungen	71
3.4	Zusammenfassung	76
4	Implementierung	79
4.1	Werkzeuge zur Implementierung	79
4.1.1	Open Inventor	79
4.1.2	MeVisLab	80
4.1.3	METK	81
4.2	Umsetzung der automatischen Kamerapositionierung	82
4.2.1	Generierung der Datenbasis	82
4.2.2	Diskussion der Laufzeit	90
4.2.3	Berechnung der Kamerapositionen	92
4.3	Zusammenfassung	98
5	Ergebnisse	99
5.1	Anwendungsbeispiele	99
5.1.1	Hals-Visualisierung	99
5.1.2	Knie-Visualisierung	100
5.1.3	Leber-Visualisierung	101
5.1.4	Zusammenfassung	101
5.2	Evaluierung	102
5.2.1	Fragebogen	103
5.2.2	Ergebnisse und Interpretation	104
6	Zusammenfassung und Ausblick	109
6.1	Weiterführende Ansätze und Verbesserungsvorschläge	110
	Abbildungsverzeichnis	113
	Literaturverzeichnis	117

1 Einleitung

Im Bereich der modernen Diagnostik kommen immer häufiger komplexe technische Systeme zum Einsatz, um den untersuchenden Mediziner (Chirurgen, Orthopäden, Strahlentherapeuten etc.) mit allen notwendigen Informationen zu versorgen. So konnte sich beispielsweise die computergestützte Operationsplanung auf der Basis dreidimensionaler Modelle der betroffenen Körperregionen in den letzten Jahren weitgehend etablieren.

Hierbei kann der Chirurg durch dreidimensionale Visualisierungen unterstützt werden, um so unnötige Operationen zu vermeiden und im Fall einer Operation das Risiko für den Patienten zu reduzieren. Damit eine Visualisierung einen solch unterstützenden Effekt hat, muss es dem Chirurgen möglich sein, die ihm präsentierten Darstellungen korrekt zu interpretieren. Durch verschiedenste Techniken werden die Visualisierungen aufgewertet, sei es um die Aufmerksamkeit des Chirurgen auf wichtige Strukturen zu lenken oder auch um zusätzliche, nicht direkt aus dem anatomischen Aufbau extrahierbare Informationen zu vermitteln.

Die virtuelle Kamera stellt dabei die Schnittstelle zwischen dem Chirurgen und dem Modell dar. Ihre Orientierung und Position entscheidet grundlegend, welche Strukturen für ihn sichtbar sind und welche Lage sie zueinander einnehmen. In bisherigen Planungssystemen gab es, wenn überhaupt, nur sehr rudimentäre Ansätze um den Arzt bei der oft sehr komplexen Aufgabe der Kamerasteuerung zu unterstützen. Auch wenn eine dreidimensionale Visualisierung sehr informativ gestaltet ist, kann der Chirurg diese Informationen nur bei einer entsprechend günstig gewählten Kameraposition korrekt interpretieren.

Eine ungünstig gewählte Kameraposition kann sehr negative Folgen haben. Um eine bestimmte Struktur auch von ihr aus sichtbar zu machen, müssen eventuell viele verdeckende Strukturen in ihrer Darstellung abgeschwächt oder sogar vollständig ausgeblendet werden. Somit sinkt der Informationsgehalt und für die Entscheidungsfindung wichtige Kontextinformationen werden zerstört oder die Kamera ist so positioniert, dass sich die Strukturen aus einem sehr ungewöhnlichen Blickwinkel präsentieren. Dann kann der Fall eintreten, dass sie trotz des Einsatzes von Hervorhebungstechniken übersehen oder ihre Größe, Position und die Abstände zu anderen Strukturen falsch eingeschätzt werden.

Der Prozess der manuellen Exploration einer Visualisierung kann erleichtert werden, indem eine günstige, automatisch berechnete Startposition angeboten wird. In diesem

Fall sind oft nur noch geringfügige Navigationsschritte notwendig, um sich einen ausreichenden Überblick über eine Struktur zu verschaffen. Solche automatisch generierten Positionen können auch in Form von Animationen zum Einsatz kommen, um den Chirurgen einen Überblick über das Anwendungsszenario zu liefern. Durch eine derartige Vorinformation wird es ihm erleichtert, sich in der Visualisierung zurechtzufinden.

1.1 Zielsetzung

Ziel dieser Diplomarbeit ist es, ein Verfahren zur automatischen, parametrisierbaren Kamerapositionierung in dreidimensionalen medizinischen Visualisierungen zu entwickeln. Die Aufgabenstellung unterteilt sich dabei in zwei große Teilbereiche: die Szenenanalyse und die parametrisierte Kamerapositionierung.

Die Entscheidung, ob eine Kameraposition geeignet ist, richtet sich unter anderem nach den Verhältnissen innerhalb der dreidimensionalen Visualisierung. Es müssen Sichtbarkeitsverhältnisse und gegenseitige Verdeckungen analysiert werden. Die Ergebnisse eines entsprechenden Analyseverfahrens sollten über ein breites Informationsspektrum verfügen. Nur so ist es möglich, auf der Basis dieser Informationen schnell und flexibel auf sich eventuell ändernde Gesichtspunkte, unter denen die Visualisierung exploriert werden soll, zu reagieren.

Damit entschieden werden kann, wie geeignet eine Kameraposition ist, wird ein Bewertungsmaß konzipiert. Dieses basiert nur zum Teil auf aus der Szenengeometrie ableitbaren Informationen. Eine Analyse von Anwendungsszenarien ist notwendig, um entscheidende Parameter zu finden, an denen sich die Güte einer Kameraposition orientiert. Diese Parameter müssen entsprechend repräsentiert werden, um sie für die Ergebnisfindung und letztendliche Kamerapositionierung nutzbar zu machen.

Beide Verfahren, die Szenenanalyse und die automatische Kamerapositionierung, werden auf der Basis MEVISLAB umgesetzt. Dadurch wird die Anbindung an das ebenfalls in MEVISLAB realisierte *Medical Exploration Toolkit* (METK) begünstigt.

1.2 Gliederung

Die vorliegende Arbeit ist folgendermaßen gegliedert:

Kapitel 2 – Für die Positionierung der Kamera in einer dreidimensionalen Visualisierung ist entscheidend auf welche Weise die Visualisierung präsentiert wird. Demzufolge wird ein Einblick in die Grundlagen der medizinischen Volumenvisualisierung geliefert. Darauf aufbauend werden Probleme der interaktiven Exploration von Visualisierungen erörtert und Lösungsstrategien vorgestellt. Des Weiteren werden Grundlagen über Abbildungsmodelle und Steuerungsmetaphern der virtuellen

Kamera vermittelt. Verfahren zur Sichtbarkeitsanalyse werden klassifiziert und vorgestellt. Abschließend wird die Problematik der Positionsbewertung erörtert und Systeme zur automatischen Kamerapositionierung ebenfalls klassifiziert.

Kapitel 3 – Anhand typischer Anwendungsszenarien werden die Anforderungen an Verfahren zur Szenenanalyse und automatischen Kamerapositionierung analysiert und beschrieben. Darauf folgend wird ein bildbasiertes Verfahren zur Sichtbarkeitsanalyse vorgestellt. Die aus diesem Verfahren gewonnenen Informationen werden für die automatische Kamerapositionierung genutzt. Dieses Positionierungsverfahren und alle im Rahmen dieser Arbeit ermittelten Parameter zur Kamerabewertung sowie weiterführende Modifikationen des Verfahrens werden im darauf folgenden Abschnitt erläutert.

Kapitel 4 – In diesem Kapitel wird die Realisierung beider Verfahren auf der Basis von C++-Modulen, welche in MEVISLAB zum Einsatz kommen, erläutert. Es wird auf den algorithmischen Ablauf und Datenstrukturen eingegangen. Laufzeiten werden vorgestellt und sich daraus ergebende Schlussfolgerungen diskutiert.

Kapitel 5 – Die Ergebnisse in Form von automatisch generierten Kamerapositionen werden untersucht. Dabei wird exemplarisch auf typische Problemstellungen im Bereich der medizinischen Visualisierung eingegangen und der Einfluss der verschiedenen Parameter diskutiert. Des Weiteren werden die Ergebnisse einer auf der Basis dieser Arbeit durchgeführten Evaluierung präsentiert und ausgewertet.

Kapitel 6 – Abschließend werden noch offene Problemstellungen erörtert und Vorschläge für mögliche, weiterführende Projekte präsentiert.

2 Grundlagen und verwandte Arbeiten

Wie eine Visualisierung dargestellt wird, hat Einfluss darauf, welche Techniken bei der automatischen Positionierung der Kamera zum Einsatz kommen können. Deshalb wird auf die Thematik der medizinischen Visualisierung näher eingegangen, wobei speziell die Datenakquisition, die gängigsten Visualisierungsverfahren und Anwendungsbereiche erörtert werden.

Ziel von Visualisierungen ist es, Informationen zu vermitteln. Da diese im medizinischen Bereich häufig sehr komplex gestaltet sein können, wurden verschiedene Techniken entwickelt, um die Aufmerksamkeit des Betrachters zu steuern. Die automatische Auswahl günstiger Kamerapositionen wird dem Bereich dieser Hervorhebungstechniken zugeordnet. Neben einer näheren Betrachtung der auftretenden Probleme bei der interaktiven Exploration medizinischer Visualisierungen werden übliche Hervorhebungstechniken vorgestellt und klassifiziert, um so den Inhalt dieser Arbeit in den Kontext einordnen zu können.

Eine wichtige Voraussetzung für die automatische Positionierung der Kamera ist die Analyse der Sichtbarkeitsverhältnisse. Nur wenn eine Struktur sichtbar ist, können effektiv Informationen über sie gewonnen werden. Des Weiteren müssen Aussagen über den Informationsgehalt einer Struktur bezüglich einer Betrachtungsrichtung getroffen werden. Auch die Modellierung der virtuellen Kamera ist entscheidend, da die Kameraparameter bestimmen, auf welche Weise dem Nutzer die medizinische Visualisierung präsentiert wird. Neben diesen Parametern, den verschiedenen Arten der Analyse von Sichtbarkeitsverhältnissen in dreidimensionalen Szenen und Ansätzen zur Bewertung von Kamerapositionen werden im dritten Abschnitt dieses Kapitels Systeme zur Kameraplanung vorgestellt.

2.1 Medizinische Volumenvisualisierung

Bei der medizinischen Volumenvisualisierung werden Volumendatensätze genutzt, um dreidimensionale Darstellungen anatomischer Strukturen zu erzeugen. Diese Datensätze sind das Resultat verschiedener bildgebender Verfahren. Im Folgenden wird der Weg von der Gewinnung der Daten bis zu ihrer Darstellung mithilfe verschiedener Visualisierungsverfahren kurz umrissen. Des Weiteren wird ein Einblick in mögliche Anwendungsbereiche für medizinische Volumenvisualisierungen gegeben.

2.1.1 Gewinnung der Daten

Für die Volumenvisualisierungen sind vor allem tomographische Aufnahmeverfahren von Bedeutung. Sie liefern, im Gegensatz zu Verfahren wie dem konventionellen Röntgen, überlagerungsfreies Datenmaterial in Form von transversalen Schnittbildern der entsprechenden Körperregion. Der Volumendatensatz stellt sich als Summe aller Schnittbilder dar. Die tomographischen Aufnahmeverfahren unterscheiden sich hauptsächlich in der Art des Signals, welches zur Informationsgewinnung verwendet wird. Die zwei wichtigsten Verfahren sind die Computertomographie und die Magnetresonanztomographie.

Computertomographie (CT). Wie beim konventionellen Röntgenverfahren wird das Absorptionsverhalten der verschiedenen anatomischen Strukturen zur Abbildung und Kontrastierung genutzt. Dies führt dazu, dass sehr dichte und somit stark absorbierende Strukturen wie Knochen sich durch einen hohen Abbildungscontrast auszeichnen (siehe Abb. 2.1(a)) [MORNEBORG, 1995]. Aufgrund der Normierung (Hounsfield-Skala) hat ein bestimmter Gewebetyp in jedem Datensatz denselben Intensitätsbereich, was eine Klassifizierung der enthaltenen anatomischen Strukturen und somit nachfolgende bildverarbeitende Prozesse begünstigt.

Magnetresonanztomographie (MRT). Die Bilderzeugung bei der MRT basiert auf der Orientierungsänderung der Drehmomente (*Spins*) von Atomkernen mit ungerader Protonen- und Neutronenanzahl unter dem Einfluss von Magnetfeldern. Der häufig im menschlichen Organismus vorkommende Wasserstoff besitzt die entsprechenden Eigenschaften. Aufgrund der guten Kontrasteigenschaften bei Weichteilgewebe wird die MRT zur Abbildung von anatomischen Strukturen wie Organen, Gelenknorpeln oder dem Gehirn genutzt (siehe Abb. 2.1(b)). Die Intensitätswerte im Ergebnisdatensatz sind nicht normiert und können starke Inhomogenitäten und Artefakte aufweisen. Dies wirkt sich nachteilig auf nachfolgende Bildverarbeitungsprozesse aus.

2.1.2 Visualisierungsverfahren

Bei den aus den tomographischen Aufnahmeverfahren gewonnenen Daten handelt es sich um schichtweise aufgebaute Volumendaten. Die einfachste Form der Darstellung ist die Präsentation der einzelnen Schichten. Der Vorteil ist, dass die Originaldaten präsentiert werden und die Informationen sehr genau und detailliert vorliegen. Bei dieser 2D-Visualisierung fällt jedoch das Erkennen von Lagebeziehungen und Größenverhältnissen schwer. Eine dreidimensionale Darstellung der Daten erleichtert die Analyse und verringert das Risiko von Fehleinschätzungen. Verfahren, die eine solche 3D-Visualisierung ermöglichen, unterteilen sich in direkte und indirekte Visualisierungsverfahren.

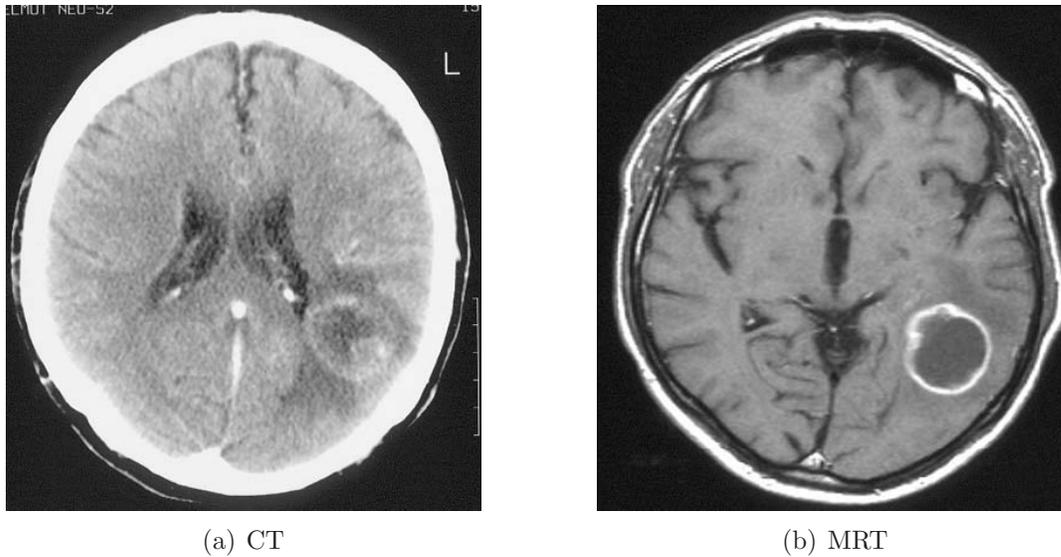


Abb. 2.1: Die Ergebnisse von CT und MRT unterscheiden sich in ihrem Kontrastverhalten: CT bildet die Schädelstruktur kontrastreich ab, MRT hebt die feinen Gehirnstrukturen hervor. [Quelle: *Universitätsklinikum Carl Gustav Carus Dresden, Abteilung für Neuroradiologie*]

Häufig geht der Visualisierung eine Segmentierung der Daten voraus, um bestimmte im Datensatz enthaltene Strukturen zu extrahieren und gesondert darzustellen. Damit es zu keiner Informationsverfälschung kommt, ist vor allem in medizinischen Anwendungen der Toleranzbereich für die Genauigkeit der Segmentierung entsprechend gering. Abhängig von der Art der zu extrahierenden Struktur werden verschiedene, teilweise interaktive Verfahren verwendet, um eine entsprechend genaue Segmentierung zu erhalten [HINTZE et al., 2005].

Direkte Volumenvisualisierung (DV)

Bei dieser Art der Visualisierung werden die einzelnen Voxel des Datensatzes genutzt, um eine dreidimensionale Darstellung zu erzeugen. Da keine Vorverarbeitungsschritte durchgeführt werden müssen, kann die Visualisierung direkt aus den im Datensatz enthaltenen Informationen erzeugt werden.

Abhängig von der Größe des Datensatzes sind oft sehr viele Voxel an der Darstellung beteiligt. Des Weiteren ist zum Erreichen einer guten Abbildungsqualität eine, meist trilineare, Interpolation zwischen den auf diskreten Voxelpositionen vorhandenen Informationen notwendig [LEVOY, 1988]. Dies hat einen hohen Speicher- und Rechenaufwand zur Folge. Um eine interaktive Darstellung zu ermöglichen, wurden verschiedene optimierte Verfahren entwickelt, wie die *Shear Warp* Transformation [LACROUTE und LEVOY, 1994], welche auf Grund ihrer Struktur parallelisierbar ist und so hardwareseitig noch

weiter beschleunigt werden kann [SCHULZE und LANG, 2002]. Neben hochspezialisierter Hardware, z.B. Komponenten der Firma TERARECON [TERARECON, 2006], ermöglichen mittlerweile fast alle aktuellen Grafikkarten den Einsatz sogenannter 3D-Texturen, mit deren Hilfe eine hardwarebeschleunigte direkte Volumenvisualisierung ebenfalls möglich ist.

Um das Kontrastverhalten bei der Darstellung des Volumendatensatzes zu steuern, werden Transferfunktionen (TF) genutzt. Diese übertragen die skalaren Werte der Voxel auf die bei deren Darstellung verwendeten Farb- bzw. Transparenzinformationen. Zum Erreichen aussagekräftiger Visualisierungen sind mehrdimensionale TFs notwendig, die meist einen hohen Interaktionsaufwand erfordern [TAPPENBECK et al., 2005].

Indirekte Volumenvisualisierung (IDV)

Bei der indirekten Volumenvisualisierung werden die Oberflächen der im Datensatz enthaltenen anatomischen Strukturen extrahiert und dargestellt. Die meisten Extraktionsverfahren basieren auf dem *Marching Cubes* Algorithmus [LORENSEN und CLINE, 1987]. Die extrahierte Oberfläche orientiert sich an einem vorher definierten Grenzwert. Voxel mit skalaren Werten unter diesem Grenzwert befinden sich außerhalb, Voxel mit höheren skalaren Werten innerhalb der von der Oberfläche eingeschlossenen Region.

Das Ergebnis ist ein polygonales Modell der Oberfläche. Dies ermöglicht auch auf nicht spezialisierter Grafikhardware eine effektive Visualisierung. Des Weiteren können Beleuchtungsmodelle, Texturen oder Shader-Effekte einfach implementiert werden, um so die Qualität der Darstellung zu steigern.

Die Beschleunigung der Darstellung wird durch die Überführung in eine Zwischenrepräsentation geringerer Dimension erreicht. Somit sinkt der Informationsgehalt der Visualisierung. Die Exaktheit der polygonalen Darstellung kann unter Umständen zu Fehleinschätzungen führen, da sie das Ergebnis des Extraktionsverfahrens ist und nicht zwangsläufig auf den im Datensatz vorliegenden Verhältnissen beruht.

Durch die Verwendung hybrider Visualisierungsverfahren können die Vorzüge der direkten und indirekten Volumenvisualisierung in einer Darstellung kombiniert werden [SEEMANN et al., 2000].

2.1.3 Anwendungsbeispiele

Die mit Hilfe bildgebender Verfahren gesammelten und durch entsprechende Visualisierungsverfahren dreidimensional dargestellten Daten können in verschiedensten Anwendungsbereichen zum Einsatz kommen. Im medizinischen Bereich gibt es eine Vielzahl

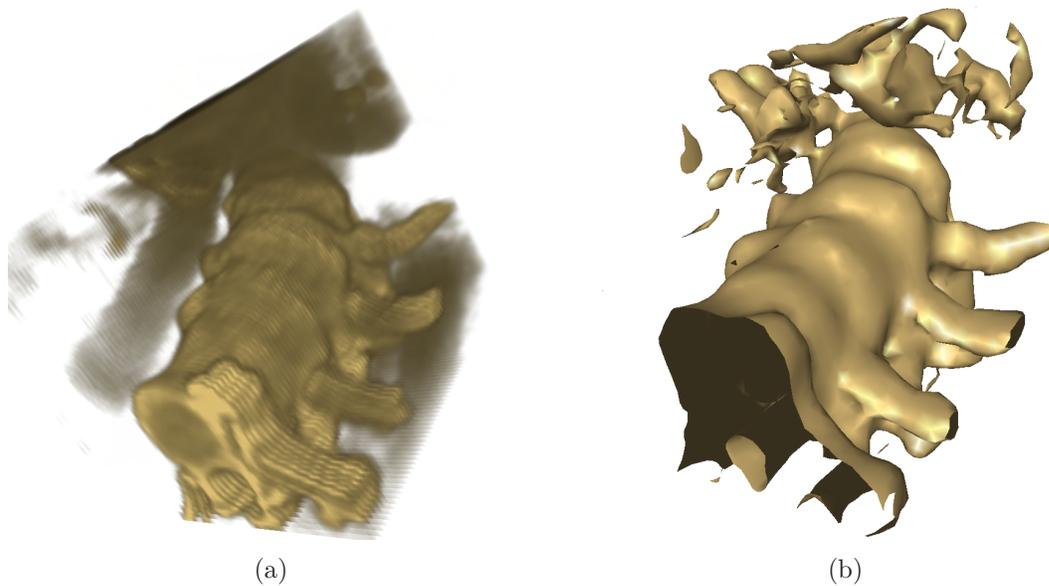


Abb. 2.2: Gegenüberstellung von direkter (a) und indirekter (b) Volumenvisualisierung: die Oberfläche bei der IDV (*Marching Cubes*) ist exakt abgegrenzt. Die Darstellung der DV (3D-Texturen) wirkt aufgrund der durch die TF definierten Teiltransparenz amorph, einzelne Bildschichten sind erkennbar.

von Anwendungsmöglichkeiten für dreidimensionale Visualisierungen. Von einer automatischen Vorauswahl günstiger Kamerapositionen profitieren vor allem die computergestützte Lehre und das Training sowie die präoperative Planung.

Medizinische Lehre und Training

Der Einsatz computergestützter 3D-Visualisierungen ermöglicht es, medizinische Fachkräfte anhand interaktiver Modelle, in Kombination mit verschiedensten Medien, aus- bzw. weiterzubilden. Im Folgenden werden Beispiele für solche Lehr- bzw. Trainingssysteme präsentiert.

Der VOXEL-MAN von HÖHNE et al. [2003] ermöglicht die Exploration des VISIBLE HUMAN (siehe Abb. 2.3(b)). Es ist möglich, interaktiv einzelne Organe auszublenden oder Schnittansichten bestimmter Körperregionen zu erzeugen. Das Modell kann auch stereoskopisch dargestellt werden, um mithilfe einer 3D-Brille räumliche Zusammenhänge besser zu erfassen. Des Weiteren können radiologische Abbildungsverfahren simuliert werden, um so z. B. die Analyse von Röntgenbildern zu trainieren.

Beim VOXEL MAN wird eine sehr originalgetreue Darstellung erreicht. Dies wird zum einen durch den hybriden Einsatz von DV und IDV, zum anderen durch den hohen Detailgrad des zugrunde liegenden Datensatzes ermöglicht. Dieser stammt aus dem

VISIBLE HUMAN-Projekt [TIEDE et al., 1996]. Dabei wurde eine menschliche Leiche einem CT-Scan unterzogen. Die hohe Genauigkeit konnte nur durch eine sehr hohe Strahlenbelastung und das Wegfallen von Bewegungsartefakten erreicht werden.

Beim LIVER SURGERY TRAINER handelt es sich um ein fallbasiertes Trainingssystem für die Behandlung von Lebertumoren [BADE et al., 2006]. Auch nicht interaktive, geführte Präsentationen sind möglich. Die dabei notwendige Vorauswahl günstiger Kamerapositionen ist Thema dieser Arbeit. Die 3D-Modelle werden mittels IDV visualisiert. Da im Gegensatz zum VISIBLE HUMAN-Projekt die Daten von lebenden Patienten stammen, besitzen sie meist eine geringere Auflösung. Die Darstellung mittels IDV besitzt aufgrund der Interpolation bei der Erstellung der Oberflächen und dem Einsatz von Beleuchtungsmodellen eine höhere optische Qualität (siehe Abb. 2.3(a)).

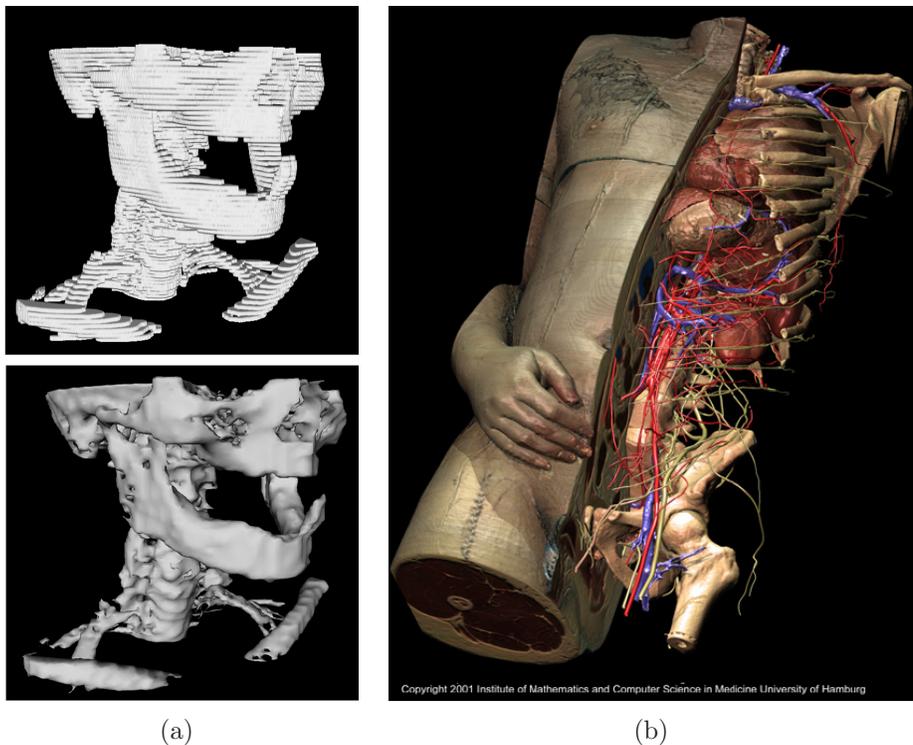


Abb. 2.3: Der relativ niedrig aufgelöste Datensatz in (a) gewinnt durch indirekte Volumenvisualisierung an optischer Qualität. Im Vergleich dazu hat der Datensatz des VISIBLE HUMAN-Projektes (b) eine sehr hohe Auflösung, wodurch eine ansprechende Präsentation mittels DV möglich ist (kleinere Objekte wie Venen oder Aterien werden durch IDV visualisiert). Quelle (b): HÖHNE et al. [2003]

Operationsplanung

Vor allem bei komplexen Eingriffen ist eine exakte Planung der Operation nötig, um deren Erfolg präoperativ abschätzen zu können und um die Belastung und das Verletzungsrisiko für den Patienten so gering wie möglich zu halten. Durch den Einsatz computergestützter Modelle kann eine solche Planung effektiv durchgeführt werden.

Ein spezielles Beispiel dafür ist der NECK SURGERY PLANNER [TIETJEN et al., 2005a]. Die mit dieser Software planbare Halslymphknotenausräumung wird durchgeführt, wenn Patienten bösartige Tumore im Kopf- und Halsbereich entwickeln. Dies ist notwendig, da sich an den Lymphknoten dieser Patienten häufig Metastasen bilden.

Die Visualisierung der Halsregion bildet beim NECK SURGERY PLANNER die Grundlage für die Planung dieses Eingriffs. Sie ist so gestaltet, dass alle nötigen Informationen möglichst schnell und effektiv erfasst werden können. Dazu gehört die Hervorhebung der Zielstrukturen bezüglich Form und Größe ebenso wie die Darstellung von Abstands- bzw. Durchdringungsinformationen. *Cut-Aways* und NPR-Techniken, wie das Silhouetten-Rendering, dienen dazu, den Fokus in dem sehr komplexen Halsdatensatz auf die Zielstrukturen zu lenken und so die Informationsgewinnung zu vereinfachen bzw. überhaupt zu ermöglichen (siehe Abb. 2.4). Die im Rahmen dieser Arbeit entwickelte automatische Kamerapositionierung dient dabei als unterstützende Technik. Die Darstellung erfolgt in Form von IDV. Der Informationsgehalt einer oberflächenbasierten Repräsentation ist ausreichend. Aufgrund der IDV kann die Grafikhardware effektiv für die Darstellung der diversen Hervorhebungstechniken verwendet werden.

2.1.4 Zusammenfassung

Medizinische Visualisierungen kommen in verschiedenen Anwendungsbereichen zum Einsatz. Hochaufgelöste, qualitativ hochwertige Datensätze und entsprechend gewählte Extraktionsverfahren sind essentiell, damit die resultierenden Visualisierungen von Nutzen sind. Visualisierungen auf der Basis ungenauer Informationen können zu Fehleinschätzungen führen, was vor allem im medizinischen Bereich nicht tolerierbar ist.

Aber auch wenn die Visualisierung alle notwendigen Informationen korrekt bereitstellt, ist es aufgrund ihrer meist komplexen Struktur schwierig, die richtigen Informationen zu ermitteln. Auftretende Probleme und Lösungsstrategien werden im nächsten Abschnitt diskutiert.

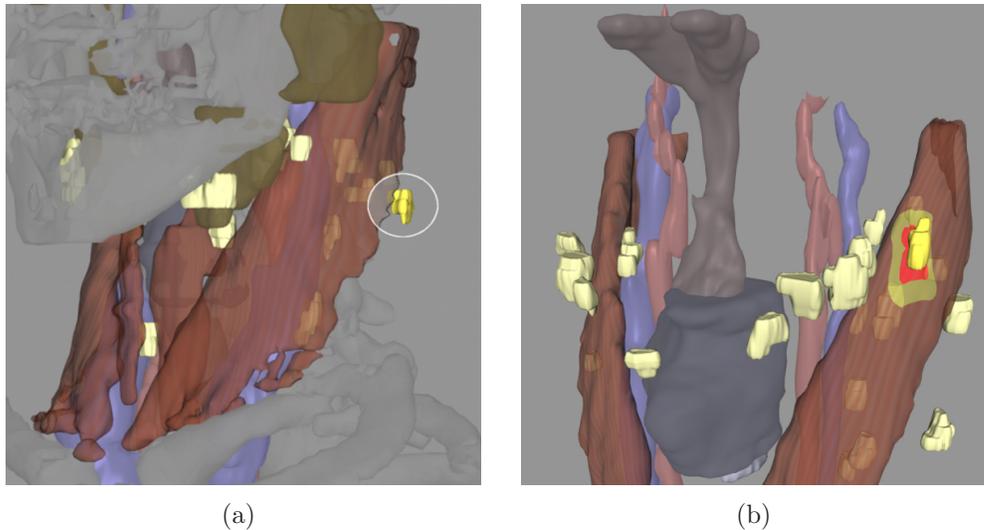


Abb. 2.4: Visualisierung im NECK DISSECTION PLANNER: (a) zeigt die Anwendung eines *Cut-Aways*, um einen Lymphknoten hervorzuheben. Gleichzeitig wird die Silhouette dargestellt, um räumliche Informationen zu wahren. In (b) ist die strukturierte Textur des Muskels zu erkennen, welche sich an der Gewebefaser orientiert. Als zusätzliche Information wurden Abstände zwischen Lymphknoten und Muskel farblich kodiert. Quelle: KRÜGER et al. [2005]

2.2 Interaktive Exploration medizinischer Visualisierungen

Bei der Darstellung medizinischer Zusammenhänge sind oft sehr spezifische Informationen von Interesse, wie z. B. die Morphologie oder die Lagebeziehung zwischen krankhaften und vitalen Strukturen. Bei Handzeichnungen in medizinischen Lehrbüchern und Atlanten kommen entsprechende Techniken zur Informationsverdeutlichung und Aufmerksamkeitssteuerung zum Einsatz. Bei der interaktiven Exploration computergenerierter Modelle birgt sowohl das Ziel einer möglichst umfangreichen Informationsvermittlung, als auch die interaktive Wahl der Betrachterposition neue Problemstellungen. Verschiedene Techniken zur Aufmerksamkeitssteuerung und Kontextverdeutlichung wurden entwickelt, um diesen Problemen entgegenzuwirken und den Informationsgehalt computergenerierter Visualisierungen zu steigern. Viele dieser Techniken kommen bei computergestützten Lehr- bzw. Planungssystemen zum Einsatz.

2.2.1 Zielsetzung von Visualisierungen

In der Medizin basieren die Darstellungen auf Patienteninformationen, welche durch verschiedene Verfahren gewonnen und entsprechend verarbeitet wurden. Die Informationen,

welche aus den Visualisierungen gewonnen werden sollen, umfassen unter anderem die Lagebeziehungen der dargestellten Strukturen, Abstands- und Größeninformationen und diverse statische oder auch nichtstatische Strukturparameter. Je nach Art der Visualisierung kann so eine Vielzahl von Informationen in ihr enthalten sein. Abhängig von der jeweiligen Anwendung müssen meist mehrere, unter Umständen verschiedene Informationstypen gleichzeitig vermittelt werden, um dem Betrachter schnell und effizient einen Überblick über deren Zusammenhänge zu geben.

Aufgrund dieser hohen Informationsdichte müssen verschiedene Techniken angewandt werden, um die Aufmerksamkeit entsprechend zu steuern.

2.2.2 Statische Informationsvermittlung

Abbildungen in medizinischen Lehrbüchern bieten Beispiele für grundlegende Mechanismen der Informationsverdeutlichung.

Neben apparativen Aufnahmen wie Fotografien oder Röntgenaufnahmen dienen vor allem Handzeichnungen dazu, anatomische Gegebenheiten bildhaft zu beschreiben. Durch die manuelle Anfertigung der Abbildungen können sie leicht an die zu vermittelnden Informationen angepasst werden. Unnötige Details werden leicht angedeutet oder gar nicht dargestellt (siehe Abb. 2.5(a)), wohingegen Fokusstrukturen durch entsprechende Farb- und Detailgestaltung in den Vordergrund rücken. Die Farbgebung orientiert sich häufig an der Originalfarbe der anatomischen Struktur oder entspricht definierten Normen. Die Abbildungen werden so gestaltet, dass die Strukturen dem Betrachter aus einer für ihn gewohnten Blickrichtung präsentiert werden. Auch hier kommen teilweise vordefinierte Blickrichtungen (*cranialis* - kopfwärts, *caudalis* - steißwärts, *dorsalis* - rückenwärts etc.) zum Einsatz. Um den Blick auf innere Strukturen zu ermöglichen, werden Schnitte dargestellt, verdeckende Strukturen nicht gezeichnet oder der Einsatz medizinischer Instrumente (z.B. Klammern) dargestellt, mit deren Hilfe der Blick in einer für den Mediziner gewohnten Art und Weise freigelegt wird (siehe Abb. 2.5(b)).

Durch all diese Techniken können viele Informationen präzise vermittelt werden. Allerdings sind diese Bilder statisch. Dadurch entfallen Probleme, welche sich bei einer interaktiven Exploration der Visualisierungen ergeben.

2.2.3 Probleme interaktiver Exploration

Bei der interaktiven Exploration werden dem Nutzer dreidimensionale Szenen präsentiert, durch die er im Rahmen der Informationsgewinnung navigiert. Die Navigation erfolgt durch die Manipulation einer virtuellen Kamera mithilfe eines entsprechenden Eingabegerätes. Dabei können sich die Steuerung der Kamera, die daraus resultierenden Verdeckungen und das mögliche Übersehen einzelner Strukturen als problematisch erweisen.

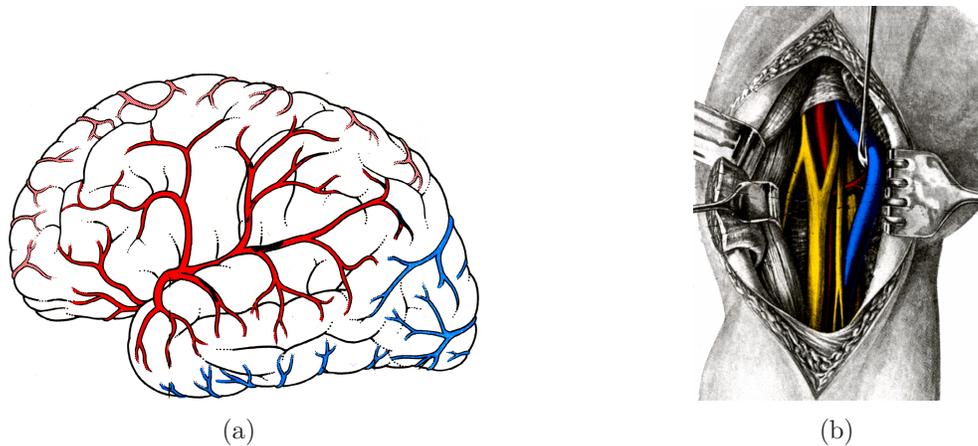


Abb. 2.5: Eine auf die essentiellen Inhalte beschränkte Darstellung der Arteriensysteme des Gehirns in (a) und die Darstellung des Gefäß- und Nervenstrangs am Seitenrand der Achselgegend (b). Der *M. pectoralis major* wurde durchtrennt. Des Weiteren sind medizinische Geräte wie Haken und Klammern abgebildet. Quellen: (a) ROGERS [1992], (b) LIPPERT [2000]

Eine Problemstellung der interaktiven Exploration ist die Steuerung der virtuellen Kamera. Durch die dreidimensionale Abbildung und demzufolge räumliche Navigation ergeben sich für die virtuelle Kamera sieben Freiheitsgrade, welche sich durch die Translation in allen drei Dimensionen, die Rotation und die Fokussierung darstellen [CHRISTIE und LANGUÉNOU, 2003].

Das meistverbreitete Eingabegerät zur Steuerung virtueller Kameras ist die Maus. Mit diesem Gerät ist eine direkte Navigation nur auf einer 2D-Ebene möglich. Eine Erweiterung auf zusätzliche Freiheitsgrade wird erreicht, indem die 2D-Bewegung der Maus durch nutzerseitige Auswahl auf die entsprechenden Freiheitsgrade angewendet wird. Eine synchrone Manipulation aller Freiheitsgrade ist auf diese Art und Weise nicht möglich und schon die auf drei Freiheitsgraden basierende Rotation erweist sich als komplexe Problemstellung [BADE et al., 2005].

Eine 3D-Navigation gestaltet sich somit kompliziert. Dies gilt vor allem für Nutzer, welche nicht alltäglich damit konfrontiert werden. Die Verwendung spezieller Eingabegeräte, wie die von HACHET et al. [2003] entwickelte *Control Action Table* (CAT) [HACHET et al., 2003] kann die Navigation einfacher gestalten (siehe Abb. 2.6). Aber auch softwareseitig kann durch die automatische Generierung von Ausgangspositionen die Maus-basierte Navigation unterstützt werden. Die Generierung solcher Positionen ist ein Bestandteil des im Rahmen dieser Arbeit entwickelten Verfahrens.

Neben den Schwierigkeiten bei der Nutzerinteraktion ergeben sich allerdings auch Probleme, welche auf der internen Struktur und Präsentation der Visualisierung beruhen. Im Optimalfall sollen interaktive 3D-Visualisierungen eine ebenso hohe Informationsdichte und Verständlichkeit wie die in Abschnitt 2.2.2 erwähnten Handzeichnungen besitzen.

Dazu müssen zwei Faktoren für die Steuerung der Aufmerksamkeit beachtet werden: die Hervorhebung der Fokusstrukturen bei gleichzeitig freier Wahl der Betrachterposition. Für die Hervorhebung gelten ähnliche Mechanismen wie für statische Abbildungen. Verschiedene Arten von Verfahren können dabei genutzt werden, um die Effekte der von den Künstlern angewandten Techniken nachzubilden. Aus der Möglichkeit zur freien

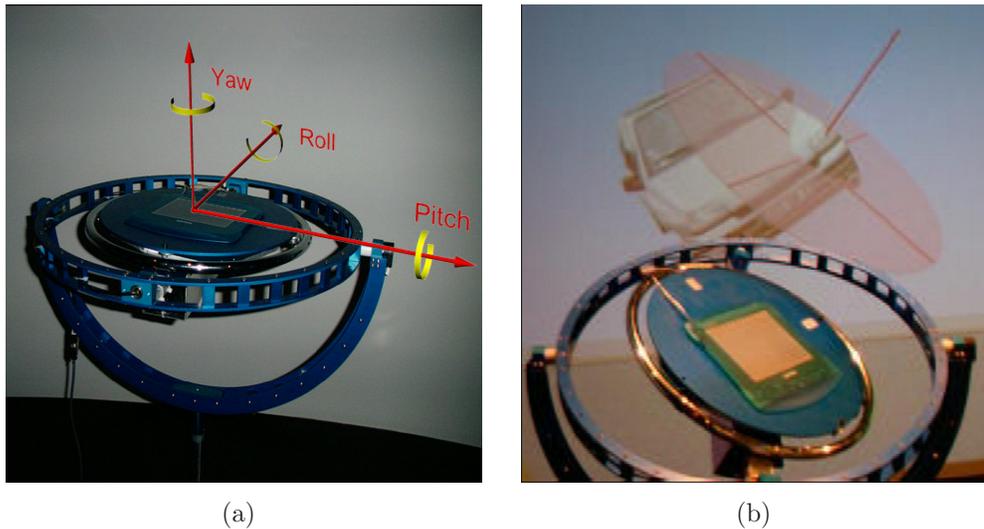


Abb. 2.6: Mit der CAT können für die Kamerasteuerung 6 Freiheitsgrade gleichzeitig genutzt werden (a). Die Bewegung der Tafel wird direkt in eine Bewegung der Kamera bzw. Szene übertragen (b). Quelle: HACHET et al. [2003]

Wahl der Betrachterposition ergeben sich neue Herausforderungen, welche bei statischen Darstellungen nicht auftreten. Dort werden die Blickwinkel so gewählt, dass Größen- bzw. Abstandsverhältnisse korrekt eingeschätzt werden können, sofern dies eine Anforderung an den Informationsgehalt der Darstellung ist. Einen solchen Blickwinkel interaktiv zu bestimmen, gestaltet sich kompliziert, was unter Umständen zu Fehleinschätzungen führen kann. Des Weiteren können wichtige Informationen aufgrund einer ungünstigen Betrachterposition übersehen werden. Auch mit den aus der manuellen Kamerasteuerung resultierenden Verdeckungen muss entsprechend umgegangen werden, da sie aus beliebigen Betrachterpositionen heraus auftreten können und dabei immer so behandelt werden müssen, dass durch Ausblendungen oder *Cut-Aways* keine wichtigen Kontextinformationen zerstört werden.

Die im nachfolgenden Abschnitt beschriebenen Klassen von Verfahren dienen dazu, die Informationsgewinnung bei der Exploration von dreidimensionalen Visualisierungen zu unterstützen.

2.2.4 Techniken zur Hervorhebung und Kontextverdeutlichung

Um eine computergestützte, interaktiv explorierbare Visualisierung ebenso informativ zu gestalten wie die klassischen statischen Abbildungen, kommen verschiedene Techniken zum Einsatz. Anhand des Grades der Einflussnahme auf die Visualisierung lassen sie sich, wie in [PREIM und RITTER, 2002] beschrieben, in drei Klassen unterteilen: lokale, regionale und globale Verfahren.

Lokale Verfahren. Darunter sind Techniken zu verstehen, welche sich direkt auf die hervorzuhebende Struktur beziehen. Sie wird in ihrer Darstellung angepasst, um den Fokus auf sie zu lenken. Dies kann durch eine farblich veränderte Darstellung der Struktur erreicht werden [DÖRGE, 2002]. Das Hinzufügen von metagraphischen Symbolen in abstrakter (z. B. Pfeilen, *Bounding Box*, Annotationen) oder an realen Objekten orientierter Form (Werkzeuge wie z. B. ein Skalpell) ist ebenfalls dieser Klasse von Hervorhebungstechniken zuzuordnen. Dadurch können zusätzliche, nicht direkt aus der Strukturgeometrie ableitbare Informationen visualisiert werden.

Regionale Verfahren. In vielen Fällen müssen zur Hervorhebung nicht nur die Fokusstruktur selbst, sondern auch andere, z. B. benachbarte Strukturen in ihren Darstellungen angepasst werden. Diese strukturübergreifenden Techniken fallen in den Bereich der regionalen Verfahren. Durch freie Wahl der Betrachterposition im Bereich der interaktiven Exploration ist die Verdeckung der Fokusstruktur ein häufig auftretendes Problem. Zur Behandlung solcher Verdeckungen sind verschiedene regionale Verfahren entwickelt worden. Um den Blick auf die Fokusstruktur zu ermöglichen, wird die Darstellung der verdeckenden Strukturen abgeschwächt bzw. angepasst.

Auf die gesamte verdeckende Struktur bezogene Techniken sind z. B. das vollständige Ausblenden, die Semitransparenz- bzw. Drahtgitterdarstellung oder NPR-Techniken wie die Darstellung in Form von Konturlinien [TIETJEN et al., 2005b]. Je nach Art dieser Verdeckungsbehandlungen gehen mehr oder weniger Kontextinformationen verloren. Um diesen Verlust zu reduzieren, kann die Darstellung verdeckender Strukturen durch sogenannte *Cut-Aways* auch nur teilweise abgeschwächt bzw. ausgeblendet werden [VIOLA et al., 2004; DIEPSTRATEN et al., 2003].

Globale Verfahren. Diese Hervorhebungstechniken beeinflussen die Visualisierung in ihrer Gesamtheit, indem globale Visualisierungsparameter angepasst werden. Ein typisches Beispiel hierfür ist die Positionierung der virtuellen Kamera, um die Sichtbarkeit der Fokusstruktur zu gewährleisten. Das in [MUEHLER et al., 2006] vorgestellte Skriptsystem ermöglicht neben der Anpassung von lokalen Darstellungsparametern auch die skriptbasierte Steuerung der Kamera und das Erzeugen von Animationen. Die zugrunde liegenden Kamerapositionen müssen manuell ermittelt werden.

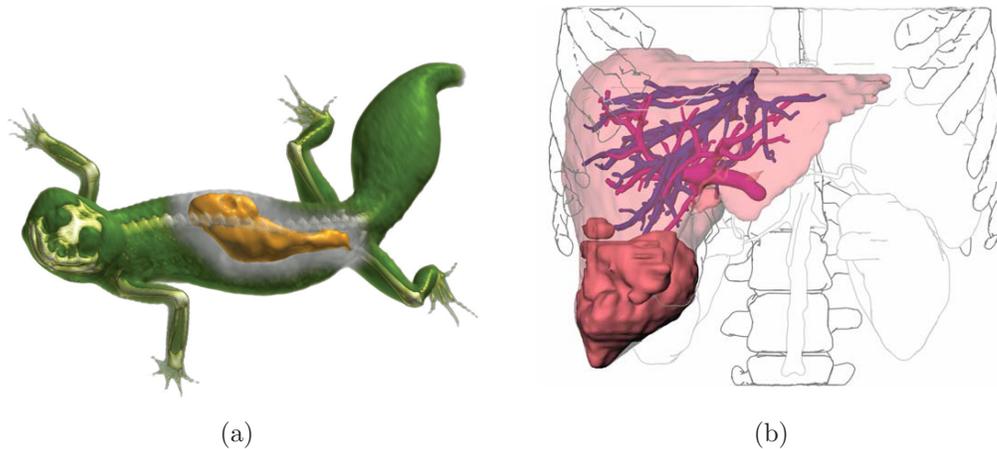


Abb. 2.7: Regionale Hervorhebungstechniken: In (a) werden verdeckende Strukturen entsprechend ihrer Wichtigkeit abgeschwächt dargestellt. Die Form dieses *Cut-Aways* orientiert sich an der Form der Fokustruktur. Mittels der teiltransparenten Darstellung der Leber und der Liniendarstellung der umgebenden Strukturen in (b) wird der Blick auf die Lebergefäße ermöglicht und der räumliche Kontext gewahrt. Quelle: (a) [VIOLA et al., 2004], (b) TIETJEN et al. [2005b]

Einen automatisierten Ansatz präsentiert der ZOOMILLUSTRATOR [PREIM et al., 1997]. Aus einer endlichen Menge an Sichtrichtungen wird jene gewählt, von der aus die Fokusstruktur durch die größte Anzahl von Pixeln repräsentiert wird. Eine solche Blickrichtung kann sich trotz guter Sichtbarkeit der Fokusstruktur als problematisch erweisen, da Betrachtungsgewohnheiten des Nutzers nicht mit einfließen. Ein sich eventuell ergebender, ungewohnter Blickwinkel kann zu Fehleinschätzungen der Lage- und Abstandsverhältnisse führen.

2.2.5 Zusammenfassung

Im günstigsten Fall sollten aus interaktiv explorierbaren, computergenerierten Visualisierungen ebenso leicht Informationen gewonnen werden, wie aus klassischen, handgezeichneten Darstellungen. Verschiedene Techniken zur Aufmerksamkeitssteuerung und Informationsvermittlung kommen zum Einsatz, um dem Nutzer die Zusammenhänge innerhalb der Visualisierung zu verdeutlichen. Der Vorteil der freien Exploration der Visualisierung birgt die Gefahr in sich, dass trotz dieser Techniken Strukturen übersehen oder Fakten falsch eingeschätzt werden. Hieraus ergibt sich, dass eine korrekte Wahl der Kameraposition entscheidend ist. Sie bildet die Basis, um aus den angewandten Hervorhebungstechniken die korrekten Schlussfolgerungen ziehen zu können. Eine manuelle Positionierung der Kamera kann sich vor allem für Nutzer, welche nicht im alltäglichen Umgang mit 3D-Anwendungen geschult sind, als sehr kompliziert erweisen.

Eine automatische Ermittlung günstiger Kamerapositionen, um die manuelle Navigation in den Visualisierungen zu unterstützen, ist somit wünschenswert. Am Beispiel des ZOOMILLUSTRATORS zeigt sich, dass die maximale Sichtbarkeit einer Fokusstruktur allein kein Garant für eine optimale Informationsgewinnung ist. Weitere Parameter müssen bei der Positionierung miteinbezogen werden. Da es trotzdem, gerade in den häufig kompakten medizinischen Visualisierungen, zu Verdeckungen kommen kann, sollte ein automatisiertes Positionierungsverfahren neben einer Kameraposition auch Informationen über vorhandene Verdeckungen liefern. Auf der Basis dieser Informationen könnten dann geeignete, regionale Hervorhebungstechniken ausgewählt werden, welche eine gute Sichtbarkeit der Fokusstruktur bei gleichzeitig möglichst geringem Informationsverlust gewährleisten.

Diese Problemstellungen liefern die Motivation für die hier vorliegende Arbeit. In den folgenden Abschnitten werden grundlegende Zusammenhänge und Mechanismen bezüglich der automatischen Kamerapositionierung erläutert.

2.3 Grundlagen zur virtuellen Kamera

Eine virtuelle Kamera stellt bei der Exploration dreidimensionaler Szenen die optische Schnittstelle zwischen Nutzer und Daten dar. Die verschiedenen Modellparameter der Kamera und die Art der Projektion haben Einfluss darauf, auf welche Weise die Szene dem Nutzer präsentiert wird.

Unterschiedliche Metaphern für die Kamerasteuerung beeinflussen nicht nur, wie der Nutzer mit der Kamera interagiert, sondern bestimmen auch, wie ihn automatische Verfahren bei der Navigation der Kamera durch die virtuelle Szene unterstützen können.

2.3.1 Kameramodell

Für die Bilderzeugung müssen die dreidimensionalen Objektkoordinaten in Bildschirmkoordinaten transformiert werden. Die Konzepte, welche bei dieser Transformation zum Einsatz kommen, orientieren sich an den Abbildungsprozessen einer realen Kamera, bei denen verschiedene Parameter Einfluss auf die resultierende Abbildung haben.

Dabei handelt es sich um Position und Blickrichtung der Kamera, welche in Weltkoordinaten angegeben werden. Weitere Faktoren sind Brennweite, Schärfentiefe und Objektivart, wobei letzteres Einfluss auf den Öffnungswinkel und somit auf das Projektionsverhalten hat.

Das resultierende Bild besitzt ebenfalls spezifische Parameter, wie die Größe und Seitenverhältnisse sowie die Auflösung. Bei der einfachen Visualisierung computergenerierter Objekte kommt ein vereinfachtes Kameramodell zum Einsatz. Durch einen linsenabhängigen Strahlengang hervorgerufene Effekte, wie z. B. der Schärfenbereich (*Depth of Field*) einer realen Kamera, werden nicht nachgebildet.

Die Parameter zur Beschreibung einer virtuellen Kamera sind die Position, die Blickrichtung und die Orientierung der Kamera, definiert durch den sogenannten *Up*-Vektor. Zusätzlich werden zwei Ebenen orthogonal zur Blickrichtung definiert. Nur Objekte zwischen diesen Ebenen werden zur Bilderzeugung verwendet. Diese sogenannten *Near* und *Far Clipping Planes* dienen zur Optimierung (sehr klein bzw. gar nicht abgebildete Objekte werden nicht gezeichnet), bilden allerdings auch die Grundlage für die notwendigen Tiefensortierung der abgebildeten Punkte (*z-Buffering*).

Die Abbildungsebene wird im sogenannten Viewport repräsentiert. Die Dimensionen und somit die Auflösung des Viewports ist für den Detailgrad des resultierenden Bildes verantwortlich, da die exakten, in Raumkoordinaten vorliegenden Objektpunkte in Form von Pixeln diskretisiert werden und hochfrequente Signale verloren gehen können [SCHULMAN, 1992]. Die Anordnung der Strahlen, welche zur Bilderzeugung mit der Abbildungsebene geschnitten werden, ist bei realen Kameras von den Faktoren Brennweite und Öffnungswinkel abhängig. Die Strahlen können dabei (annähernd) parallel zueinander verlaufen (große Brennweite) oder nicht parallel (kleine Brennweite), was eher der menschlichen Wahrnehmung entspricht. Bei der virtuellen Kamera spiegelt sich dies in der Verwendung einer orthographischen bzw. perspektivischen Projektionsmatrix wider [ANGEL, 2005].

2.3.2 Projektionsarten

Bei der **perspektivischen Projektion** treffen sich alle aus der Szene kommenden Sichtstrahlen in einem Punkt, dem *Center of Projection* (COP). Dieser Punkt ist im Normalfall symmetrisch bezüglich der Abbildungsebene angeordnet. Der durch COP und Abbildungsebene gebildete Projektionsraum entspricht somit einem symmetrischen Pyramidenstumpf.

Die **orthographische Projektion** kann als Sonderfall der perspektivischen Projektion interpretiert werden. Wird angenommen, dass der COP ins Unendliche verschoben wurde, liegen die einzelnen Sichtstrahlen parallel zueinander. Aus dem *Center of Projection* wird eine *Direction of Projection* (DOP) (siehe Abb. 2.8).

Aufgrund der parallelen Anordnung der Sichtstrahlen kommt es zu keiner abstandsbedingten Größenänderung bei der Projektion von Objekten auf die Abbildungsebene. Dadurch ist die Rate der perspektivischen Verkürzung nicht vom Abstand, sondern nur von der Orientierung des abgebildeten Objektes abhängig. Es können somit bildbasierte Messungen und Abstandseinschätzungen durchgeführt werden.

Eine klassische, auf der parallelen Projektion basierende Abbildung ist die isometrische Darstellung. Zwar verkürzt sich die Länge einer projizierten Linie verglichen zu ihrer wahren Ausdehnung im Objektraum, allerdings findet diese Verkürzung entlang aller drei Hauptachsen im selben Maße statt, wodurch Distanzmessungen weiterhin möglich sind. Erreicht wird die isometrische Darstellung, indem die Abbildungsebene die drei Hauptachsen in einem Winkel von 45° schneidet. Eine exakte Darstellung von Längen und Winkeln ist nur durch die Anwendung mehrerer, parallel zu den Hauptachsen ausgerichteter Abbildungsebenen möglich (*Multiview Orthographic Projection*).

Diese Art der Abbildung entspricht zwar nicht exakt der menschlichen Wahrnehmung, die Vorteile bei der Einschätzung räumlicher Verhältnisse begünstigen jedoch eine Anwendung im Rahmen medizinischer Visualisierungen.

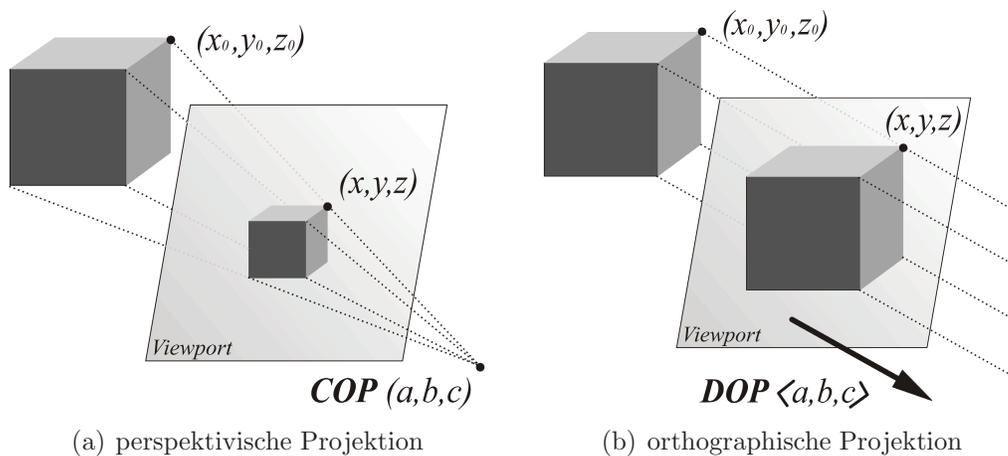


Abb. 2.8: Projektionsarten: In (a) tritt eine entfernungsabhängige Verkleinerung der Darstellung im Viewport auf. Durch die parallelen Projektionstrahlen in (b) bleiben die relativen Größenverhältnisse unabhängig vom Abstand zur Bildebene erhalten.

2.3.3 Kamerasteuerung

Bei der Aufgabe eine virtuelle Kamera zu positionieren, existieren sechs Freiheitsgrade. Drei Freiheitsgrade beziehen sich auf die Position der Kamera im Raum (x, y, z) , weitere drei bestimmen die Ausrichtung der Kamera in Form von drei, an den Achsen orientierten Winkeln. Ein für die Kamera gültiger siebter Freiheitsgrad bezieht sich auf den Öffnungswinkel und somit auf die Größenverhältnisse der Darstellung auf der Abbildungsebene [CHRISTIE und LANGUÉNOU, 2003]. Da dieser im Rahmen der Positionierung nicht zum Einsatz kommt, wird er im Folgenden vernachlässigt. Um verschiedene Konzepte der Kamerakontrolle zu beschreiben, werden Metaphern verwendet. Diese beschreiben den zu erwartenden Effekt bei der Steuerung der Kamera

anhand „natürlicher“ Entsprechungen. Die hier vorgestellten Metaphern sind *Eyeball in Hand*, *Scene in Hand* sowie *Flying Vehicle Control* [WARE und OSBORNE, 1990]. Diese Metaphern unterscheiden sich bezüglich der Art und Weise, wie nutzerseitige Eingaben auf die Freiheitsgrade abgebildet werden. Auch wenn sie sich vorrangig auf die Anwendung spezieller Eingabegeräte zur gleichzeitigen Manipulation der sechs Freiheitsgrade beziehen, ist eine Übertragung auf Standardeingabegeräte wie die Maus durchaus möglich.

„Eyeball in Hand“-Metapher

Die Kamera wird frei in der virtuellen Umgebung platziert. Dies wird erreicht, indem durch ein Eingabegerät die Translation und Rotation der Kamera gesteuert wird. Diese Art der Kamerasteuerung eignet sich vor allem für die Simulation endoskopischer Eingriffe [STAMPE, 2006]. Dabei kommen meist spezielle Eingabegeräte zum Einsatz, mit denen alle sechs Freiheitsgrade gleichzeitig abgedeckt werden können. Eine Übertragung auf ein zweidimensionales Eingabegerät ist nur über Umwege erreichbar. So kann ein Modell einer Kamera abgebildet und mittels Mauseingabe (evtl. unter Zuhilfenahme der Tastatur) rotiert und transliert wird, während auf einem zweiten Viewport das resultierende Kamerabild dargestellt wird (Kamerapositionierung in Modellierungssoftware wie z. B. 3D STUDIO MAX).

Ein speziell für die *Eyeball in Hand*-Metapher geeignetes Eingabegerät ist z. B. das Phantom [MASSIE und SALISBURY, 1994]. Es ermöglicht die taktile Wahrnehmung virtueller Objekte durch ein entsprechendes Feedback bei der Steuerung. Dies ermöglicht z. B. die realistische Simulation eines endoskopischen Eingriffs und unterstützt den Benutzer direkt bei der Exploration der virtuellen Umgebung. Ohne ein haptisches Feedback stellt sich die Steuerung vor allem für ungeübte Nutzer als schwierig dar, wie eine Evaluierung von WARE und OSBORNE [1990] ergeben hat. Vor allem die Invertierung der Steuerung, die auftritt wenn die Kamera dem Nutzer bei der rückwärtigen Betrachtung virtueller Objekte gegenübersteht, ist aufgrund der Unausgeglichenheit zwischen Bewegung und optischem Feedback mit Irritationen verbunden.

„Scene in Hand“-Metapher

Die Orientierung und Position der Szene richtet sich direkt nach den Nutzereingaben. Wenn der Nutzer das Eingabegerät rotiert, rotiert die Szene, transliert er das Eingabegerät, so kommt es zu einer Bewegung der Szene. Voraussetzung für eine solche direkte und gleichzeitige Manipulation ist ein Eingabegerät mit sechs Freiheitsgraden, wie z. B. die sogenannte BAT [WARE und JESSOME, 1988]. Das Gerät ermöglicht die Steuerung der relativen Position, ähnlich wie eine Maus, liefert allerdings zusätzlich Daten bezüglich aller sechs Dimensionen, welche zur Steuerung virtueller Objekte genutzt werden.

Durch die Mehrfachbelegung der zwei Freiheitsgrade einer konventionellen Maus und entsprechende nutzerseitige Auswahl der betreffenden Freiheitsgrade kann die *Scene in Hand*-Metapher auch bei der Kamerasteuerung durch 2D-Eingabegeräte zum Einsatz kommen.

Die Rotation erfolgt bei dieser Metapher um das Zentrum der Szene. Vor allem kompakte Szenen lassen sich so leicht explorieren. Bei ausgedehnten Szenen und einem entsprechend großen Abstand zum Mittelpunkt erweist sich die Steuerung allerdings als kompliziert, da kleine Änderungen der Rotationswinkel in großen Positionsänderungen resultieren. Wenn die Winkelgeschwindigkeit verkleinert wird, um diesem Effekt entgegenzuwirken, ist die Rotation der gesamten Szene mit entsprechend hohem Zeit- und Interaktionsaufwand verbunden, da die Maus nachgeführt werden muss. Während kompakte Szenen in ihrer Gesamtheit und auch die entsprechenden Lagebeziehungen innerhalb der Szene einfach erfasst werden können, erweist sich das Nachverfolgen eines Pfades, wie z. B. bei der inneren Exploration eines Gefäßes, aufgrund des am Szenenmittelpunkts orientierten Bezugspunktes als sehr kompliziert.

„Flying Vehicle Control“ - Metapher

Der Benutzer ist in der Lage, den Betrachterstandpunkt durch die virtuelle Umgebung zu steuern. Er kontrolliert dabei die räumliche Geschwindigkeit sowie die Geschwindigkeit der Winkeländerung. Diese Art der Steuerung ist mit einem sehr einfachen Flugmodell zu vergleichen. Wie schon bei der *Scene in Hand*-Metapher ist sie durch entsprechende Anpassungen auf ein 2D-Eingabegerät übertragbar.

Die Verfolgung eines bestimmten Pfades ist auf diese Art und Weise einfach zu bewerkstelligen. Das „Umfliegen“ einer Szene, bei gleichzeitigem Blick auf die Szene, ist allerdings nur durch ein komplexes Zusammenspiel der spatialen und winkelbezogenen Geschwindigkeiten möglich. Für diese Art der Exploration ist die *Scene in Hand*-Metapher besser geeignet. Auch das schnelle Wechseln der Kamerapositionen ist nicht möglich, da die Positionen nicht direkt durch die Nutzereingaben beeinflusst werden. Die Geschwindigkeit kann zwar entsprechend erhöht werden, jedoch muss der Nutzer darauf warten, bis das entsprechende Ziel erreicht wurde.

2.3.4 Zusammenfassung

Keine der hier vorgestellten Metaphern ist durchgehend geeignet bzw. ungeeignet. Wie die in WARE und OSBORNE [1990] durchgeführte Evaluierung zeigt, ist die Anwendbarkeit der verschiedenen Steuerungstechniken abhängig vom Einsatzgebiet. Die *Eyeball in Hand*-Metapher eignet sich vor allem für Anwendungsfälle, bei denen eine entsprechende Kamerasteuerung auch in der Realität durchgeführt wird (z. B. Endoskopie).

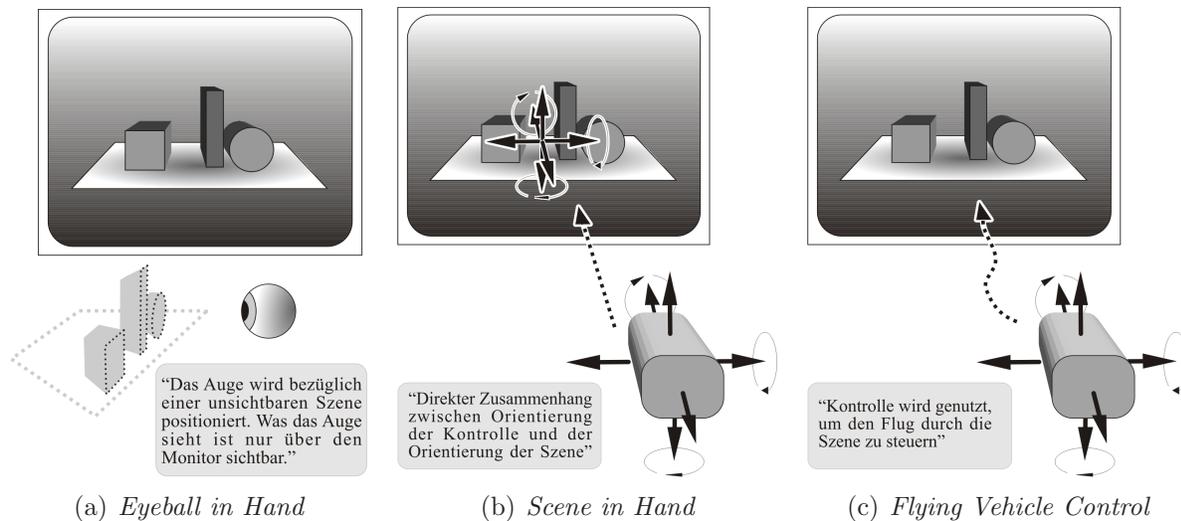


Abb. 2.9: Die drei Metaphern der Kamerasteuerung

Die *Scene in Hand*-Metapher dient dazu, sich einen Überblick über kompakte Szenen und deren interne Lagebeziehungen zu verschaffen. Sie ist somit für die Exploration kompakter, medizinischer Visualisierungen geeignet. Diese werden meist von einer externen Betrachterposition aus untersucht. Abhängig von der spezifischen Aufgabenstellung, unter der eine solche Exploration stattfindet, sind des Weiteren oft Abstands- und Größeninformationen zu den einzelnen enthaltenen Strukturen von Interesse. Für die Exploration ausgedehnter Szenen („Landschaften“) oder das Nachverfolgen bestimmter Pfade („Gefäßexploration“) ist diese Metapher ungeeignet. Hier bietet sich am ehesten die *Flying Vehicle Control*-Metapher an.

Neben den verschiedenen Einsatzgebieten sind allerdings auch die Gewohnheiten der Nutzer für den effektiven Einsatz der Steuerungstechniken entscheidend. Durch die entsprechende Vorbildung fällt es einem Chirurgen nicht allzu schwer, sich daran zu gewöhnen, mittels eines Gerätes wie einem „Phantom“ einen endoskopischen Eingriff zu simulieren, da er diese Art der Steuerung zumindest ähnlich aus seinem alltäglichen Arbeitsumfeld kennt. Solches Vorwissen kann allerdings auch einen negativen Einfluss darstellen, wie der Fall eines Piloten zeigt, der bei der Evaluierung aus WARE und OSBORNE [1990] große Probleme mit der *Flying Vehicle Control*-Metapher hatte, da sie teilweise einem realistischen Flugverhalten entspricht und er aufgrund dieser Ähnlichkeit die „unrealistische“ Möglichkeit rück-, auf- oder abwärts zu fliegen gar nicht erst nutzte.

Es zeigt sich, dass eine gewisse Erfahrung notwendig ist, um eine Kamera effektiv durch eine dreidimensionale Szene zu navigieren. Eine automatische Positionierung der Kamera, um eine günstige Ausgangsposition für eine weiterführende, manuelle Exploration zu erhalten, kann dem Nutzer bei der Navigation helfen. Im folgenden

Abschnitt werden Verfahren präsentiert, die bei einer automatischen Positionierung der Kamera zum Einsatz kommen.

2.4 Verfahren zur Kamerapositionierung

Es gibt einige Aufgabenstellungen im Bereich der dreidimensionalen Visualisierung, bei denen eine automatische Positionierung der Kamera wünschenswert ist. So z. B. als globale Hervorhebungstechnik, um den Nutzer auf eine ausgewählte Struktur hinzuweisen und so die Gefahr, dass diese Struktur übersehen wird, zu minimieren. Auch bei der komplexen Aufgabe der Navigation durch eine dreidimensionale Szene können automatisch generierte Positionen als Ausgangs- und Orientierungspunkte dienen.

Als Voraussetzung für ein derartiges, unterstützendes Verfahren müssen die Sichtbarkeitsverhältnisse der Szene analysiert werden, was je nach Anforderung bezüglich der Laufzeit und Genauigkeit auf verschiedene Weise durchgeführt werden kann.

Aussagen über die Sichtbarkeitsverhältnisse allein reichen nicht aus, um den Informationsgehalt und somit die Güte einer Kameraposition zu bewerten. Dazu müssen weitere Parameter aus der Szene gewonnen werden. Je nach Art dieser Parameter kann in *Low*-, *Mid*- und *High-Level*-Verfahren zur Positionsbewertung unterschieden werden.

Die auf diesen und weiteren Informationen basierenden Systeme zur automatischen Positionierung der Kamera unterscheiden sich durch verschiedene Lösungsstrategien, unterschiedliche Interpretationen der dreidimensionalen Szenen und die Vollständigkeit und Genauigkeit der Resultate.

2.4.1 Sichtbarkeitsbestimmung

Eine günstige Kameraposition definiert sich unter anderem dadurch, dass die Fokusstruktur sichtbar ist. Um Kamerapositionen zu ermitteln, von denen aus diese Sichtbarkeit gewährleistet ist, müssen Sichtbarkeitsverhältnisse innerhalb der Szenen ermittelt werden.

Klassifizierung

Im Bereich der Computergrafik gibt es eine Vielzahl an Problemstellungen, die eine Analyse der Sichtbarkeitsverhältnisse erfordern. Die häufigste Anwendung findet im Bereich der *Visible Surface Determination* statt [BITTNER und WONKA, 2003]. Es wird ermittelt, welche Szenenelemente dargestellt werden müssen, damit die Szene von der Betrachterposition aus gesehen vollständig erscheint. Verdeckte Elemente oder Elemente außerhalb des Sichtbereiches der Kamera werden nicht dargestellt. So ist es möglich, die

Anzahl der zu berechnenden Geometrien zu verringern, um auch komplexe Szenen in Echtzeit darstellen zu können [HILLESLAND et al., 2002].

Neben dieser Standardanwendung ergeben sich im Rahmen steigender Ansprüche an die Computergrafik auch weitere Problemstellungen, wie die Schattenberechnung oder globale Beleuchtungsberechnungen. Eine Vielzahl von Verfahren mit verschiedenen Anforderungen und Mächtigkeiten wurde entwickelt.

Durch die Anwendung bei der Positionierung einer Kamera ergeben sich ebenfalls spezielle Anforderungen und Einschränkungen und somit eine bestimmte Klassifizierung der Probleme und Verfahren, welche bei der Sichtbarkeitsbestimmung auftreten bzw. zum Einsatz kommen. Nach [BITTNER und WONKA, 2003] ergeben sich folgende Problemklassen:

- Sichtbarkeit entlang einer Linie
- Sichtbarkeit von einem Punkt aus
- Sichtbarkeit von einem Liniensegment aus
- Sichtbarkeit von einem Polygon aus
- Sichtbarkeit von einer Region aus
- Globale Sichtbarkeit

Für die Kamerapositionierung kommen zwei Problemklassen bei der Sichtbarkeitsbestimmung in Frage und zwar Sichtbarkeit von einem Punkt aus (*Point Visibility* - PV) und die globale Sichtbarkeit (*Global Visibility* - GV). Die PV entspricht dem Fall, dass die Kamera an einem bestimmten Punkt positioniert wurde und nun ermittelt werden muss, welche Sichtbarkeitsverhältnisse vorliegen, d.h. ob die Fokusstruktur sichtbar ist oder nicht. Eine weitere Verfeinerung stellt die sogenannte *Point to Region Visibility* (PRV) dar, welche Aussagen darüber beinhaltet, welcher Anteil einer Region (z. B. die Fokusstruktur) von einem bestimmten Punkt aus sichtbar ist [TELLER und SÉQUIN, 1991].

Die Lösung des GV-Problems ist insofern mächtiger, als dass alle Szenenkomponenten in Betracht gezogen werden und so positionsunabhängig und allgemeingültig die Sichtbarkeitsverhältnisse in der Szene beschrieben werden. Anhand dieser Informationen kann die Kamera so positioniert werden, dass die gewünschten Sichtbarkeitsverhältnisse vorliegen. Zusammenfassend kann gesagt werden, dass die Bestimmung der Sichtbarkeitsverhältnisse entweder von der Kamera (PV bzw. PRV) oder global von der Szene ausgehend (GV) durchgeführt werden kann.

Ebenso wie die Problembeschreibung können auch die Verfahren zur Lösung in verschiedene Klassen unterteilt werden. Die im Rahmen dieser Arbeit gewählte Klassifizierung bezieht sich dabei auf die Mächtigkeit der Ergebnisse. Lokale Verfahren zur Sichtbarkeitsbestimmung liefern Informationen darüber, inwieweit die Fokusstruktur von der

Kameraposition aus sichtbar ist. Globale Verfahren liefern zusätzlich szenenbezogene Informationen, z. B. über verdeckende Strukturen oder die Grade der gegenseitigen Verdeckung bzw. Sichtbarkeit.

Angelehnt an [COHEN-OR et al., 2003] können diese Klassen abhängig von der Genauigkeit der Ergebnisse weiter differenziert werden. Exakte Verfahren liefern als Ergebnisse analytisch exakte Informationen über die Sichtbarkeitsverhältnisse, unter Vernachlässigung der endlichen Genauigkeit computergestützter Fließkommaberechnung. Approximative Verfahren beschreiben die Sichtbarkeitsverhältnisse nur näherungsweise, indem ein endlicher, diskreter Lösungsraum, etwa durch die Verwendung einer endlichen Menge möglicher Kamerapositionen, geliefert wird. Welche Verfahren letztendlich zum Einsatz kommen, richtet sich danach, welche Anforderungen an die Laufzeit sowie die Genauigkeit und Mächtigkeit der Ergebnisse gestellt werden.

Ansätze zur Sichtbarkeitsbestimmung

Im Folgenden werden verschiedene Verfahren und ihre Ansätze zur Sichtbarkeitsbestimmung vorgestellt. Im Rahmen der Recherche hat sich ergeben, dass es eine starke, wenn auch nicht vollständige Korrelation zwischen der Lösung des GV-Problems und der obigen Definition globaler Verfahren gibt. Deshalb wurde die Unterscheidung der Verfahren nach PV- und GV-Ansätzen, jeweils exakt und approximativ, strukturiert. Lösungen für PV liefern der VISIBILITY SOLVER (approximativ) und der Ansatz nach Chu (exakt). GV-orientiert sind die Radiosity-Verfahren (approximativ) und das VISIBILITY SKELETON (exakt).

Visibility Solver. In Anwendungsgebieten, bei denen eine echtzeitfähige Berechnung der Kameraposition notwendig ist, kommen vorrangig lokale Verfahren zum Einsatz. Diese besitzen, verglichen mit den globalen Verfahren, einen geringeren Berechnungsaufwand. Computerspiele sind ein üblicher Anwendungsbereich, bei dem eine echtzeitfähige Positionierung der Kamera notwendig ist. Nur wenn dem Spieler eine optimale Sicht auf das Spielgeschehen ermöglicht wird, kann er den Spielverlauf effektiv steuern. In [HALPER, 2003] S.98ff wird der sogenannte VISIBILITY SOLVER vorgestellt. Er liefert Kamerapositionen, von denen aus die Fokusstruktur, in diesem Fall z. B. die Spielerfigur, sichtbar ist.

Es wird ein bildbasierter, d.h. approximativer Ansatz verwendet. Ausgehend von der Fokusstruktur wird in Richtung der Kameraposition eine z -Buffer-Bild erzeugt. In den Buffer werden dann geometrische Strukturen gezeichnet, welche die *Potential Visibility Regions* (PVRs) repräsentieren. Die PVRs stellen dabei jene Bereiche dar, in denen die Kamera positioniert werden darf (z. B. eine die Spielszene umgebende Kugel). Unterschiedliche Intensitäten auf einer PVR dienen dazu, verschiedene Bereiche zu wichten. Je heller ein Bereich ist, desto wünschenswerter ist die Positionierung der Kamera in diesem Bereich der PVR. Die Teile der PVR, welche den Tiefentest bestehen,

sind potentielle Kamerapositionen. Der hellste Bereich, welcher sich am dichtesten an der Ausgangsposition der Kamera befindet, stellt die Ergebnisposition dar (siehe Abb. 2.10(a)).

Dieses Verfahren ist nach der oben eingeführten Definition ein lokales Verfahren, da keine Informationen über die verdeckenden Elemente vorliegen. Des Weiteren wird das Problem der Sichtbarkeitsanalyse eher nach dem PV-Ansatz gelöst, da keine allgemeingültigen Sichtbarkeitsinformationen ermittelt werden, sondern von einer mehr oder weniger fest definierten Kameraposition bzw. einem Positionierungsbereich ausgegangen wird.

Die Kameraposition ist nicht beliebig wählbar, sondern orientiert sich an den PVRs. Durch entsprechende Formwahl bei der PVR kann so Einfluss auf das Kameraverhalten genommen werden (siehe Abb. 2.10(b)). Der Fall einer vollständigen Verdeckung wird nur dadurch behandelt, dass die Kamera so dicht an die Fokusstruktur herangeführt wird, bis diese sichtbar ist. Es liegen somit viele Einschränkungen vor, die aber aufgrund des kontrollierbaren Rahmens, in Form einer definierten geometrischen Regeln folgenden Spielumgebung, trotzdem die Anwendbarkeit des Verfahrens kaum beeinträchtigen.

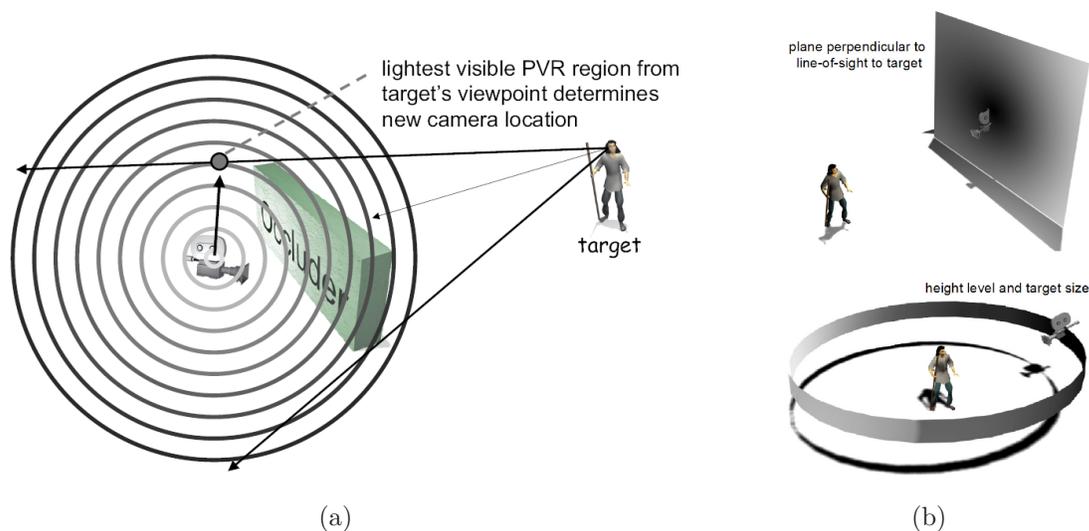


Abb. 2.10: Die hellste PVR, welche von der Zielstruktur aus sichtbar ist, liefert die am nächsten gelegene Kameraposition, von der aus das Ziel sichtbar ist (a). Die verschieden geformten PVRs in (b) werden genutzt, um den Bewegungsbereich der Kamera gezielt einzuschränken. Quelle: HALPER [2003]

Ansatz nach Chu. Im Bereich der z. B. bei der automatischen Fertigung angewandten *Machine Vision* ist es wichtig, verlässliche visuelle Informationen zu erhalten. In [CHU und CHUNG, 2002] wird ein System vorgestellt, welches im Bereich aktiver Kamerasysteme zum Einsatz kommt. Die Aufgabe ist dabei, die Kamera so zu positionieren, dass eine freie Sicht auf das Zielobjekt ermöglicht wird.

Um dies zu erreichen, wird eine sphärische Projektion genutzt. Dies ist möglich, da die potentiellen Kamerapositionen auf eine, die Szene umspannende Halbkugel eingeschränkt werden, deren Mittelpunkt durch das Zielobjekt definiert wird. Um die Sichtbarkeitsverhältnisse auf diese Halbkugel zu projizieren, werden zwei Schritte durchgeführt, die Projektion und die Ermittlung der konvexen Hülle der projizierten Punkte (siehe Abb. 2.11).

Jeder Eckpunkt des Zielobjektes und die Eckpunkten eines verdeckenden Objektes werden zur Definition von Geraden verwendet, welche die Halbkugel durchstoßen. Die auf der Kugeloberfläche berechnete konvexe Hülle aller Durchstoßpunkte stellt den Verdeckungsbereich dar, welcher durch das verdeckende Objekt erzeugt wird. Die Anzahl aller Verdeckungsbereiche entspricht der Anzahl der verdeckenden Objekte. Wird die Kamera auf der Kugeloberfläche außerhalb dieser Verdeckungsbereiche positioniert, ist der Blick auf das Zielobjekt frei.

Dieser Ansatz entspricht der Lösung des PV-Problems, da nur bezüglich definierter Kamerapositionen eine Lösung präsentiert wird. Die Genauigkeit des Ergebnisses ist nach der oben genannten Definition exakt, da es auf analytisch ermittelten Geraden und Schnittpunkten basiert. Des Weiteren ist es als lokales Verfahren zu betrachten, da es in der beschriebenen Form keine Informationen über die verdeckenden Objekte liefert. Durch die Möglichkeit zur Unterscheidung der einzelnen projizierten Punkte lässt es sich jedoch erweitern, so dass es bei Verdeckung zusätzliche Informationen über die verdeckenden Objekte liefern kann.

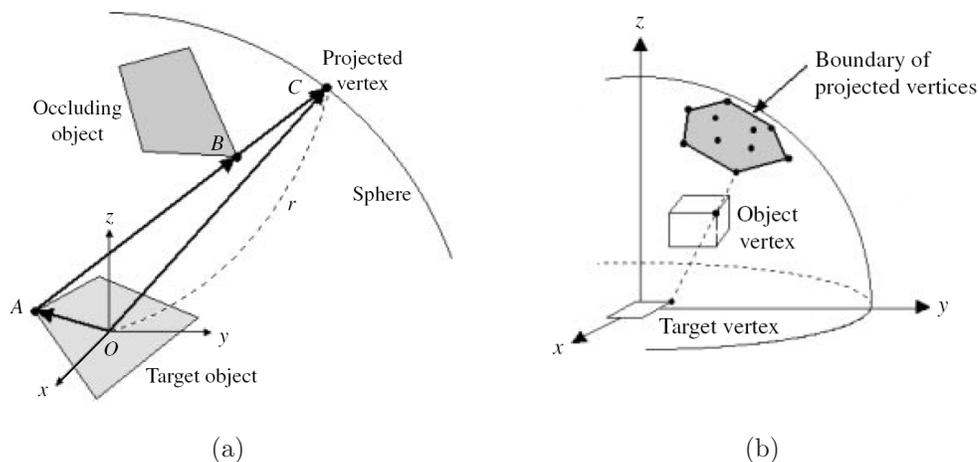


Abb. 2.11: Die Berechnung der projizierten Punkte (a) und der resultierende Verdeckungsbereich (b) nach der Berechnung der konvexen Hülle auf der Kugeloberfläche. Quelle: CHU und CHUNG [2002]

Radiosity-Verfahren. Diese Klasse von Verfahren basiert auf der Berechnung der Lichtverteilung im Raum [HANRAHAN et al., 1991]. Die mathematische Grundlage dafür bietet die Rendering Gleichung. Sie gibt an, wie viel Lichtenergie von der Umgebung auf einen Oberflächenpunkt fließt. Da der sich ergebende Funktionsraum unendlich ist und nur unter sehr speziellen Umständen eine analytische Beschreibung der Funktion möglich ist, kann im Allgemeinen eine Lösung nur auf der Basis von Approximationen ermittelt werden.

Die Oberflächen innerhalb der Szenen werden in kleine Elemente unterteilt (siehe Abb. 2.12). Für jedes Element wird ermittelt, wie viel Lichtenergie von der Umgebung an dieses Element abgegeben wird. Bei der Durchführung des Verfahrens wird die Verteilung der Lichtenergie in der Szene iterativ simuliert. Während zu Beginn des Verfahrens nur die Lichtquelle Energie an die Umgebung abgibt, sind in den nächsten Schritten auch nicht selbstleuchtende Elemente in der Lage, Energie abzugeben, da sie Energie von der Lichtquelle erhalten haben. Je mehr Schritte durchgeführt werden, desto stärker nähert man sich an eine optimale Verteilung des Lichtes an. Der Verlauf dieser Annäherung hat einen konvergierenden Charakter. Die Formfaktoren (FF) sagen aus, wie viel Energie

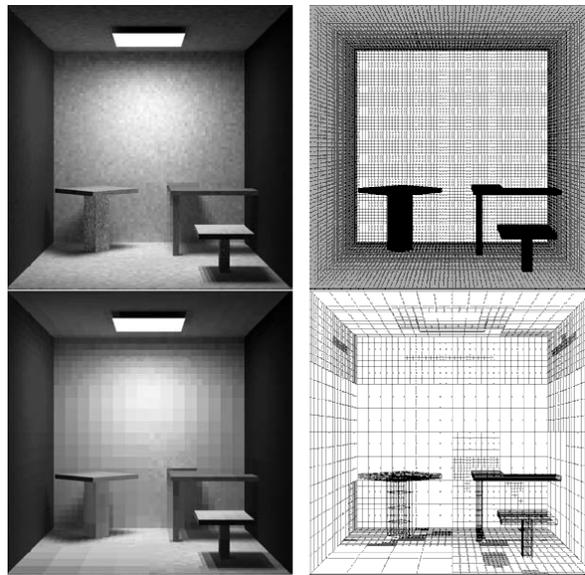


Abb. 2.12: Gleichmäßige (oben) und adaptive (unten) Unterteilung in Oberflächenelemente. Anhand dieser Elemente wird die Verteilung der Lichtenergie in der Szene approximativ ermittelt. Ein wichtiger Faktor ist dabei, welche Elemente Quelle: TOBLER et al. [1997]

von einem Oberflächenelement auf ein anderes Element übertragen wird. Neben der Orientierung der Elemente spielt vor allem ihre gegenseitige Sichtbarkeit eine Rolle. In einigen Sonderfällen ist es möglich, die FF analytisch zu ermitteln. Im Allgemeinen wird auf numerische Verfahren zurückgegriffen, wie z. B. das *Sampling* über einen das Element umgebenden *Hemicube* [COHEN et al., 1988].

Das Verfahren ist globaler Natur, da als Ergebnis die Sichtbarkeitsverhältnisse der kompletten Szene vorliegen. Es ist bekannt, ob und zu welchem Grad die Szenenbestandteile gegenseitig sichtbar sind. Die Lösung entspricht einem GV-Ansatz für die Sichtbarkeitsbestimmung. Des Weiteren ist das Verfahren approximativ, da die Oberfläche in eine endliche Menge an Elementen unterteilt wird. Kommt es zu einer numerischen Berechnung der Formfaktoren, dann wird zusätzlich über diese eine weitere Approximation durchgeführt.

Visibility Skeleton. Zwei Punkte sind gegenseitig sichtbar, wenn eine sie verbindende Linie keine weiteren Szenenelemente schneidet. Daraus lässt sich ableiten, dass Linien zur Beschreibung von Sichtbarkeitsverhältnissen genutzt werden können [BITTNER und WONKA, 2003]. Das VISIBILITY SKELETON [DURAND et al., 1997a] ist eine Datenstruktur, welche die Sichtbarkeitsverhältnisse innerhalb einer Szenen anhand einfacher geometrischer Strukturen beschreibt. Diese Strukturen werden durch sogenannte visuelle Ereignisse gebildet. Die Ereignisse beschreiben die Änderung der Sichtbarkeitsverhältnisse in einer Szene.

Ein solches Ereignis tritt auf, wenn z. B. der Eckpunkt eines Polygons bei der Änderung des Betrachterstandpunktes die Kante eines dahinter liegenden Polygons „überschritten“ hat, d.h. nicht mehr das dahinter liegende Polygon verdeckt, sondern nur noch den Hintergrund (siehe Abb.2.13(a)). In diesem Fall bilden der Eckpunkt des vorderen Polygons und die Kante des hinteren Polygons die Grundlage für eine Linienschar, welche das visuelle Ereignis beschreibt. Diese sogenannten kritischen Linienscharen bilden die Kanten des VISIBILITY SKELETON Graphen. Die aus der Überschneidung von Scharen resultierenden sogenannten äußeren Schnittlinien fließen in Form von Knoten in den Graphen ein (siehe Abb.2.13(c)).

Aus den verschiedenen Kombinationsmöglichkeiten geometrischer Elemente ergibt sich eine Vielzahl geometrischer Strukturen, welche die möglichen visuellen Ereignisse in einer Szene beschreiben und im VISIBILITY SKELETON repräsentiert werden.

Das Verfahren ist global, da für jedes Element der Szene Informationen über die Sichtbarkeitsverhältnisse vorliegen. Des Weiteren ist es gemäß der oben genannten Definition exakt, da die Sichtbarkeitsverhältnisse und deren Änderung durch die Verwendung geometrischer Konstrukte analytisch und nicht annäherungsweise beschrieben werden. Die Lösung entspricht, wie die des Radiosity-Verfahrens, einer von der Kameraposition unabhängigen GV-Lösung.

Zusammenfassung

Wie eingangs erwähnt, können globale Sichtbarkeitsinformationen genutzt werden, um zu entscheiden, wo die Kamera platziert werden soll. Vorteil der GV-Verfahren ist die Tatsache, dass alle Elemente der Szene in die Sichtbarkeitsanalyse eingebunden sind.

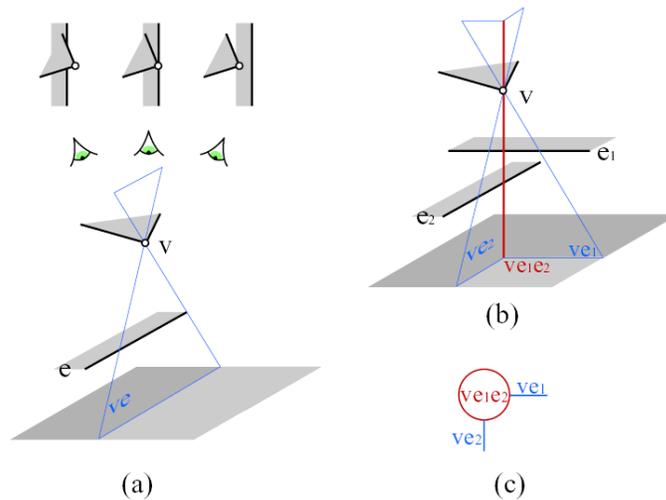


Abb. 2.13: Die Linienschar entspricht der Sichtbarkeitsänderung (a), zwei Linienscharen bilden eine äußere Schnittlinie, die sich im Graph (c) als Knoten widerspiegelt. Quelle: DURAND et al. [1997b]

Somit liegen meist nicht nur Informationen über Sichtbarkeit bzw. Nichtsichtbarkeit der Fokusstruktur vor, sondern im Falle einer Verdeckung auch Informationen über die verdeckende Struktur und den Grad der Verdeckung. Ein schneller Wechsel der Fokusstruktur ist durchführbar, da für alle Elemente der Szenen Informationen über ihre Sichtbarkeit vorliegen. Nachteilig ist der hohe Aufwand bei der vollständigen Beschreibung der Sichtbarkeitsverhältnisse.

Wenn es möglich ist, die Menge potentieller Kamerapositionen räumlich einzuschränken, bieten sich PV-Verfahren an. Diese liefern Sichtbarkeitsinformationen bezüglich einer bestimmten Kameraposition, bzw. Region. Allerdings haben diese Verfahren oft lokalen Charakter. Die Sichtbarkeitsinformationen liegen nur für eine Fokusstruktur vor und umfassen den Grad ihrer Sichtbarkeit. Zusätzliche Informationen, welche für die Behandlung von Verdeckungen genutzt werden können, fehlen meist. Ein effektives Verfahren zur Sichtbarkeitsbestimmung im Rahmen der Kamerapositionierung sollte demnach die Vorzüge von PV- und GV-Verfahren kombinieren. Die Einschränkung des Lösungsraums auf bestimmte Bereiche, deren Form von der Art der Kamerasteuerung abhängig ist, ist in vielen Fällen möglich. Zusätzlich müssen globale Informationen gewonnen werden, um im Falle einer Verdeckung z. B. Aussagen über auszublenkende Strukturen zu treffen und einen effektiven Wechsel der Fokusstruktur zu ermöglichen.

2.4.2 Viewpoint Quality

Ein hoher Grad an Sichtbarkeit der Fokusstruktur ist grundsätzlich eine gute Ausgangsbasis, um Informationen über sie zu gewinnen. Allerdings kann über die Güte einer Kameraposition in vielen Fällen nicht allein aufgrund der Sichtbarkeit entschieden werden. Eine Fokusstruktur kann von vielen Positionen aus sichtbar sein, aber unter Umständen liefern nur wenige dieser Positionen die Art von Informationen, an denen der Nutzer interessiert ist.

Die Auswahl solcher günstigen Kamerapositionen ist somit von weiteren Faktoren abhängig. Dazu zählen neben den verschiedenen Anwendungsbereichen zumindest teilweise definierbare Faktoren wie das Fachgebiet und die Vorbildung des Nutzers, aber auch sehr subjektive, kulturell geprägte Faktoren wie dessen Vorlieben und ästhetisches Empfinden.

Im Bereich der wissenschaftlichen Visualisierungen ist eine günstige Kameraposition diejenige, welche den Nutzer befähigt, schnell und korrekt die gewünschten Informationen aus der virtuellen Szene zu gewinnen. Die Güte einer Kameraposition steht demzufolge im direkten Zusammenhang zur Menge der durch sie erfassbaren Informationen. Verschiedene Ansätze wurden entwickelt, um quantitative Aussagen über diesen Informationsgehalt treffen zu können. Sie lassen sich in *Low-*, *Mid-* und *High-Level*-Verfahren einteilen.

Der Begriff *Viewpoint Entropy* als Maß für den Informationsgehalt einer Kameraposition wird in [VÁZQUEZ et al., 2001] eingeführt. Vázquez geht von der Annahme aus, dass gute Sichtbarkeit gleichbedeutend mit hohem Informationsgehalt ist. Dabei bezieht er sich nicht auf die Sichtbarkeit einer speziellen Fokusstruktur, sondern auf die Sichtbarkeit der Polygone aus denen die Szene aufgebaut ist. Es handelt sich um einen *Low-Level*-Ansatz, da keine semantische Abgrenzung der einzelnen Szenenelemente durchgeführt wird. Die Szene wird als Einheit betrachtet. Des Weiteren werden für die Einschätzung des Informationsgehaltes die Polygone unabhängig voneinander betrachtet. Das Maß für die Entropie wird aus der projizierten Fläche der Szene und der Anzahl der sichtbaren Polygone gewonnen.

Der aus diesen Parametern abgeleitete Informationsgehalt ist allerdings stark von der initialen Unterteilung der Szenen abhängig. So wird z. B. einer quadratischen Fläche, die aus wenigen Polygonen besteht, ein geringerer Informationsgehalt zugeordnet, als einer gleichgroßen, aus vielen Polygonen aufgebauten Fläche, obwohl diese optisch völlig gleich erscheinen können (siehe Abb 2.14(a)).

Dies wird durch den *Mid-Level*-Ansatz in [SOKOLOV und PLEMENOS, 2005] verhindert. Dabei werden zusätzlich die Verhältnisse zwischen benachbarten Polygonen bei der Bewertung des Informationsgehaltes miteinbezogen. Es wird von der Annahme ausgegangen, dass stärker gekrümmte Oberflächenbereiche informativer sind als ebene Bereiche.

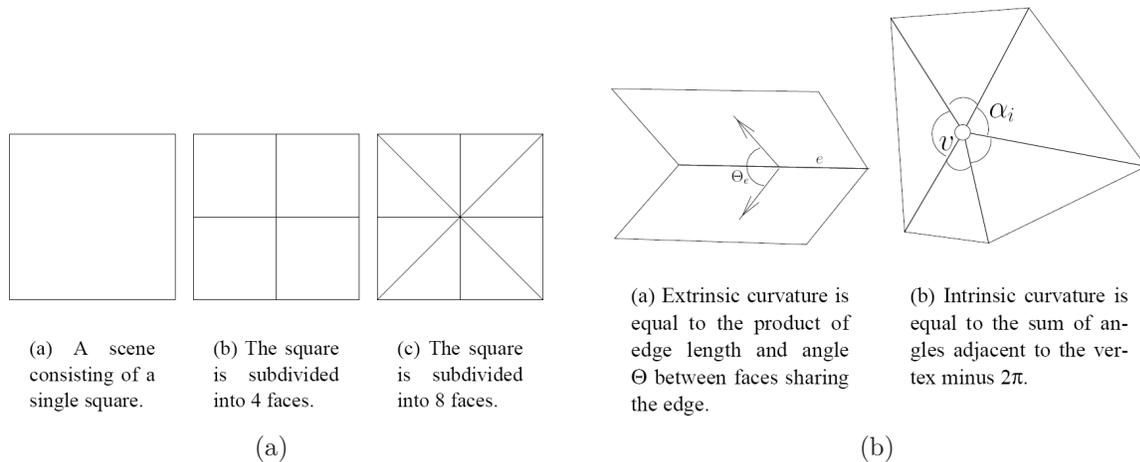


Abb. 2.14: Das Verfahren von VÁZQUEZ et al. [2001] würde für die drei Quadrate in (a) unterschiedliche Informationsgehalte ermitteln. Durch den Einsatz der in (b) gezeigten Maße für die Krümmung kann dies verhindert werden. Quelle: [SOKOLOV und PLEMENOS, 2005]

Um Aussagen über die Krümmungen treffen zu können, wird an Kanten die extrinsische und an Eckpunkten die intrinsische Krümmung ermittelt (siehe Abb. 2.14(b)). Bei dem oben erläuterten Beispiel liefert eine auf der Krümmung basierende Bewertung des Informationsgehaltes für beide quadratischen Flächen identische Werte, da die Winkel zwischen den Polygonen und somit auch die darauf basierende Krümmung in beiden Fällen gleich null ist.

Die Szene wird in den beiden Verfahren als Einheit aufgefasst. Deshalb können keine Aussagen über den Informationsgehalt einzelner darin enthaltener Objekte getroffen werden. Somit ist es nicht möglich, die Güte einer Kameraposition bezüglich eines einzelnen Objektes zu bewerten. Hierzu sind nicht direkt aus der Szenengeometrie ableitbare Metainformationen notwendig, welche die Szene semantisch in einzelne Objekte unterteilen. Ansätze, welche auf derartigen Informationen eine Bewertung des Informationsgehaltes durchführen, sind der Klasse der *High-Level*-Verfahren zuzuordnen.

In [SOKOLOV et al., 2006] wird jedem Objekt der Szene ein sogenannter Erkennbarkeitswert zugewiesen. Dieser ist ein Maß dafür, wie viel Oberfläche von einem Objekt sichtbar sein muss, um es zu erkennen. Bei einem hohen Erkennungswert muss nur ein kleiner Teil des Objektes sichtbar sein, während dem Nutzer bei schlecht erkennbaren Objekten ein großer Teil der Oberfläche präsentiert werden muss.

Allerdings ist ein solcher Erkennbarkeitswert ein subjektives Maß, da er z. B. von der Vorbildung des Betrachters abhängig ist. Hier zeigt sich, dass *High-Level*-Ansätze keine generellen Aussagen über die Güte einer Kameraposition treffen können. Sie sind von dem Aufgabengebiet abhängig, in dem sie angewendet werden. Ein Beispiel für eine solche Spezialisierung ist der auf die Moleküldarstellung fokussierte Ansatz in [VÁZQUEZ

et al., 2006]. Dabei wird davon ausgegangen, dass Informationen durch die Färbung und Größe der dargestellten Atome, der Anzahl der sichtbaren Verbindungselemente und die Anordnung der Moleküle im Raum vermittelt werden. Anhand einer darauf bezogener Bewertung des Informationsgehaltes und der Anwendung von Low- und Mid-Level-Ansätzen lassen sich Kamerapositionen ermitteln, welche viele Informationen über das Molekül vermitteln (siehe Abb. 2.15(a)).

Ein auf Volumendaten anwendbarer *High-Level*-Ansatz wird in [VIOLA et al., 2006] präsentiert. Je nach Art der Anwendung werden den Voxeln zusätzlich Wichtigkeiten zugeordnet, um den Fokus auf bestimmte Teilstrukturen im Datensatz zu lenken (siehe Abb. 2.15(b)).

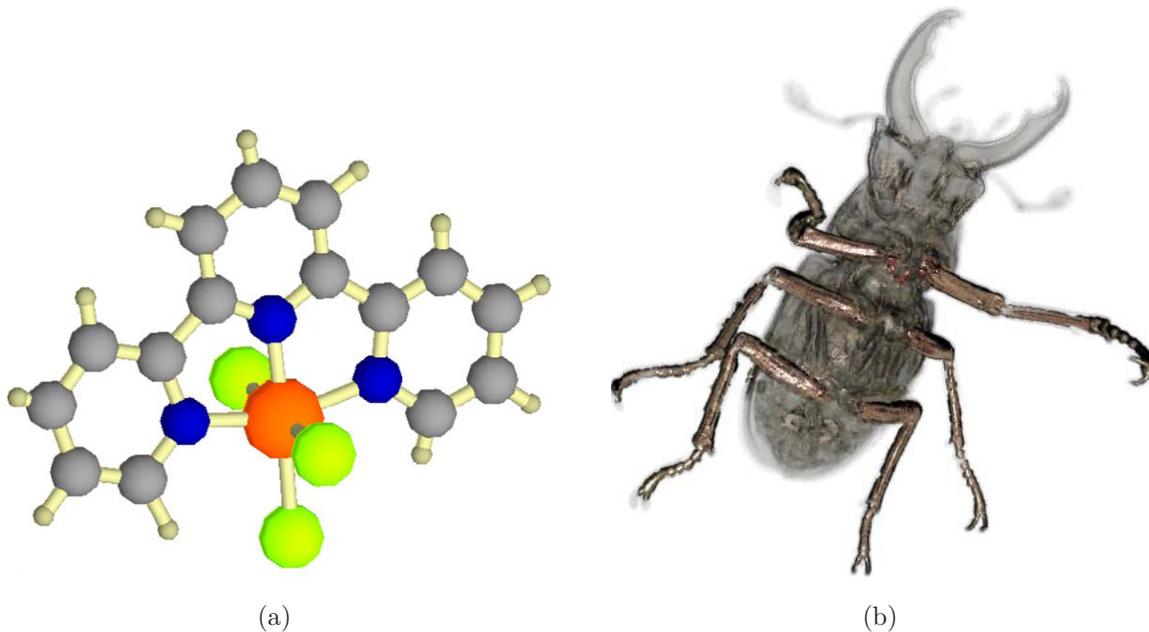


Abb. 2.15: Zwei Beispiele für *High-Level*-Anwendungen. In (a) wurde die Kamera so positioniert, dass alle für die Molekülerkennung wichtigen Informationen sichtbar sind. In (b) wurden die Voxel, welche die Beine des Hirschkäfers repräsentieren als wichtig eingestuft. Quellen: (a) [VÁZQUEZ et al., 2006], (b) [VIOLA et al., 2006]

Für die Bewertung von Kamerapositionen in medizinischen Visualisierungen ist ein *High-Level*-Verfahren notwendig. Derselbe Datensatz kann unter völlig verschiedenen Gesichtspunkten exploriert werden. Die Güte einer Kameraposition ist direkt davon abhängig. Das Finden geeigneter Parameter für die Bewertung einer Kameraposition ist eine Problematik, mit der sich im Rahmen dieser Arbeit auseinandergesetzt wurde.

Im nächsten Abschnitt werden Verfahren vorgestellt und klassifiziert, welche verschiedene Ansätze verfolgen, um auf der Basis der Positionsbewertung eine günstige Kameraposition zu finden.

2.4.3 Systeme zur Kameraplanung

Unter dem Begriff *Virtual Camera Planner* (VCP) sind verschiedenartige Verfahren zusammengefasst, die halb oder vollautomatisch eine Kameraposition ermitteln oder einen Kamerapfad planen [CHRISTIE et al., 2005] und bei unterschiedlichen Anwendungen zum Einsatz kommen. Dazu gehört unter anderem die Auswahl günstiger Kamerapositionen im Bereich der automatische Objekterkennung [DEINZER et al., 2000] oder für die vollständige bildbasierte Erfassung von dreidimensionalen Szenen [STUERZLINGER, 1999]. Neben diesen komplett automatischen Anwendungen ist die Planung von günstigen Kamerapositionen für die Unterstützung im Bereich der nutzerseitigen Exploration dreidimensionaler Szenen ebenfalls von großem Interesse und wird im Rahmen dieser Arbeit behandelt.

Ein VCP passt verschiedene Parameter so an, dass dieses Ziel erreicht oder sich ihm zumindest angenähert werden kann. CHRISTIE et al. [2005] unterteilen diese in kamera- und bildschirmbezogenen Parameter. Kamerabezogene Parameter sind die Position, die Orientierung und die Bewegung der Kamera im Raum. Bildschirmbezogenen Parameter beziehen sich darauf, was dem Nutzer ausgehend von einer bestimmten Kameraposition visuell präsentiert wird. Neben direkt aus dem Viewport ableitbaren Informationen, wie z. B. Position oder Größe der Objekte auf dem Bildschirm, kann auch die bereits beschriebene *Viewpoint Entropy* dieser Parameterklasse zugeordnet werden. Sie beschreibt wie informativ die Szene ist, welche sich dem Nutzer auf dem Bildschirm bietet.

Wie gut ein Objekt auf dem Bildschirm sichtbar ist, ist ein grundlegender Parameter für alle VCP. Wie bereits erläutert, kann sich die dafür notwendige Analyse der Sichtbarkeitsverhältnisse abhängig von der geforderten Genauigkeit als sehr komplex und aufwändig erweisen. Einige Ansätze wie [BLINN, 1988] oder darauf aufbauende Arbeiten [BUTZ, 1997; CHRISTIANSON et al., 1996] vereinfachen die Ermittlung der Sichtbarkeitsverhältnisse, indem Szenenobjekte durch einfache geometrische Strukturen, wie z. B. Punkte, abstrahiert werden.

Diese Art der Vereinfachung ist für die Anwendung auf medizinischen Visualisierungen ungeeignet. Selbst der Einsatz von komplexeren Abstraktionen, wie *Bounding Boxes* oder *Bounding Spheres*, liefert in vielen Fällen eine ungenügenden Genauigkeit, da die Szenen meist sehr komplex gestaltet sind und durch die Abstraktion keine genaue Aussage über die gegenseitige Verdeckung geliefert werden kann (z. B. Gefäßbaum - sehr große *Bounding Box*, entspricht nicht der verursachten Verdeckung).

Anhand von n Parametern können Bedingungen $b_i(x)$, $i = 1 \dots n$ formuliert werden, welche eingehalten werden müssen. Mithilfe dieser Bedingungen lässt sich eine Zielfunktion $f(x)$ definieren, welche maximiert bzw. minimiert werden muss. Grundsätzlich muss der durch die Menge möglicher Kamerapositionen definierte Lösungsraum L nach derjenigen Position x durchsucht werden, welche die Zielfunktion entsprechend den Anforderungen

maximiert bzw. minimiert und dabei die festgelegten Bedingungen einhält ($b_i^{\min} \geq b_i(x) \geq b_i^{\max}$).

$$\begin{aligned} f(x) &= \sum_{i=1}^n b_i(x) & , \text{ fuer } & b_i^{\min} \geq b_i(x) \geq b_i^{\max}, i = 1 \dots n, x \in L \\ f(x) &= \theta & , \text{ sonst} & \end{aligned} \quad (2.1)$$

Die auf diesem Ansatz basierenden Verfahren unterscheiden sich in der Art und Weise, wie mit zu stark oder zu schwach eingegrenzten Fällen umgegangen wird, wie vollständig die Suche nach der Lösung durchgeführt wird und welche Optimierungsverfahren bei der Lösungsfindung zum Einsatz kommen. Nach [CHRISTIE et al., 2005] ergeben sich somit zwei Klassen solch bedingungsbasierter Systeme: vollständige und unvollständige (siehe Abb. 2.16).

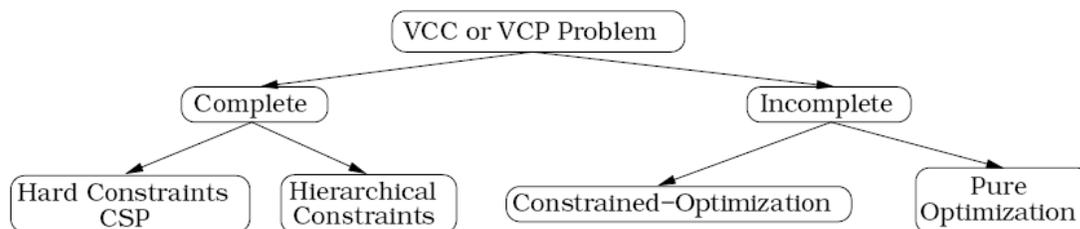


Abb. 2.16: Die Klassifizierung der verschiedenen Verfahren zur Lösung von VCC(*Virtual Camera Composition*)- und VCP(*Virtual Camera Planning*)-Problemen. Quelle: CHRISTIE et al. [2005]

Vollständige Verfahren. Vollständige Verfahren liefern als Ergebnis alle möglichen Lösungen oder keine, wenn die gesetzten Bedingungen nicht erfüllt werden können. In [CHRISTIE und LANGUÉNOU, 2003] wird ein vollständiges, bedingungsbasiertes Verfahren zur Planung von Kamerapfaden vorgestellt. Die Kamerapfade werden durch eine Sequenz verschiedener Bewegungsmodi der Kamera beschrieben (*Hypertubes*). Dieser sequenzbasierte Ansatz ermöglicht die Einhaltung der Bedingungen während der kompletten Animationssequenz, während herkömmliche Verfahren dies meist nur an bestimmten Punkten auf dem Pfad gewährleisten können.

Der Nachteil der Vollständigkeit ist der hohe Rechenaufwand. Des Weiteren wird im Falle der Nichterfüllung der Bedingungen kein Ergebnis geliefert. Da der Lösungsraum vollständig durchsucht wird, liefert dies zwar einen Beleg dafür, dass es im Sinne der Bedingungen keine optimale Lösung gibt, eine weniger optimale, aber dafür durchführbare Lösung wird jedoch nicht geliefert. Diesem Nachteil kann durch die gezielte Relaxation von Bedingungen entgegengewirkt werden. Das in [BARES et al., 1998] vorgestellte System CONSTRAINTCAM nutzt die Technik der teilweisen Erfüllung von Bedingungen. Wenn unter Berücksichtigung aller Bedingungen keine Lösung gefunden

werden kann, werden als schwach eingestufte Bedingungen erweitert oder vollständig ignoriert. Des Weiteren kann die Problematik aufgespalten und eine Menge von Kamerapositionen erzeugt werden, welche unter Berücksichtigung der Bedingungen eine alternative Lösung für die Informationsgewinnung darstellen.

Unvollständige Verfahren. Unvollständige Verfahren liefern eine Lösung bzw. eine Approximation der Lösung, wobei es möglich ist, dass noch weitere Lösungen vorhanden sind. Sie basieren meist auf einer stochastischen Untersuchung des Lösungsraumes, indem Verfahren wie lokale Suchen, genetische Algorithmen oder Simulationen (z. B. *Simulated Annealing*) zum Finden einer günstigen Kameraposition verwendet werden.

CAMDROID [DRUCKER und ZELTZER, 1995] kapselt die unterschiedlichen Aufgaben im Bereich der Kamerainteraktion in sogenannte *Camera Modules* (CM). Durch die Möglichkeit der Sequenzierung und Programmierung wird ein Framework geschaffen, welches die Kontrolle der Kamera in viele Arten von virtuellen Umgebungen ermöglicht. Die in den CMs kodierten Bedingungen werden kombiniert und an ein Optimierungsmodul übergeben. Dieses basiert auf der CSFQP - Bibliothek [DRUCKER und ZELTZER, 1994], welche für die Lösung von bedingungsbasierten, nicht-linearen Optimierungsproblemen ausgelegt ist.

BARES ET AL. (CONSTRAINTCAM) schlagen in [BARES et al., 2000] ein heuristische Suchmethode vor, welche die Grenzwerte (minimal - maximal) vordefinierter Bedingungsbereiche verwendet, um den Lösungsraum möglicher Kameraparameter zu verkleinern. Da unvollständige Verfahren den Lösungsraum nicht komplett durchsuchen, sind die Wahl der Ausgangswerte und die Wichtung der einzelnen Parameter der Optimierungsverfahren und stochastischen Untersuchungsmethoden entscheidend für die Qualität der Ergebnisse. Häufig ist eine feine, an die jeweilige Aufgabenstellung angepasste Abstimmung der jeweiligen Lösungsverfahren notwendig.

Nutzbarkeit in medizinischen Visualisierungen. Eine durch ein vollständiges Verfahren ermittelte Kameraposition stellt das globale Maximum des Lösungsraumes dar. Wird die Einhaltung der durch die Parameter definierten Bedingungen streng gefordert, kann es passieren, dass keine Position diesen Kriterien entspricht und somit auch keine Lösung geliefert werden kann.

Gerade in den komplexen medizinischen Visualisierungen kann es des Öfteren auftreten, dass für bestimmte Strukturen keine Kameraposition die Forderungen erfüllt. Dies ist z. B. bei einem Tumor innerhalb der Leber und der Forderung nach einer vollständigen Sichtbarkeit des Tumors der Fall. Trotzdem sollte ein automatisiertes Positionierungsverfahren unter diesen Umständen eine Kameraposition liefern, da im Bereich der medizinischen Visualisierung davon ausgegangen werden kann, dass im Fall einer Verdeckung der Blick auf die Fokusstruktur durch entsprechende Hervorhebungstechniken ermöglicht

wird. Ein weniger starrer Ansatz mit der Möglichkeit, die Forderung an die Kameraposition zu relaxieren, ist demzufolge für medizinische Visualisierungen eher geeignet.

Die Kamerapositionierung soll den Nutzer unter anderem bei der interaktiven Exploration unterstützen. Die Gesichtspunkte, unter denen dabei exploriert wird, können sich dynamisch ändern. Ein Verfahren zur Kamerapositionierung muss demzufolge in der Lage sein, schnell und effektiv auf diese Veränderungen zu reagieren. Eine vollständige Untersuchung des Lösungsraumes scheint für eine solch interaktive Anwendung demzufolge weniger geeignet, da sie mit einem hohen rechnerischen Aufwand verbunden ist. Für dieses Aufgabengebiet ist eine unvollständige Suche aufgrund des reduzierten Aufwands sinnvoller.

Allerdings bergen lokale Verfahren immer die Gefahr in sich, dass das globale Maximum nicht gefunden wird und ein lokales Maximum als Ergebnis präsentiert wird. Dies ist abhängig davon, wie günstig die Ausgangsbedingungen für eine unvollständige Suche, z. B. durch eine entsprechende Startkonfiguration der Parameter, gewählt wurden. Die Gesichtspunkte, unter denen die Exploration einer medizinischen Visualisierung stattfindet, lassen sich in Klassen einteilen, z. B. danach, ob die räumlichen Zusammenhänge zwischen den Strukturen von Interesse sind oder eher die Beschaffenheit einer einzelnen Struktur. Aufgrund dieser Einteilung könnten Vorannahmen über potentiell günstige Kamerapositionen getroffen und daraus wiederum eine Startkonfiguration für die lokale Suche abgeleitet werden.

Auch wenn sich aus diesen Überlegungen eine Tendenz in Richtung der unvollständigen, auf relaxierbaren Bedingungen basierende Untersuchung des Lösungsraumes für die Anwendung im Bereich der medizinischen Visualisierung anbietet, ist eine genauere Analyse der sich aus konkreten Fällen ergebenden Anforderungen notwendig.

2.5 Zusammenfassung

Es wurde erörtert, wie die aus verschiedenen, bildgebenden Verfahren resultierenden Informationen visualisiert werden und in welchen Anwendungsbereichen diese Visualisierungen zum Einsatz kommen.

Des Weiteren wurden Probleme aufgezeigt, welche bei der Informationsgewinnung aus dreidimensionalen, medizinischen Visualisierungen auftreten können. Es gibt verschiedene Techniken zur Aufmerksamkeitssteuerung, die es dem Nutzer erleichtern, Informationen aus den häufig sehr komplexen Visualisierungen zu gewinnen. Die automatische Ermittlung günstiger Kamerapositionen ist eine solche Technik. Sie kann genutzt werden, um sicherzustellen, dass die Fokusstruktur sichtbar ist und aus einem für den Nutzer gewohnten Blickwinkel präsentiert wird. Neben dem Einsatz als Hervorhebungstechnik können automatisch ermittelte, informative Kamerapositionen

unter anderem dazu genutzt werden, die recht komplexe Problematik der Steuerung einer virtuellen Kamera in dreidimensionalen Szenen zu erleichtern.

Neben den Grundlagen eines virtuellen Kameramodells wurden verschiedene Verfahren, welche sich mit der automatischen Positionierung einer Kamera in einer virtuellen Umgebung beschäftigen, vorgestellt und klassifiziert. Für medizinische Visualisierungen ergeben sich spezielle Anforderungen an ein Verfahren zur automatischen Kamerapositionierung. Im folgenden Kapitel werden Konzepte entworfen, welche die Analyse der Sichtbarkeitsverhältnisse und darauf aufbauend, die Positionierung der Kamera zum Inhalt haben.

3 Entwurf eines Verfahrens zur automatischen Kamerapositionierung

Im Folgenden werden Verfahren zur automatischen Ermittlung günstiger Kamerapositionen konzipiert. Die Anforderungen für solche Verfahren und die Kriterien für die Bewertung der damit generierten Kameraposition richten sich nach dem Anwendungsgebiet. Deswegen wird zu Beginn eine Anforderungsanalyse durchgeführt, welche sich an typischen Anwendungsszenarien und der später erfolgenden Verwendung der Kamerapositionen orientiert. Darauf folgend wird ein bildbasiertes Verfahren für die Analyse der Sichtbarkeitsverhältnisse vorgestellt, welches sich an Erkenntnissen aus der Anforderungsanalyse. Die aus diesem Verfahren resultierende Datenbasis dient als Grundlage für die eigentliche Positionierung der Kamera. Ein an die Anforderungen angepasstes Verfahren und die im Rahmen dieser Arbeit ermittelten Parameter für die Kamerabewertung werden im dritten Abschnitt dieses Kapitels erläutert.

3.1 Anforderungsanalyse

Die zentrale Aufgabenstellung eines Verfahrens zur automatischen Kamerapositionierung ist es, den Nutzer dabei zu unterstützen, eine Visualisierung zu explorieren. So soll sichergestellt werden, dass keine entscheidenden Strukturen übersehen oder Zusammenhänge zwischen verschiedenen Strukturen falsch eingeschätzt werden. Das Verfahren soll in der Lage sein, dem Nutzer automatisch einen Überblick über die Verhältnisse in einer Visualisierung zu liefern, z.B. in Form von Kamerapositionen für eine Animation, so dass dieser sich nicht mit der komplexen Aufgabenstellung der manuellen Steuerung der Kamera auseinandersetzen muss.

Im Bereich der medizinischen Visualisierung gibt es, bezogen auf die Gesichtspunkte, unter denen sie erstellt und exploriert werden, eine Vielzahl von Anwendungsmöglichkeiten. Deshalb wird beim Entwurf des Verfahrens zur Kamerapositionierung Wert darauf gelegt, die zugrunde liegende Datenbasis möglichst allgemeingültig zu konzipieren. Sie muss dementsprechend Informationen zu allen Strukturen bezüglich ihrer Sichtbarkeit und der gegenseitigen Verdeckung enthalten.

Um das Verfahren an die Gegebenheiten eines speziellen Anwendungsfalles anzupassen, werden verschiedene Parameter bereitgestellt. Durch deren Variation können die

Ergebnisse des Verfahrens beeinflusst werden. So ist es möglich, das Verfahren auf derselben Visualisierung unter verschiedenen Gesichtspunkten anzuwenden und auf die speziellen Eigenschaften einer Visualisierung zu reagieren.

Damit gewährleistet werden kann, dass der Informationsgehalt der Datenbasis ausreichend ist, um adäquate Ergebnisse für den jeweiligen Anwendungsfall liefern zu können, müssen die Charakteristika möglicher Visualisierungen analysiert werden. In Abschnitt 3.1.1 werden exemplarisch die Visualisierungen von drei typischen Anwendungsszenarien untersucht, um das Verfahren zur Generierung der Datenbasis entsprechend der daraus gewonnenen Erkenntnisse konzipieren zu können.

Die Anforderungen an die Datenbasis und das darauf aufbauende Verfahren der Positionsbestimmung hängen nicht nur von der Form der Visualisierung, sondern auch von der Anwendung ab, in welche die Visualisierung eingebunden ist. Die Art und Weise, auf welche die Visualisierung exploriert wird, hat beispielsweise Einfluss auf die notwendige Mächtigkeit der durch das Verfahren zur Verfügung gestellten Ergebnisse und die Laufzeit, in der diese bereitgestellt werden müssen. Aufgrund dessen wird die Anwendung, in der das Verfahren zur automatischen Positionsbestimmung zum Einsatz kommt, in Abschnitt 3.1.2 analysiert.

3.1.1 Analyse der Szenen

Im Folgenden werden die Visualisierungen dreier medizinischer Datensätze aus unterschiedlichen Bereichen unter den Gesichtspunkten der möglichen Anwendungsszenarien, der notwendigen Genauigkeit und dem Szenenaufbau untersucht. Es wurde sich für die Visualisierungen der Halsregion, der Leber und des Kniegelenkes entschieden (siehe Abb. 3.1). Sie liefern einen Einblick in drei voneinander unabhängige Aufgabenstellungen in verschiedenen Regionen des Körpers. Die in ihnen enthaltenen Strukturen variieren bezüglich ihrer Art (Gefäße, Gewebe, Knochen etc.), ihrer Form (kompakt, verästelt) und der relativen Größenverhältnisse. Fälle, in denen Strukturen zu einem hohen Grad umschlossen sind, treten ebenso auf wie sehr dicht aneinander liegende Strukturen oder gegenseitige Durchdringungen. Somit werden viele Besonderheiten abgedeckt, die auch in anderen Datensätzen auftreten können.

Ein mögliches Anwendungsszenario für die Visualisierung der Halsregion ist die Lymphknotenausräumung, welche bei einem Tumorbefall notwendig sein kann. Dabei ist es wichtig, dass keine Lymphknoten oder Tumore übersehen werden und kritische Abstände zu anderen Strukturen erkennbar sind. Da die Lymphknoten im Vergleich zu den anderen Strukturen sehr klein sind (siehe Abb.3.2)), müssen die Sichtbarkeits- und Verdeckungsinformationen entsprechend genau sein. Diese Genauigkeit ist durch einen analytischen, objektbasierten Ansatz, oder durch einen bildbasierten Ansatz mit einer ausreichend hohen Auflösung zu erreichen.

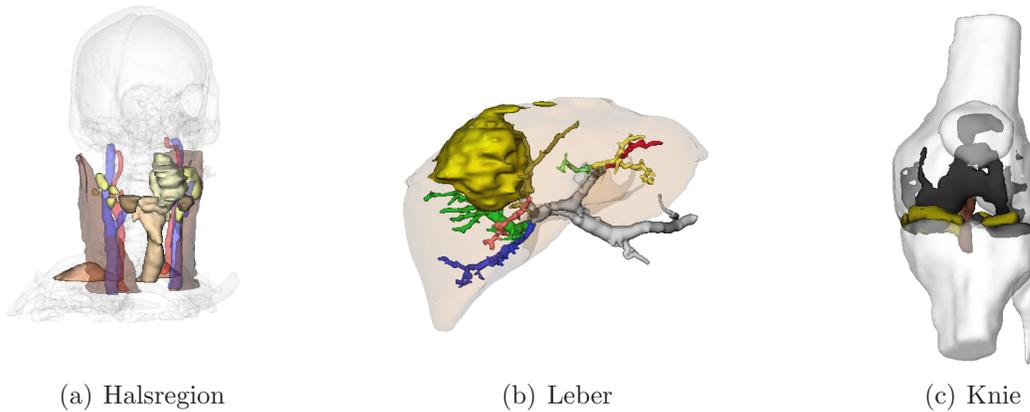


Abb. 3.1: Die Visualisierungen von Halsregion, Leber und Knie. Die Transparenzwerte der Strukturen sind dabei abhängig von ihrer aktuellen Wichtigkeit.

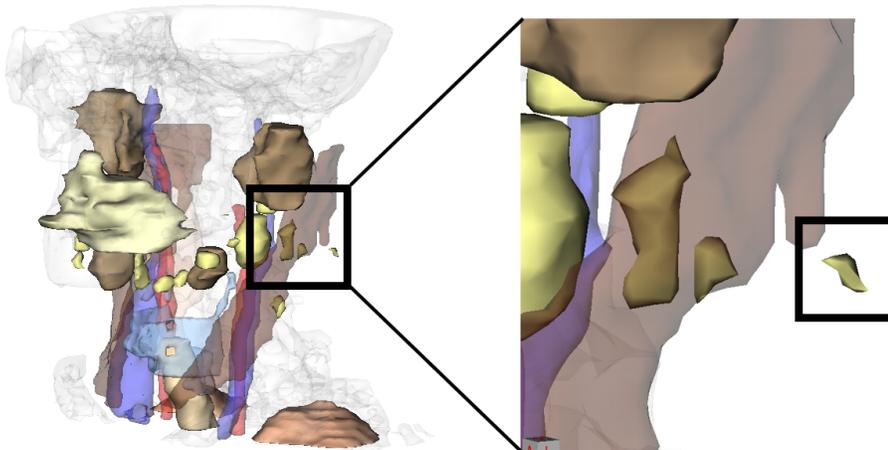


Abb. 3.2: Größenvergleich zwischen einem Lymphknoten und der vollständigen Halsvisualisierung

In der Halsregion sind viele verschiedene Strukturen auf engem Raum angeordnet. Die Halsmuskulatur, versorgende Gefäße und Lymphknoten liegen dicht beieinander und verdecken sich gegenseitig. Durch ihre Nähe zu den Lymphknoten können z.B. Gefäße von Interesse sein, so dass für diese Kamerapositionen ermittelt werden müssen. Ein völlig unverdeckter Blick auf die Gefäße ist unter Umständen nur aus ungewöhnlichen und während einer OP nicht einnehmbaren Positionen möglich, wie z.B. der Blick von „unten“ in den Hals. Hier ergeben sich zwei Problemstellungen. Zum einen ist eine solche Position trotz der vollständigen Sichtbarkeit des Gefäßes nicht sehr aussagekräftig, da man auf den Gefäßdurchschnitt blickt (siehe Abb.3.3), zum anderen entspricht dieser Blickwinkel nicht den Gewohnheiten des untersuchenden Mediziners, wodurch es zu

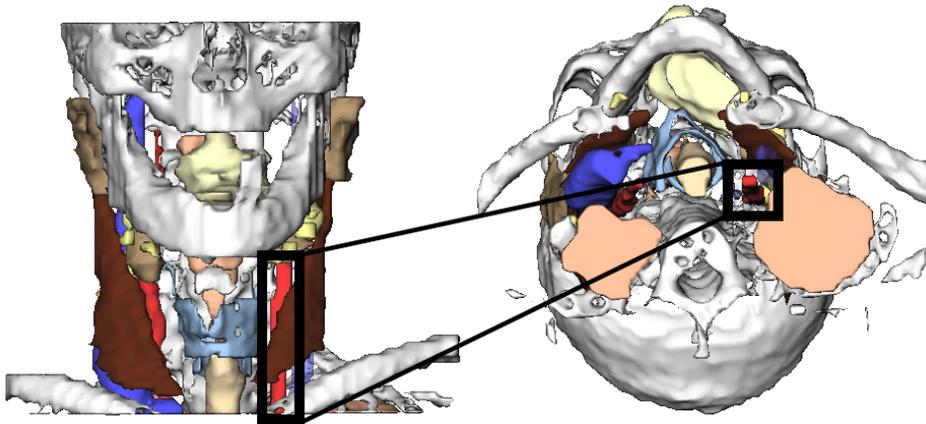


Abb. 3.3: Bei einer opaken Darstellung aller Strukturen ist eine vollständige Darstellung des Gefäßes (Kasten) nur durch den Blick von unten nach oben möglich.

Fehleinschätzungen kommen kann. Es ist also wichtig, dass das Verfahren die Möglichkeit bietet, Aussagen über die Entropie einer Struktur bezüglich einer Kameraposition zu treffen und die Vorzugsblickwinkel für den jeweiligen Anwendungsfall zu definieren.

Wenn ein solcher Vorzugsblickwinkel eingenommen wird, kann es passieren, dass die Zielstruktur von anderen Strukturen zum Teil oder auch vollständig verdeckt wird. Dieser Fall tritt vor allem bei der Visualisierung der Leber häufig auf. Ein möglicher Anwendungsfall ist hier die Planung einer Tumorentfernung. Es ist von Interesse zu erfahren, welche Gefäße wie dicht am Tumor liegen, da bei einer Verletzung der Gefäße die Blutversorgung bestimmter Leberterritorien unterbrochen werden kann. Sowohl der Tumor als auch die Gefäße befinden sich innerhalb der Leber (siehe Abb.3.1(b)). Dies führt dazu, dass sie von fast allen Blickrichtungen aus gesehen von Lebergewebe überdeckt werden. Eine vollständige Ausblendung der Leber würde zwar den Blick auf die interessanten Strukturen freilegen, allerdings würden für die Planung wichtige Kontextinformationen zerstört werden. Um einen differenzierteren Umgang mit solchen Verdeckungen zu ermöglichen, muss das Verfahren Informationen über den Grad der gegenseitigen Verdeckung liefern. So kann bei der Wahl einer dem Vorzugsblickwinkel entsprechenden Kameraposition, eine adäquate Behandlung der Verdeckung durch eine entsprechende Anpassung der Darstellungsparameter der verdeckenden Struktur durchgeführt werden. Wenn in Form von Metainformationen die aktuelle Kontextbedeutung¹ der einzelnen Strukturen vorliegt, sollte diese in die automatische Positionierung der Kamera mit einfließen. So kann vermieden werden, dass die Kamera so positioniert wird, dass wichtige Kontextstrukturen den Blick auf die Zielstruktur verdecken und entsprechend behandelt werden müssen.

¹Die Wichtigkeit der jeweiligen Struktur für das Verständnis der Szenenverhältnisse.

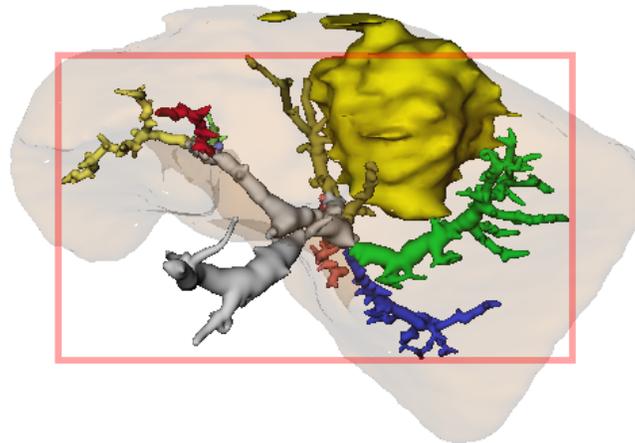


Abb. 3.4: Gerade bei filigranen Strukturen wie Gefäßen, ist eine Approximation durch einfache Hüllkörper für die Sichtbarkeitsanalyse ungeeignet. Aus einer *Bounding Box* (angedeutet durch roten Kasten) würde man eine fast vollständige Verdeckung des Tumors (gelb) durch das Gefäß schlussfolgern, obwohl dieser nur zu einem geringen Grad verdeckt wird.

Das Gefäßsystem der Leber weist die typische Verästelung auf. Bei dieser Art von Struktur ist eine Approximation durch die Verwendung von Hüllkörpern wie *Bounding Box* oder *Bounding Sphere* bei der Sichtbarkeits- und Verdeckungsanalyse nicht ratsam. Die Verdeckung durch den Hüllkörper unterscheidet sich zu stark von der Verdeckung durch die eigentliche Struktur (siehe Abb.3.4). Komplexere Hüllkörper wie *Alpha Shapes* umschließen die Struktur zwar enger, ihre Erzeugung ist allerdings mit einem höheren rechnerischen Aufwand verbunden. Da sie aufgrund ihrer komplexeren Form aus vielen Polygonen bestehen, ist der Geschwindigkeitsgewinn für objektbasierte Verfahren geringer. Es muss abgewägt werden, ob der rechnerische Aufwand der *Alpha Shapes* diesen Geschwindigkeitsgewinn rechtfertigt. Ein objektbasiertes Analyseverfahren sollte demzufolge direkt auf den Polygonen der Struktur oder sehr komplexer Hüllkörper arbeiten und ein bildbasierter Ansatz mit einer entsprechend hohen Auflösung, um alle wichtigen strukturellen Besonderheiten abzubilden.

Ein mögliches Anwendungsszenario für die Visualisierung des Kniegelenks ist die Planung einer Behandlung verletzter Gelenkknorpel. Diese Strukturen befinden sich zwischen den Gelenkknochen und sind nur von wenigen Positionen aus teilweise sichtbar (siehe Abb.3.5). Sie entsprechen zwar der gewohnten Betrachtungssituation, die sich dem Mediziner auch während einer möglichen Operation bietet, allerdings ist der Informationsgehalt von diesen Positionen aus gesehen gering, da nur die Seiten des flachen Knorpels sichtbar sind. Die Vorteile, welche eine freie Exploration einer computergestützten Darstellung mit sich bringen, können nur genutzt werden, wenn vom Vorzugsblickwinkel abgewichen wird und verdeckende Strukturen wie z.B. Knochen in ihrer Darstellung entsprechend angepasst werden. Hieraus ergibt sich die Anforderung,

dass die automatisch generierte Kameraposition an solche speziellen Problemstellungen flexibel angepasst sein sollte.

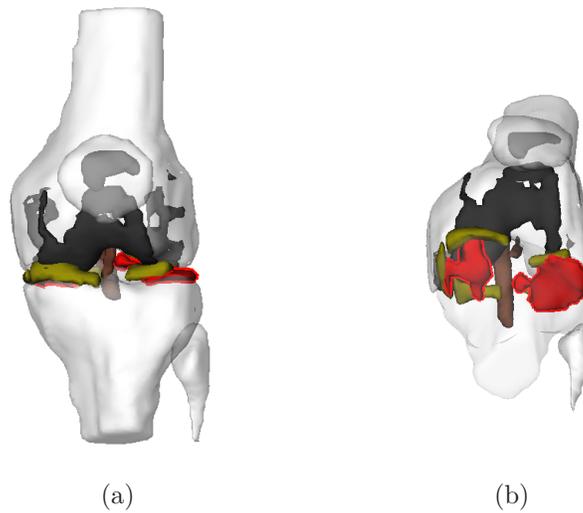


Abb. 3.5: Während die Gelenkknorpel (rot - rot umrandet) der *Tibula* in (a) kaum zu erkennen sind, können durch das Ausweichen in eine während einer OP nicht einnehmbaren Betrachterposition in (b) Aussagen über Form und Größe der Knorpel getätigt werden.

Alle hier beschriebenen Szenen bestehen aus polygonalen Modellen. Die Komplexität ist zum einen durch die Verfahren der Modellerzeugung (z.B. *Marching Cubes*), zum anderen durch Anforderungen bezüglich der Genauigkeit der Modelle sehr hoch (z.B. Hals-Visualisierung: 139536 Dreiecke, Leber-Visualisierung: 360524 Dreiecke). Dadurch ergibt sich für objekt- und somit polygonbasierte Verfahren der Sichtbarkeitsanalyse ein sehr hoher zeitlicher und rechnerischer Aufwand.

3.1.2 Analyse der Anwendung

Das im Rahmen dieser Arbeit entwickelte Verfahren zur automatischen Kamerapositionierung wird in ein System eingebettet, welches sich bei der Exploration der Visualisierungen der „scene in hand“-Metapher bedient. Diese Steuerungsmetapher ist günstig, da kompakte medizinische Visualisierungen exploriert werden sollen und die Gefäßexplorationen keine Rolle spielt. Aufgrund der sich daraus ergebenden Vorteile wird bei der Darstellung der Szenen eine orthogonale Projektion verwendet. Da so die Nähe der Kamera zur Szene keinen Einfluss auf die Sichtbarkeitsverhältnisse hat², kann die Menge möglicher Kamerapositionen auf eine die Szene umgebende Kugel eingegrenzt werden.

²Es wird dabei davon ausgegangen, dass die Kamera immer ausserhalb der Szene positioniert wird und sich somit keine Strukturen „hinter“ der Kamera befinden.

Der Nutzer ist nicht ausschließlich an die automatisch erzeugten Kamerapositionen gebunden. Von diesen Positionen ausgehend, ist zu Zwecken der Orientierung mit einer nutzerseitigen Manipulation der Kameraposition zu rechnen. Demzufolge müssen die ermittelten Kamerapositionen eine bestimmte Stabilität aufweisen. Dies bedeutet, dass kleine Änderungen der Kameraposition nur einen geringfügigen Einfluss auf die Qualität der sich dem Nutzer bietenden Sicht haben sollten.

Aus dieser Forderung nach Stabilität ergibt sich, dass nicht für jede mögliche Position auf der die Szene umgebenden Kugel Sichtbarkeits- und Verdeckungsinformationen vorliegen müssen. Es kann auf eine analytisch genaue Analyse verzichtet werden, da ab einer geringen Größenordnung Fluktuationen, bezogen auf die Qualität der Sicht, für das Verfahren keine Rolle mehr spielen. Darauf basierende Kamerapositionen würden der Forderung nach Stabilität nicht genügen. Bei einer Approximation über die Kugeloberfläche muss jedoch ein Maß für eine ausreichende Auflösung ermittelt werden.

Die automatische Positionierung der Kamera soll in Echtzeit erfolgen, da sie die nutzerseitige Exploration unterstützen soll und demzufolge dynamisch anpassbar sein muss. Die Ausgangsposition der Kamera sollte bei der automatischen Bestimmung einer neuen Kameraposition mit einfließen. Je näher die neue Position an der Ausgangsposition liegt, desto weniger muss sich der Nutzer neu orientieren. Die Gefahr einer Fehleinschätzung wird verringert. Es ist davon auszugehen, dass die Zielstruktur sich während der Laufzeit beliebig ändern kann, da bei einer Exploration häufig mehrere Strukturen und deren Zusammenhänge von Interesse sein können.

Die Visualisierungen sind durchgehend statischer Natur. Da sich demzufolge während der Laufzeit die geometrischen Verhältnisse nicht ändern, ist es möglich darauf basierende Informationen, wie die Sichtbarkeitsverhältnisse, in einem Vorverarbeitungsschritt zu ermitteln.

3.1.3 Zusammenfassung und Schlussfolgerung

Aus der Analyse der Szenen und der Anwendung haben sich zwei Hauptanforderungen ergeben. Das Verfahren muss ausreichend genau sein und Ergebnisse in Echtzeit liefern. Diese beiden Forderungen konkurrieren miteinander, da eine hohe Genauigkeit einen rechnerischen Aufwand bedeutet, der für eine Echtzeitanwendung nicht förderlich ist. Da jedoch eine Vorverarbeitung möglich ist, können statische Informationen in einem Vorverarbeitungsschritt ermittelt und in einem zweiten Schritt dynamisch in Echtzeit ausgewertet werden, um zu einem endgültigen Resultat zu kommen. Demzufolge bietet sich eine Zweiteilung des Verfahrens zur automatischen Positionierung der Kamera an.

In einem ersten Schritt müssen die Sichtbarkeitsverhältnisse der Szene, ausgehend von allen potentiellen Kamerapositionen, ermittelt werden. Hierbei kann von einer endlichen Menge an Positionen ausgegangen werden, da aufgrund der Stabilitätsforderung eine

Approximation der Lösungsmenge möglich ist. Aufgrund der Tatsache, dass während der Laufzeit Informationen zu beliebigen Strukturen von Interesse sein können, müssen für jede potentielle Kameraposition Informationen zu jeder Struktur vorliegen. Diese Informationen lassen sich folgendermaßen einteilen:

- **Sichtbarkeit:** Die aus der Szenengeometrie resultierenden Sichtbarkeitsverhältnisse stellen, bezogen auf eine bestimmte Zielstruktur, einen grundlegenden Faktor bei der Bewertung der Kameraposition dar. Als Maß für die Sichtbarkeit kann der prozentuale Anteil der tatsächlich projizierten Fläche der jeweiligen Struktur, bezogen auf die Gesamtfläche bei völlig verdeckungsfreier Projektion, genutzt werden.
- **Entropie:** Darunter wird im Allgemeinen der Informationsgehalt verstanden. Wie in dem Beispiel der Halsgefäße gezeigt wurde, garantiert vollständige Sichtbarkeit nicht, dass ausreichende Informationen zu einer Zielstruktur gewonnen werden können. Demzufolge muss die Datenbasis zusätzliche Informationen enthalten, welche diesen Informationsgehalt, ausgehend von einer bestimmten Betrachtungsrichtung, beschreiben. Welche Parameter zur Beschreibung des Informationsgehaltes genutzt werden, ist stark von der aktuellen Aufgabenstellung, unter deren Gesichtspunkten die Visualisierung exploriert wird, abhängig. Als solche Parameter können die projizierte Fläche, aber auch objektbezogenen Faktoren wie die Oberflächenkrümmung oder Metainformationen herangezogen werden.
- **Verdeckung:** Eine nicht vollständige Sichtbarkeit der Zielstruktur tritt in den kompakten und aus vielen verschiedenen Strukturen aufgebauten medizinischen Visualisierungen häufig auf. In diesem Fall ist es von Interesse, welche Strukturen zu welchem Grad an der Verdeckung beteiligt sind. Diese Informationen bieten eine Entscheidungsbasis, welche bei der Behandlung von Verdeckungen durch Anpassung der Darstellung genutzt werden kann.

Obwohl es sich bei der Ermittlung dieser Informationen um einen Vorverarbeitungsschritt handelt, ist die Laufzeit von Interesse. Die Ergebnisse sollten möglichst schnell berechnet werden. Bei einem analytischen, polygonbasierten Ansatz kann sich die Ermittlung der Informationen aufgrund der hohen Komplexität der Modelle sehr rechenintensiv und demzufolge zeitaufwändig gestalten. Aufgrund der meist hardwarebasierten Darstellung der Modelle hängt die Laufzeit bei einem bildbasierten Verfahren primär von der Auflösung und nicht von der Komplexität des Modells ab. Allerdings muss hier eine angemessen hohe Auflösung gewählt werden, so dass keine wichtigen Details verloren gehen. Des Weiteren ist eine Diskretisierung des Lösungsraums notwendig, da eine Projektion der Geometrie, ausgehend von einer eindeutigen Betrachterposition, notwendig ist.

Aus der Analyse der Szenencharakteristika und der durch die Anwendung gegebenen Bedingungen ergeben sich folgende Faktoren, welche bei der Positionierung der Kamera in Betracht gezogen werden müssen.

- **Sichtbarkeit der Zielstruktur:** Im Optimalfall sollte die resultierende Position so gewählt werden, dass die Zielstruktur sichtbar ist. Im Falle einer Verdeckung muss sichergestellt werden, dass nur kontextuell unwichtige³ Strukturen daran beteiligt sind, da ihre Darstellungsparameter bei der Behandlung der Verdeckung so angepasst werden, dass ein Informationsgewinn aus der Zielstruktur möglich ist. Dies hat eine Abschwächung des Informationsgehalts der verdeckenden Strukturen zur Folge. Wären diese kontextuell bedeutsam, würden so Informationen verloren gehen, welche zur korrekten Einschätzung der Verhältnisse wichtig sind.
- **Vorzugsblickwinkel:** Der Blick auf die Zielstruktur sollte aus einer für den Nutzer gewohnten Blickrichtung erfolgen. Dadurch wird es ihm erleichtert, korrekte Schlüsse aus den dargebotenen Informationen zu ziehen. Die Definition für einen solchen Vorzugsblickwinkel ist vom spezifischen Anwendungsfall abhängig. So kann er sich an der Position während einer Operation oder auch an medizinischen Standardblickrichtungen (*cranialis* - kopfwärts, *caudalis* - steißwärts, *dorsalis* - rückenwärts etc.) orientieren.
- **Nähe zur Ausgangsposition:** Durch das Miteinbeziehen der Ausgangsposition bei der Berechnung der neuen Kameraposition kann zu einem gewissen Grad vermieden werden, dass diese Position sehr stark von der Ausgangsposition abweicht. Dadurch wird Orientierungsschwierigkeiten entgegengewirkt, welche sich bei einer völlig neuen Kameraposition für den Nutzer ergeben.
- **Stabilität:** Die Ergebnisposition sollte so geartet sein, dass eine kleine Abweichung von dieser Position nur eine kleine Änderung der Sichtbarkeitsverhältnisse zur Folge hat, da zu erwarten ist, dass der Nutzer nicht auf der ermittelten Position verharrt, sondern zur weiteren Exploration auch leicht von dieser abweicht.

Anhand dieser Faktoren können Parameter entwickelt werden, welche als Qualitätsmaß bei der Bewertung einer potentiellen Kameraposition genutzt werden. Diese Parameter sind als grundlegende Parameter anzusehen, da sie, basierend auf der Szenen- und Anwendungsanalyse, für eine große Bandbreite an Anwendungsfällen gültig sind. Es ist jedoch denkbar, dass eine spezielle Problemstellung die Einführung weiterer Parameter voraussetzt, um potentielle Kamerapositionen adäquat zu bewerten und ein korrektes Ergebnis zu liefern. Demzufolge sollte das Verfahren zur automatischen Kamerapositionierung möglichst einfach um zusätzliche Parameter erweiterbar sein.

Die Berechnung der Kamerapositionen muss in Echtzeit erfolgen, um den Nutzer bei der Exploration der Visualisierungen entsprechend unterstützen zu können. Da viele Parameter an der Berechnung beteiligt sein können, müssen diese möglichst effizient ermittelt werden.

Im Folgenden wird ein zweiteiliges Verfahren zur automatischen Positionierung der Kamera vorgestellt. Zuerst wird ein Konzept zur Ermittlung aller benötigten Informationen

³Unter Berücksichtigung der aktuellen Aufgabenstellung bei der Exploration der Visualisierung.

erläutert. Danach wird ein auf diesen Informationen basierendes Verfahren zur Berechnung der Kamerapositionen beschrieben.

3.2 Generierung der Datenbasis

Für die Ermittlung der für die automatische Kamerapositionierung benötigten Informationen wird ein bildbasierter Ansatz verwendet. Aus der Anforderungsanalyse ergibt sich, dass eine Diskretisierung des Lösungsraumes möglich ist, da das Stabilitätskriterium keine analytisch genaue Lösung fordert. Der Lösungsraum wird durch die Menge aller potentiellen Kamerapositionen gebildet. Diese Eingrenzung auf eine endliche Menge von Kamerapositionen begünstigt den Einsatz bildbasierter Analysemethoden bei der Ermittlung der Sichtbarkeitsverhältnisse. Für den Einsatz eines bildbasierten Verfahrens spricht ebenfalls die hohe Komplexität der Modelle in medizinischen Visualisierungen. Aufgrund der daraus resultierenden hohen Polygonzahl ist ein objektbasierter Ansatz zur Sichtbarkeitsanalyse mit entsprechend hohem rechnerischem Aufwand verbunden.

Aus der Anforderungsanalyse geht hervor, dass eine Approximation der Strukturen mittels einfacher Hüllkörper wie z.B. eine *Bounding Box* nicht zu empfehlen ist und somit nicht zur Verringerung des Aufwands genutzt werden kann. Algorithmen zur Vereinfachung des polygonalen Gitters sind ebenfalls mit hohem Aufwand verbunden, da sichergestellt werden muss, dass die äußere Form der Strukturen dabei nicht degeneriert.

Für die Genauigkeit des bildbasierten Verfahrens ist die Wahl einer korrekten Viewport-Auflösung⁴ ebenso entscheidend wie die räumliche Verteilung der Kamerapositionen.

3.2.1 Viewport-Auflösung

Ist die Viewport-Auflösung zu gering gewählt, werden bestimmte Details bzw. kleine Strukturen nicht abgebildet. In diesem Fall können keine Aussagen über die Sichtbarkeit und eventuelle Verdeckungen dieser Strukturen getroffen werden.

Um dies zu vermeiden, muss der minimale Durchmesser einer Struktur durch wenigstens einen Pixel auf dem Viewport repräsentiert werden. Dies wird erreicht, wenn die Ungleichung $\frac{H}{i} \leq h$ für alle Strukturen erfüllt ist, wobei H die Ausdehnung des quadratischen Ausschnitts der Projektionsebene, welche auf dem Viewport repräsentiert wird, i die Auflösung des Viewports und h die Länge des minimalen Durchmessers der Struktur beschreibt (siehe Abb.3.6). Der Abstand der Struktur von der Projektionsebene hat aufgrund der orthogonalen Projektion keinen Einfluss auf die Größe ihrer projizierten

⁴Unter dem Begriff Viewport ist im Folgenden das Sichtfeld zu verstehen, auf dem das gerasterte Ergebnis der orthogonalen Projektion auf eine entsprechende, von der aktuellen Kameraposition / -orientierung abhängige, Projektionsebene repräsentiert wird.

Abbildung. Im Grenzfall werden die kleinsten Strukturen nur durch einen Pixel repräsentiert. Dies hat zur Folge, dass nur eine binäre Aussage über deren Sichtbarkeit getroffen werden kann, je nachdem, ob der Pixel abgebildet wird oder nicht. Um differenzierte Informationen über den Grad der Sichtbarkeit zu erhalten, muss eine entsprechend höhere Auflösung gewählt werden.

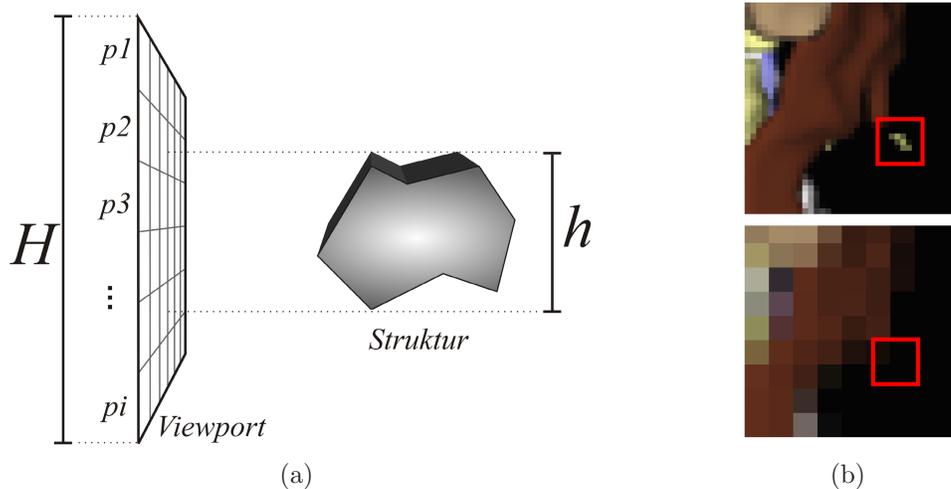


Abb. 3.6: Damit die Struktur in (a) durch wenigstens ein Pixel auf dem Viewport repräsentiert wird, muss die Ungleichung $\frac{H}{i} \leq h$ erfüllt sein. Die Folgen einer zu niedrig gewählten Viewport-Auflösung werden in (b) verdeutlicht: Die Struktur (roter Kasten) wird im unteren Bild nicht mehr repräsentiert. Somit können auch keine Aussagen über entsprechende Sichtbarkeitsverhältnisse getroffen werden.

3.2.2 Verteilung der Kamerapositionen

Wie aus der Anforderungsanalyse hervorgeht, ist es aufgrund der „Scene in Hand“-Metapher möglich, eine endliche Menge von potentiellen Kamerapositionen auf der Oberfläche einer die Szenen umgebenden Kugel zu verteilen. Positionen im Inneren der Kugel sind unnötig, da dies nur eine Abstandsänderung zur Folge hätte, die sich im resultierenden Bild aufgrund der orthogonalen Projektion nicht widerspiegeln würde.

Während die Auflösung des Viewports die Genauigkeit der Sichtbarkeitsinformationen an einer spezifischen Position beeinflusst, ist die Genauigkeit des Gesamtergebnisses direkt proportional zur Anzahl der Kamerapositionen, da diese eine Approximation des Lösungsraumes darstellen. Werden zu wenige Kamerapositionen auf der Kugel verteilt, kann es passieren, dass essentielle Änderungen der Sichtbarkeitsverhältnisse nicht erfasst werden (siehe Abb.3.7). Eine höhere Zahl von Positionen vergrößert den zeitlichen Aufwand der Sichtbarkeitsanalyse. Ein genaues Maß für die notwendige Genauigkeit

ist schwer zu finden, da sie auch von der späteren Verwendung der Ergebnisse bei der Exploration abhängig ist.

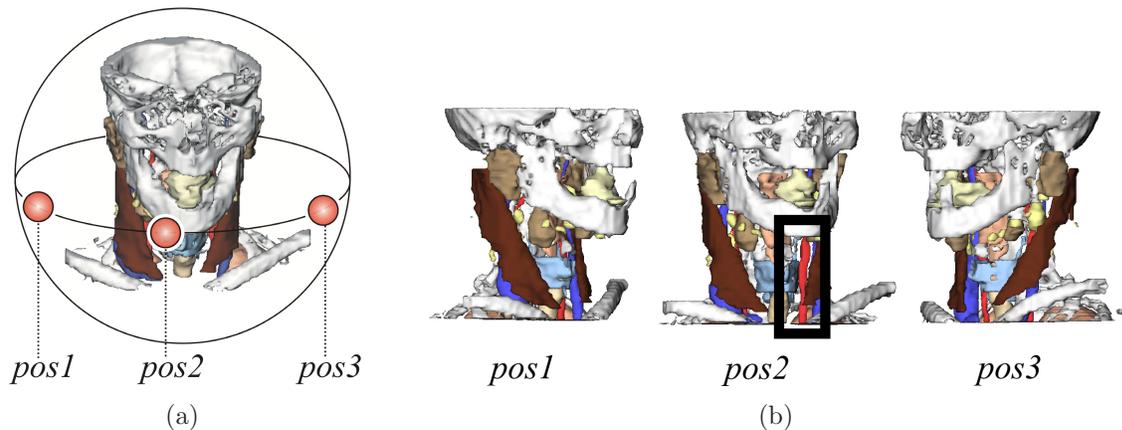


Abb. 3.7: Werden die Sichtbarkeitsverhältnisse nur von *pos1* und *pos3* aus untersucht, wird die gute Sichtbarkeit des Gefäßes (Kasten) von *pos2* aus nicht erfasst.

Allerdings lässt sich eine Heuristik definieren. Befinden sich viele sehr kleine Strukturen in der Szene, kann davon ausgegangen werden, dass sie vergrößert betrachtet werden. Eine an die vergrößerte Darstellung angepasste, manuelle Positionierung der Kamera auf der umgebenden Kugel findet in einem kleineren Maßstab statt. Um diese Navigation adäquat unterstützen zu können, muss die Approximation des Lösungsraumes entsprechend genau sein. Zusätzliche Informationen über die Rotationsgeschwindigkeit der Kamera bei verschiedenen Vergrößerungsstufen sind für eine Einschätzung dieser Genauigkeit notwendig.

Die einfachste Form, die Kamerapunkte auf der Kugel zu verteilen, ist eine **uniforme Anordnung**, orientiert an den Längen- und Breitengraden der Kugeloberfläche. Die Genauigkeit der Approximation steigt mit geringerem Abstand zwischen den einzelnen Positionen. Diese an den Längen- und Breitengraden orientierte Verteilung führt zu einer Häufung der Positionen an den Polen der Kugel (siehe Abb.3.8). Somit werden die Sichtbarkeitsverhältnisse dort genauer beschrieben als in Richtung des Äquators. Die unter Umständen sehr hohe Genauigkeit an den Polen führt zu einem unnötig hohen rechnerischen Aufwand und einer Überbewertung der Polregionen. Dies entspricht nicht der Forderung nach einer allgemeingültigen Basis. Es wird von der Gleichwertigkeit aller Sichtrichtungen ausgegangen. Erst durch eine später erfolgende Wichtung sollen ihnen unterschiedliche Wertigkeiten zugewiesen werden.

Eine Möglichkeit dieses Problem zu behandeln, ist, sich von der Orientierung an den Längen- und Breitengraden zu lösen und eine **gleichmäßige Verteilung** auf der Kugeloberfläche anzustreben. Durch eine fortlaufende Unterteilung der Flächenelemente eines entsprechenden Grundkörpers kommt es zu einer schrittweisen Annäherung

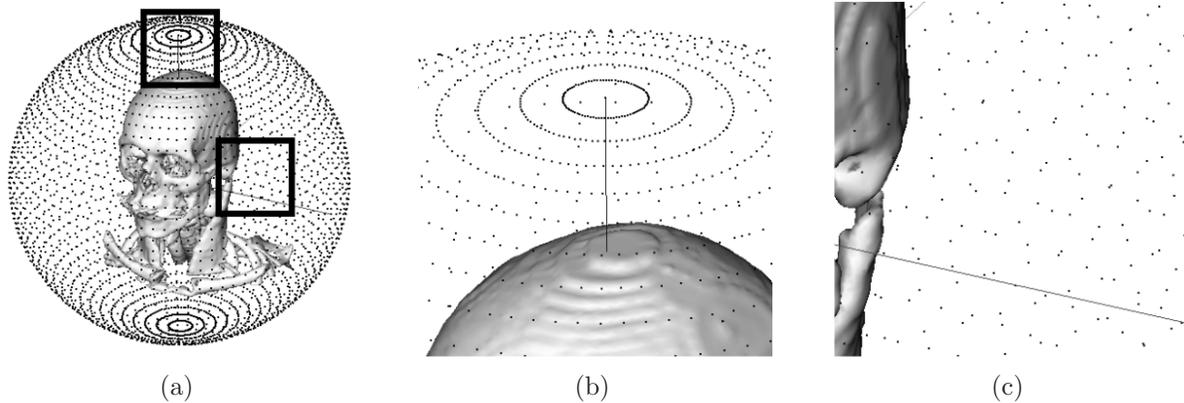


Abb. 3.8: Die Punkte auf der die Szenen umgebende Kugel in (a) entsprechen den Kamerapositionen. In (b) und (c) werden die Ausschnitte aus (a) (Kästen) vergrößert wiedergegeben. Deutlich ist in Polnähe eine viel engere Verteilung der Kamerapositionen zu erkennen, obwohl alle Positionen den selben Abstand von 4 Grad in Richtung der Längens- bzw. Breitengrade besitzen.

an die gewünschte Kugelform (siehe Abb.3.9). Die Position neu hinzukommender Eckpunkte wird so angepasst, dass sie denselben Abstand zum Mittelpunkt der entstehenden Kugelapproximation haben. Ziel ist es, die Flächenelemente möglichst gleichmäßig auf der resultierenden Kugeloberfläche zu verteilen. Dies hat zur Folge, dass an deren Eckpunkte orientierte Kamerapositionen ebenfalls gleichmäßig verteilt sind und es zu keiner Häufung an den Polen der Kugel kommt. Als Grundkörper eignen sich die sogenannten platonischen Körper⁵. Aufgrund ihrer gleichseitigen, dreieckigen Flächenelemente bieten sich Körper wie Tetraeder, Oktaeder oder Isokaeder als Grundkörper an, da eine gleichmäßige Unterteilung der Dreiecke möglich ist und eine Kugeloberfläche relativ genau durch Dreiecke beschrieben werden kann [MIAO-LE, 2004]). Die Genauigkeit der anhand einer solchen Kugel durchgeführten Sichtbarkeitsanalyse steigt, je mehr Unterteilungsschritte durchgeführt werden.

3.2.3 Sichtbarkeitsanalyse

Für jede Kameraposition auf der die Szene umgebenden Kugel werden die Sichtbarkeitsverhältnisse für alle Strukturen der Szenen ermittelt. Ist dieser Schritt vollendet, liegen alle notwendigen szenenbezogenen Informationen vor, um darauf aufbauend eine für den jeweiligen Anwendungsfall günstige Kameraposition zu ermitteln.

⁵Platonische Körper sind regelmäßige konvexe Polyeder, deren Flächenelemente zueinander kongruente, gleichmäßige Vielecke sind, wobei an jedem Eckpunkte dieselbe Anzahl an Flächenelementen aneinander grenzen.

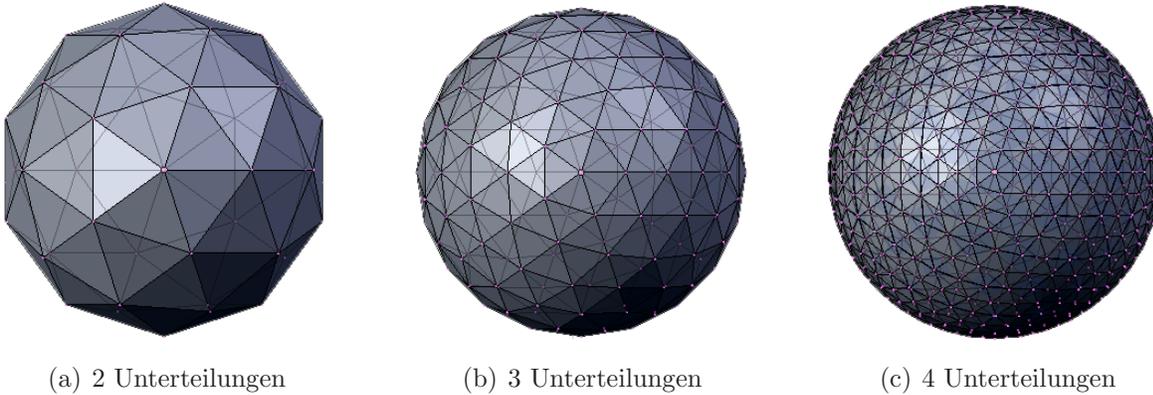


Abb. 3.9: Ausgehend von einem Isokaeder kann sich durch fortlaufende Unterteilung an eine Kugelform angenähert werden. Die Eckpunkte der Dreiecke sind gleichmäßig auf der Kugeloberfläche verteilt. Eine Häufung in Richtung der Pole tritt nicht auf.

Auslesen des z -Buffers

Um Aussagen über den Grad der Sichtbarkeit einzelner Strukturen und den Grad der eventuell auftretenden Verdeckung durch andere Strukturen treffen zu können, wird der z -Buffer genutzt. Dieser wird bei einer hardwarebeschleunigten Darstellung der Szene auf der Grafikkarte schnell und effizient erzeugt. Die im Buffer enthaltenen z -Werte stehen im direkten Zusammenhang zum Abstand der Strukturen von der Projektionsebene z' . Der z -Wert 0 entspricht dem Abstand der *Near Clipping Plane* ($near$) und der z -Wert 1 dem Abstand der *Far Clipping Plane* (far).

$$z = \frac{far + near}{far - near} + \frac{1}{z'} \left(\frac{-2 \cdot far \cdot near}{far - near} \right) \quad (3.1)$$

Die Art und Weise der z -Wert Berechnung hat zur Folge, dass die Tiefenauflösung mit steigender Entfernung von der Projektionsebene sinkt. Dies hängt mit dem primären Anwendungsgebiet des z -Buffers während der Bilderzeugung zusammen. Er dient als Entscheidungsmaß, welche Pixel den geringsten Abstand zum Betrachter haben und dementsprechend dargestellt werden. Vor allem in zum Betrachter nahen Bereichen muss diese Entscheidung möglichst genau sein, um Bildfehler zu vermeiden. Da weiter weg liegende Elemente meist kleiner dargestellt werden, ist dort keine so hohe Genauigkeit notwendig. Deshalb wird der Abstand zwischen *Near* und *Far Clipping Plane* möglichst klein gewählt, aber so, dass der durch sie aufgespannte z -Raum die kompletten Szenen enthält. Für jede einzelne Struktur wird ein z -Buffer erzeugt, der nur sie enthält. Des Weiteren wird ein z -Buffer erzeugt, welcher alle Strukturen enthält (Gesamtbuffer). Dieser dient als Entscheidungsgrundlage, an welchen Pixelpositionen die restlichen z -Buffer ausgewertet werden müssen. So wird vermieden, dass für die Sichtbarkeitsanalyse unwichtige Pixelpositionen mit in den Auswertungsprozess einfließen. Nur dort wo dieser

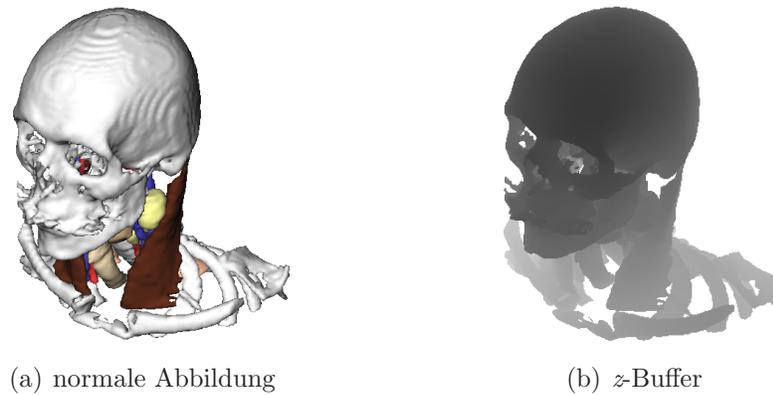


Abb. 3.10: Abbildung (a) zeigt das Bild, wie es dem Nutzer von einer Kameraposition aus präsentiert wird. Der z -Buffer in (b) ermöglicht relative Aussagen darüber, wie dicht ein zu einem Bildpunkt gehörender Strukturbestandteil zur Projektionsebenen liegt (je dunkler, desto dichter).

Buffer einen Wert ungleich dem Hintergrundwert besitzt, liegen relevante Informationen vor.

Auswertung der z -Buffer

Bei n Strukturen in der Szene liegen somit $n + 1$ z -Buffer vor. Ihre Größe entspricht der Auflösung des Viewports. Für jede relevante Pixelposition werden die z -Werte der einzelnen Buffer untersucht (siehe Abb. 3.11). Aus dem Vergleich dieser z -Werte kann eine Aussage darüber getroffen werden, in welcher Reihenfolge, bezogen auf ihren Abstand von der Projektionsebene, die Strukturen auftreten. Darauf aufbauend kann für jede Struktur ermittelt werden, durch welche Struktur sie an der jeweiligen Pixelposition verdeckt wird, bzw. welche Strukturen sie selbst verdeckt. Strukturen, welche an der jeweiligen Pixelposition einen z -Wert gleich dem Hintergrundwert besitzen, nehmen an der Auswertung nicht teil. Sie sind an dieser Pixelposition nicht vertreten und können demzufolge dort auch nicht verdeckt werden oder verdecken. Die notwendige Sortierung nach z -Werten wird mithilfe des Quicksort-Algorithmus durchgeführt. Er hat eine günstige mittlere Laufzeit und neigt nicht dazu, schnell zu entarten. Aus der so sortierten Liste können insgesamt folgende Informationen für die aktuelle Pixelposition abgeleitet werden:

- welche Strukturen an dieser Position bei vollständiger Sichtbarkeit durch ein Pixel repräsentiert werden würde
- ob und durch welche Strukturen ein Pixel einer bestimmten Struktur verdeckt wird
- ein relativer Wert für den Abstand der Struktur vom Betrachter

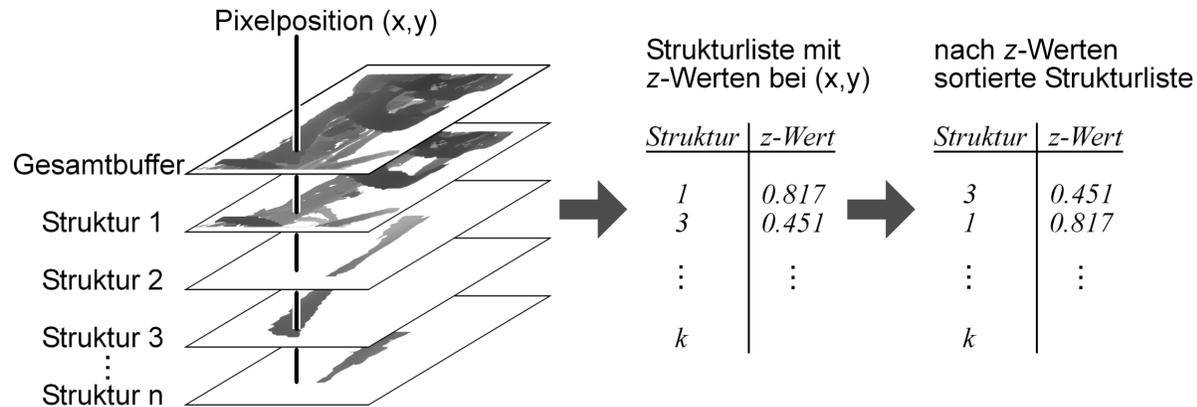


Abb. 3.11: Im Gesamtbuffer wurde an Position (x, y) ein Wert ungleich dem Hintergrundwert gefunden. Deswegen müssen dort Strukturen und evtl. Verdeckungen vorhanden sein. Die z -Werte der einzelnen Strukturen werden ausgelesen und in der Reihenfolge *Nah - Fern* sortiert. In diesem Beispiel nehmen Struktur 2 und Struktur n nicht an der Sortierung teil, da sie an der Position (x, y) nicht repräsentiert werden.

Über die Menge aller Pixelpositionen betrachtet, lassen sich so Aussagen über die Anzahl der Pixel treffen, durch die eine Struktur repräsentiert wird, wie weit diese Pixel vom Betrachter entfernt sind, wie viele davon durch andere Strukturen verdeckt werden und um welche Strukturen es sich dabei handelt.

Aus diesen Informationen lässt sich für eine Struktur der Grad der Sichtbarkeit und die Grade der Verdeckung durch andere Strukturen errechnen. Da auch im Falle einer Verdeckung bekannt ist, ob eine Struktur durch ein Pixel repräsentiert wird, kann ausgesagt werden durch wieviele Pixel es dargestellt werden würde, wenn es nicht zu dieser Verdeckung gekommen wäre. Diese Aussage über die projizierte Fläche kann z.B. als einfaches Entropiemaß genutzt werden, wenn man den Informationsgehalt mit der projizierten Fläche gleichsetzt. Die z -Werte liefern ein relatives Maß für den Abstand der Struktur vom Betrachter, wobei über die Menge aller Pixelpositionen ein Minimal- und ein Maximal-Abstand vorhanden ist.

Die Informationen über die Menge aller Pixelpositionen werden in einer Matrix der Form $n \times (n + 3)$ gespeichert. Der Aufbau dieser Matrix wird in Abbildung 3.12 näher erläutert. Die Datenbasis, auf der im weiteren Verlauf die Berechnung der automatischen Kamerapositionen stattfinden soll, stellt sich so dar, dass für jede Kameraposition auf der Kugel eine solche Matrix vorliegt, welche die Sichtbarkeitsverhältnisse der Szene, von dieser Position aus betrachtet, beschreibt. Der vollständige Ablauf während der Sichtbarkeitsanalyse wird in Abbildung 3.13 schematisch wiedergegeben.

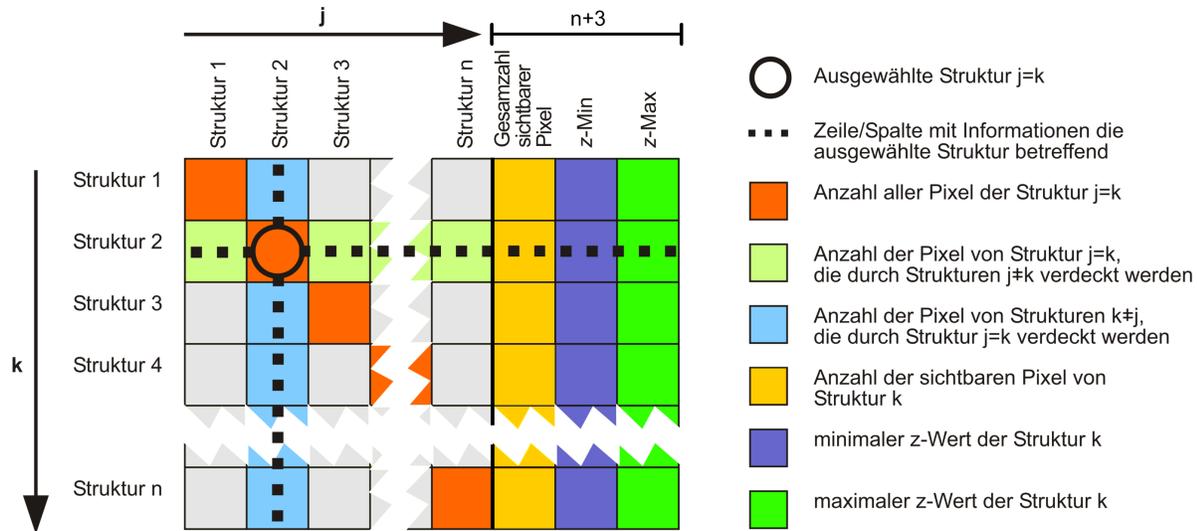


Abb. 3.12: Für die ausgewählte Struktur 2 können entlang der Zeile 2 folgende Informationen aus der Matrix gewonnen werden: aus den Spalten $i = 1 \dots n$ ($i \neq 2$) jeweils die Anzahl der Pixel, welche durch Struktur i verdeckt werden. An der Position 2 steht die Anzahl aller Pixel, durch die Struktur 2 ohne Einbeziehung der Verdeckungen repräsentiert wird. An Position $i = n + 1$ steht die Gesamtzahl der sichtbaren Pixel von Struktur 2. Der minimale und maximale z -Wert, mit dem Struktur 2 dargestellt wird, befinden sich an Position $i = n + 2$ bzw. Position $i = n + 3$. Entlang der Spalte 2 stehen in allen Zeilen $i \neq 2$ die Anzahl der Pixel von Struktur i , welche durch Struktur 2 überdeckt werden.

3.2.4 Skalierbarkeit und Aufwandsbetrachtung

Die Laufzeit der hier beschriebenen Sichtbarkeitsanalyse ist von der Anzahl der Strukturen, der Auflösung des Viewports und der Anzahl der Kamerapositionen abhängig. Um eine höhere Genauigkeit bei der Beschreibung der Sichtbarkeitsverhältnisse zu erreichen, müssen die Kamerapositionen dichter auf der Oberfläche der umgebenden Kugel angeordnet sein, was die Anzahl der benötigten Positionen erhöht. Soll der Abstand zwischen den Kamerapositionen auf der Kugel halbiert werden, vervierfacht sich deren Anzahl.

Ein solch quadratischer Zusammenhang gilt auch bei der Wahl der Viewportauflösung. Je höher diese gewählt wird, desto exakter können die Verdeckungs- und Sichtbarkeitsverhältnisse beschrieben werden. Wird die Auflösung in x - und y -Richtung verdoppelt, vervierfacht sich die Anzahl der Pixelpositionen, die bei der Sichtbarkeitsanalyse abgerastert werden müssen. Die Anzahl n der Strukturen beeinflusst die Laufzeit weniger stark. Bei einer Steigerung der Anzahl müssen mehr Buffer erzeugt werden. Da dies aber auf Basis der Grafikhardware geschieht, ist dieser Einfluss weniger gering, als der Einfluss durch die Sortierung der resultierenden z -Werte. Die Laufzeit des dafür verwendeten Quicksort-Algorithmus kann im Mittel durch $O(n * \log(n))$ beschrieben werden.

Daraus lässt sich schlussfolgern, dass die Genauigkeit der Sichtbarkeitsbeschreibung, sowohl global über alle Kamerapositionen als auch für jede Position einzeln den größten Anteil an der aufzuwendenden Laufzeit hat. Aus der Anforderungsanalyse ergibt sich, dass eine Steigerung der Genauigkeit nur bis zu einem bestimmten Grad sinnvoll ist. Eine entsprechend genaue, vorhergehende Analyse der Verhältnisse in der Szene und des Rahmens, in dem die Datenbasis genutzt wird, ist durchzuführen. Dadurch wird eine unnötig hohe Laufzeit vermieden. Die häufungsfreie Verteilung der Positionen auf der umgebenden Kugel beschleunigt das Verfahren bei gleichbleibender Genauigkeit der Ergebnisse.

Des Weiteren zeigt sich, dass der Algorithmus output-sensitiv ist. Die Laufzeit richtet sich direkt nach der Menge der zu untersuchenden Strukturen und der Genauigkeit der Resultate. Es ist leicht möglich, das Verfahren in Bezug auf Laufzeit und Genauigkeit an sich aus der Aufgabenstellung ergebende Anforderungen anzupassen.

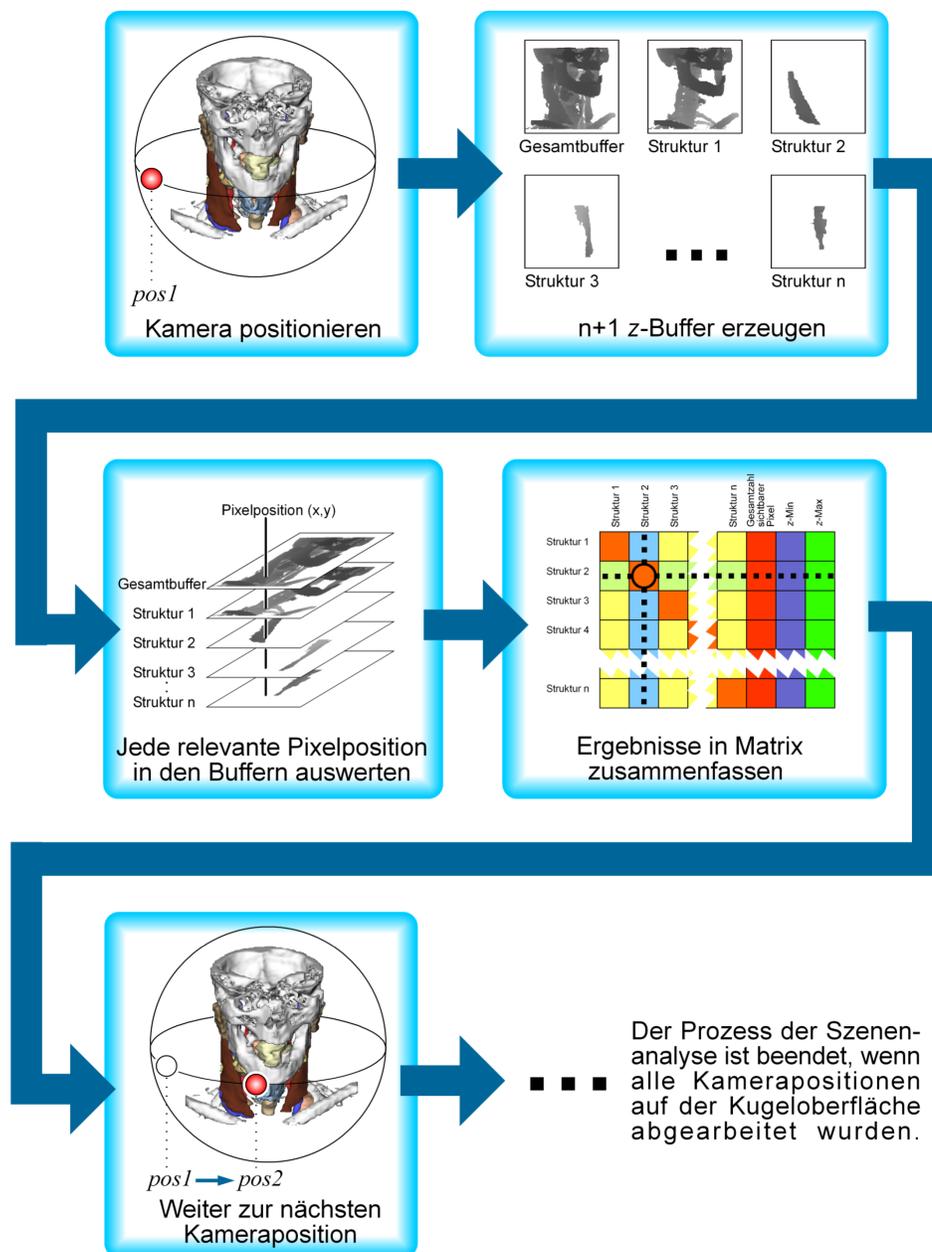


Abb. 3.13: Schematische Übersicht aller Schritte während der Sichtbarkeitsanalyse

3.3 Berechnung der Kamerapositionen

Ausgehend von der Anforderungsanalyse muss das Verfahren in der Lage sein, für jede Struktur der Visualisierung eine informative Kameraposition zu generieren. Die notwendige Qualitätsanalyse potentieller Positionen soll parametrisierbar und somit an

spezielle Anforderungen anpassbar sein. Die dafür genutzten Informationen stammen aus der im Vorverarbeitungsschritt erzeugten Datenbasis (z.B. Sichtbarkeit), aus laufzeitabhängigen Parametern (z.B. Ausgangsposition der Kamera) und fallabhängigen Metainformationen (z.B. Vorzugsblickwinkel).

Wie zuvor erläutert, sind die potentiellen Kamerapositionen auf der Oberfläche der die Szene umgebenden Kugel verteilt. Somit ergibt sich für jeden Bewertungsparameter eine zweidimensionale Funktion der Form $f(a, b) = q$, wobei a und b die diskrete Position auf der Kugeloberfläche in Form von Längen- und Breitengrad eindeutig beschreiben und q das Maß für die Qualität der jeweiligen Position darstellt. Je höher der skalare Wert q ist, desto eher befriedigt die Position die sich durch den jeweiligen Parameter ergebenden Anforderungen. Im Folgenden werden diese Funktionen als **Parameterfelder** bezeichnet.

Für eine einfache Weiterverarbeitung dieser Parameterfelder ist eine gleichmäßige, an den Längen- und Breitengraden orientierte Verteilung der Punkte günstig, da z.B. morphologische Operatoren zum Einsatz kommen. Die oben beschriebene Gleichverteilung der Positionen auf der Kugeloberfläche führt zu einer ungleichmäßigen Verteilung im Parameterfeld. Die durch diese Optimierung entstehenden Freistellen im Parameterfeld werden durch eine Interpolation der vorhandenen Werte gefüllt. Dies ist legitim, da die neu entstehenden Informationen die für die Ergebnisfindung wichtigen Minimal- und Maximalwerte nicht beeinflussen. Des Weiteren liegen sie unter der durch die Positionsdichte auf der Kugel festgelegten Mindestinformationsauflösung.

In das Endergebnis fließen alle Parameter mit ein, indem ihre Parameterfelder normiert und aufsummiert werden. Aus der Anforderungsanalyse ergibt sich, dass nicht immer ein gleichmäßiger Einfluss aller Parameter auf die resultierende Position erwünscht ist. Um zu beeinflussen, wie stark ein Parameter sich auf die Ergebnisposition auswirkt, besteht deswegen zusätzlich die Möglichkeit einer Wichtung. Für alle Parameterfelder p_i und den dazugehörigen Wichtungsfaktoren ω_i ergibt sich das Ergebnis im Summenfeld f an der Längen- und Breitengradposition (α, β) wie folgt:

$$f(\alpha, \beta) = \sum_{i=1}^n \omega_i \cdot p_i(\alpha, \beta) \quad (3.2)$$

Dies ist die allgemeine Form der Berechnung für die Ergebnisfindung. Eine detaillierte, an den im Rahmen dieser Arbeit konzipierten Parametern orientierte Form wird in Abbildung 3.28 zusammen mit einem schematischen Überblick über das Verfahren präsentiert.

Die Position mit dem maximalen Funktionswert in dem sich ergebenden Summenfeld stellt das Ergebnis des Verfahrens dar, da sie alle Parameter entsprechend ihrer Wichtung am ehesten erfüllt. Ausgehend von Informationen über den Radius und den Mittelpunkt der Kugel kann eine Position und Orientierung der Kamera im Raum errechnet und weitergehend genutzt werden.

Diese Art der Positionsbewertung erfüllt die in der Anforderungsanalyse gestellte Forderung nach Erweiterbarkeit. Durch die einheitliche Repräsentation der Parameter in Form von Feldern können zusätzliche Parameterfelder dem Verfahren problemlos hinzugefügt werden, solange sie der Dimension und Orientierung (höherer Wert entspricht positiverer Bewertung) der vorhandenen Parameterfelder entsprechen. Somit kann das Verfahren flexibel an eventuell auftretende Anforderungen angepasst werden, welche nicht durch die grundsätzlich vorhandenen Parameter abgedeckt werden. Auf diese grundsätzlichen, nicht fallspezifischen Parameter und die Erzeugung der zugehörigen Parameterfelder wird im Folgenden näher eingegangen.

3.3.1 Parameter

Um die Qualität einer Kameraposition bewerten zu können, müssen mehrere Parameter beachtet werden. Wird beispielsweise nur die Sichtbarkeit einer Struktur als Parameter verwendet, kann bei völliger Umschlossenheit dieser Struktur keine Entscheidung getroffen werden. Auch wenn dieser Fall nicht eintritt, kann die generierte Kameraposition für das Finden einer informativen Betrachterposition ungünstig sein. So kann die Position derart ungewöhnlich sein, dass die Gefahr einer Fehleinschätzung durch den Betrachter erhöht wird oder die Zielstruktur zwar vollständig sichtbar ist, aber von einer Position aus präsentiert wird, die keine korrekte Aussage über ihre Morphologie zulässt (siehe Abb.3.3).

Die Wahl der im Rahmen dieser Arbeit zur Anwendung kommende Parameter basiert auf der eingangs durchgeführten Anforderungsanalyse. Die dort betrachteten Visualisierungen stellen, bezüglich der Anforderungen an die Parameter, typische Beispiele für die *Scene in Hand* basierte Exploration dar. Im Folgenden dienen Abbildungen der Parameterfelder von Strukturen aus dem Hals-Datensatz zur Verdeutlichung der Charakteristik der einzelnen Parameter. Abbildung 3.14 stellt dabei den räumlichen Zusammenhang zwischen den Feldern und der Szenengeometrie dar. Des Weiteren sei darauf hingewiesen, dass in den im Folgenden abgebildeten Parameterfeldern ein Wert den entsprechenden Parameter stärker erfüllt, je heller er dargestellt wird.

Sichtbarkeit

Dieser Parameter beschreibt eine wichtige Anforderung an die generierte Kameraposition. Wenn möglich, sollte diese so gewählt werden, dass die Zielstruktur sichtbar ist. Demzufolge ist die Bewertung der einzelnen Kamerapositionen direkt proportional zum Grad der Sichtbarkeit. Die zur Berechnung benötigten Informationen stammen aus der Datenbasis. Sie enthält für jede Kameraposition sowohl die Gesamtzahl der Pixel s als

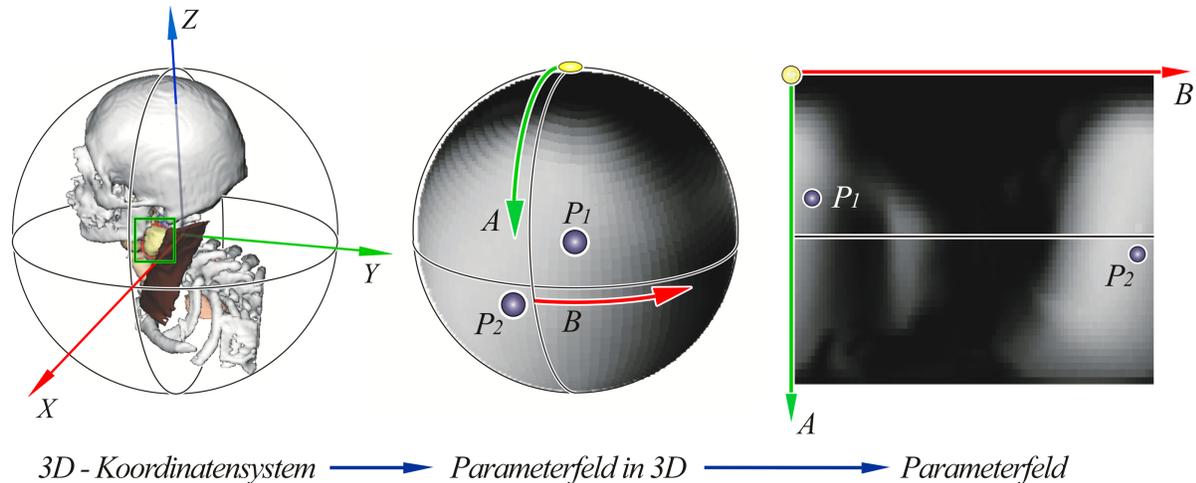


Abb. 3.14: Zusammenhang zwischen Geometrie und Parameterfeld: Das hier gezeigte Parameterfeld beinhaltet die Sichtbarkeit der hervorgehobenen Struktur (Kasten). Deutlich ist zu erkennen, dass zwei räumlich dicht beieinander liegende Positionen (P_1, P_2) auf dem Parameterfeld einen deutlich höheren Abstand aufweisen können.

auch die Anzahl der sichtbaren Pixel v einer Struktur. Das Bewertungsmaß pro Kameraposition (α, β) errechnet sich wie folgt:

$$p_{vis}(\alpha, \beta) = \frac{v_{\alpha, \beta}}{s_{\alpha, \beta}} \quad (3.3)$$

Kontextbezogene Wichtigkeit

In den oft kompakten medizinischen Visualisierungen kommt es häufig zu gegenseitigen Verdeckungen. Mit Hilfe dieses Parameters kann erreicht werden, dass im Falle einer Verdeckung eine Kameraposition eingenommen wird, von der aus gesehen sich nur kontextuell unwichtige Strukturen zwischen Betrachter und Zielstruktur befinden. Somit werden bei einer Behandlung dieser Verdeckung, z.B. durch den Einsatz einer transparenten Darstellung, *Cut-Aways* oder einer vollständigen Ausblendung der verdeckenden Strukturen keine bzw. kaum Kontextinformationen zerstört. Häufig spiegelt sich die Unwichtigkeit in Form einer grundsätzlich abgeschwächten Darstellung der betreffenden Strukturen wider (z. B. höhere Transparenz). In diesem Fall wird der Blick auf die Zielstruktur durch diese kaum gestört und eine entsprechende Kameraposition kann bezogen auf die Sichtbarkeit als günstig bewertet werden (siehe Abb.3.15). Der Parameter „kontextbezogene Wichtigkeit“ ist also als Erweiterung des Parameters „Sichtbarkeit“ zu sehen. Haben alle Strukturen die gleiche Wichtigkeit, ähneln sich beide Parameter stark.

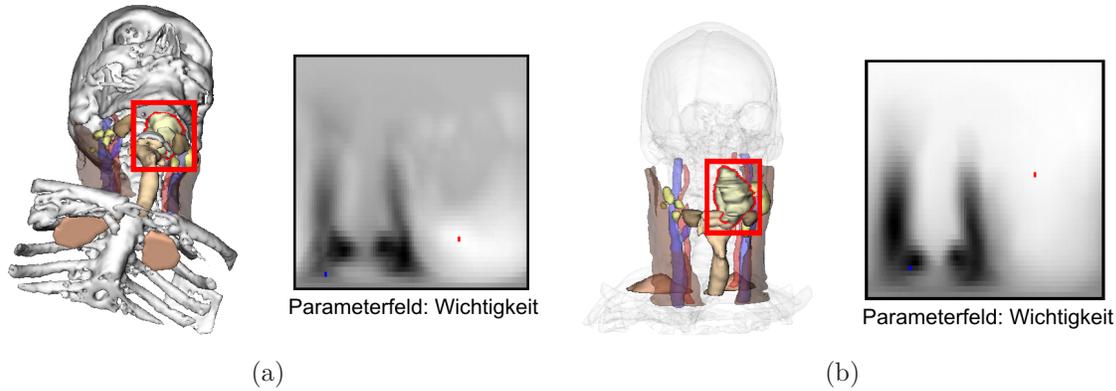


Abb. 3.15: In (a) sind die Knochenstrukturen als wichtig eingestuft und demzufolge opak. Ein Blick auf den Tumor (Kasten) ist nur von unten möglich. Durch die reduzierte Wichtigkeit der Knochenstrukturen in (b) werden diese sehr transparent dargestellt. Der Blick auf den Tumor (Kasten) ist somit auch durch die Knochenstrukturen möglich. Diese Sichtbarkeitsänderung spiegelt sich im zugehörigen Parameterfeld für die Wichtigkeit wider. Der rote Punkt stellt die automatisch gewählte Kameraposition dar.

Aus Metainformationen über die Wichtigkeit aller Strukturen w (je höher desto wichtiger), der Anzahl der Pixel s der Zielstruktur und der Anzahl der Pixel v , durch die sie von den jeweiligen Strukturen verdeckt wird, ergibt sich für jede potentielle Kameraposition folgende Gleichung:

$$p_{imp}(\alpha, \beta) = p_{imp_max} - norm_{[0,1]} \left(\sum_{i=1}^n w_i \cdot \frac{v_{\alpha,\beta}^i}{s_{\alpha,\beta}} \right) \quad (3.4)$$

Statt für w_i den Term $(1 - w_i)$, bei $0 \leq w_i \leq 1$, zu verwenden wird das entstehenden Parameterfeld durch die vorgestellte Subtraktion invertiert. Dies ist möglich, da der Bewertungsterm vorher auf den Bereich $[0, 1]$ normiert wurde ($norm_{[0,1]}$). Durch die Invertierung wird sichergestellt, dass im Fall einer Nichtverdeckung das höchste Bewertungsmaß vergeben wird. Aus der Art der Berechnung ergibt sich, dass im Falle der Verdeckung durch viele unwichtige Strukturen ein höheres Bewertungsmaß vergeben werden kann, als bei der Verdeckung durch eine wichtigere Struktur. Wird demzufolge eine Position eingenommen, bei der sich mehrere unwichtige Strukturen zwischen Betrachter und Zielstruktur befinden, kann trotz deren abgeschwächter Darstellung die Sichtqualität durch die entstehende mehrfache Überlagerung beeinträchtigt werden (siehe Abb. 3.16). Die Anzahl der verdeckenden Strukturen ist somit ein weiterer zu beachtender Parameter.

Anzahl der verdeckenden Strukturen

Die generierte Kameraposition sollte so geartet sein, dass möglichst wenige Strukturen den Blick auf die Zielstruktur verdecken, da ihre Darstellungsparameter bei der Behandlung dieser Verdeckungen so angepasst werden, dass ihr Informationsgehalt abgeschwächt wird oder vollständig verloren geht. Im Sinne der Informationserhaltung sollte eine derartige Anpassung möglichst im geringen Maße durchgeführt werden.

Demzufolge werden Positionen höher bewertet, an denen weniger Strukturen zwischen Betrachter und Zielstruktur liegen. Der Grad der Verdeckung wird miteinbezogen, um einer Unterbewertung entgegenzuwirken. Diese tritt auf, wenn die Zielstruktur von vielen Strukturen nur zu einem geringen Grad verdeckt wird. Aus der Pixelanzahl der Zielstruktur s , der Anzahl verdeckter Pixel v und der aus der Datenbasis ebenfalls ableitbaren Anzahl der verdeckenden Strukturen n ergibt sich für das Parameterfeld folgende Gleichung pro Kameraposition:

$$p_{num}(\alpha, \beta) = n \cdot \frac{v_{\alpha, \beta}}{s_{\alpha, \beta}} \quad (3.5)$$

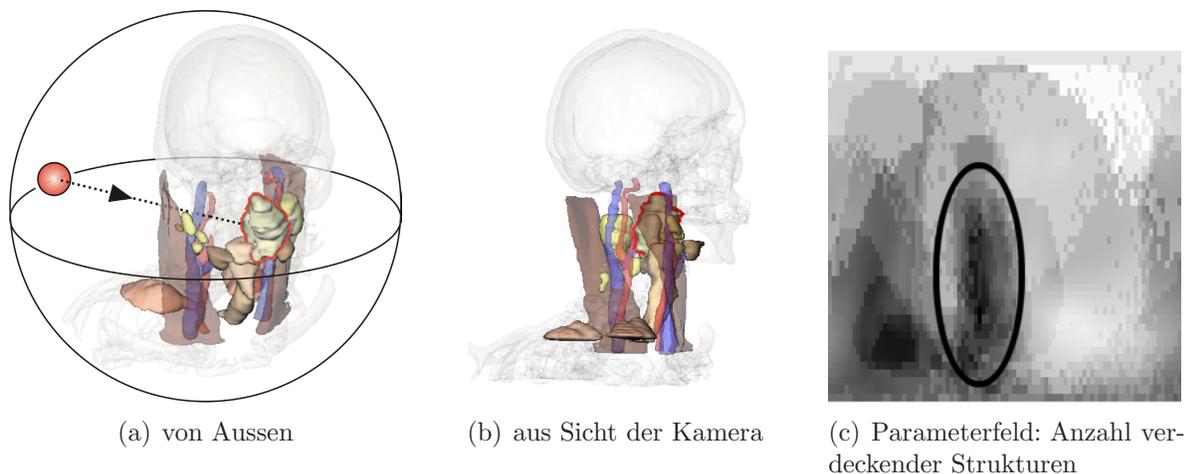


Abb. 3.16: Wird der Tumor (rot umrandet) von der Kameraposition (a) aus betrachtet, ist er trotz der teiltransparenten Darstellung der verdeckenden Strukturen kaum zu erkennen (b). Eine solch gehäufte Überlagerung spiegelt sich auch deutlich durch eine Abwertung der Positionen im Parameterfeld für die Anzahl der verdeckenden Strukturen wider (Ellipse) (c).

Strukturentropie

Unabhängig von dem umgebenden Kontext liefert dieser Parameter Informationen darüber, von welcher Richtung aus betrachtet die Zielstruktur den höchsten Informationsgehalt besitzt. Verschiedene Strukturinformationen können genutzt werden, um eine Aussage über den Informationsgehalt zu treffen. Sie sind von der Art der Struktur

abhängig und können direkt aus der geometrischen Repräsentation abgeleitet werden, z.B. in Form von Krümmungsinformationen, oder auch als vordefinierten Metainformationen vorliegen. Ein einfaches Entropiemaß, welches aus der Datenbasis gewonnen werden kann, ist die projizierte Fläche (siehe Abb. 3.17). Es wird dabei von der Annahme ausgegangen, dass die Position, von der aus die meiste Fläche sichtbar ist, die meisten Informationen über die Struktur liefert. Anhand der in der Datenbasis enthaltenen Information über die Anzahl der Pixel s , durch die sich eine Struktur ohne Einbeziehung möglicher Verdeckungen präsentieren würde, ergibt sich für eine potentielle Kameraposition folgenden Form der Entropieberechnung:

$$p_{ent}(\alpha, \beta) = \text{norm}_{[0,1]}(s_{\alpha,\beta}) \quad (3.6)$$

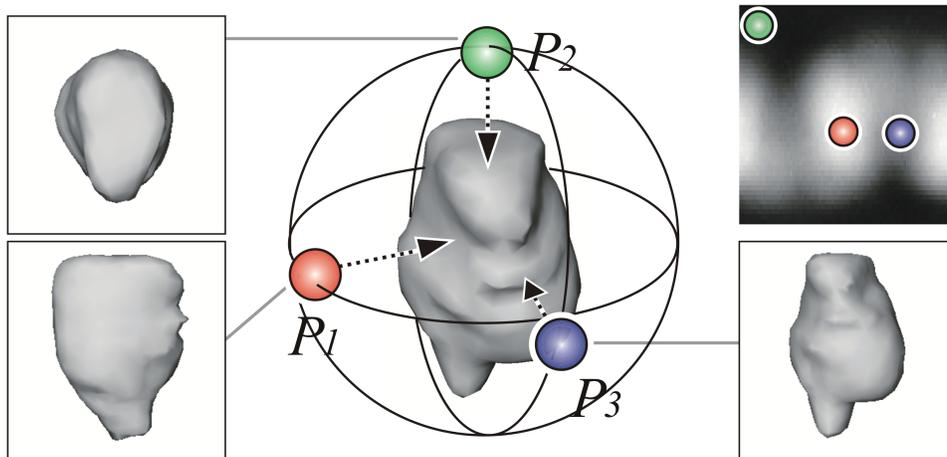


Abb. 3.17: Von Position P_1 aus gesehen präsentiert sich der Lymphknoten mit einer maximalen projizierten Fläche. Deswegen erhält diese Position ein hohes Maß für die Entropie. Es ist jedoch zu erkennen, dass von Position P_3 aus mehr Oberflächendetails zu erkennen sind, obwohl die sichtbare Fläche insgesamt kleiner ist. Hier zeigt sich, dass die projizierte Fläche allein kein sehr detailliertes Maß für den Informationsgehalt ist. Oben rechts ist das Parameterfeld für Entropie und die Anordnung der Positionen darin dargestellt.

Vorzugsblickwinkel

Auch wenn alle vorhergehend angeführten Parameter in die Summe mit einfließen, kann das Ergebnis eine Kameraposition sein, welche für den Betrachter ungewohnt ist. Dadurch fällt es ihm schwer, sich zu orientieren. Weicht der sich ergebende Blickwinkel zu stark von dem für den Betrachter gewohnten Blickwinkel ab, kann es demzufolge passieren, dass er, trotz guter Sichtbarkeitsverhältnisse, die Verhältnisse falsch interpretiert. Diese Betrachtungsgewohnheiten können z.B. aus dem chirurgischen Alltag stammen, oder auch durch die in der medizinischen Bildgebung häufig angewandten Standardblickrichtungen (siehe Abb.3.18) geprägt worden sein.

Um diese Gewohnheiten bei der automatischen Positionierung der Kamera zu berücksichtigen, werden Punkte bzw. Regionen auf der umgebenden Kugel definiert, von denen aus sich der gewohnte Blickwinkel ergibt. Ausgehend von diesen Regionen kommt es mittels einer Distanzfunktion mit steigender Distanz zu einer Abwertung der entsprechenden Kamerapositionen. Es wird eine Abwandlung der klassischen, linearen Distanzfunktion verwendet (siehe Abschnitt 4.2.3), da die Besonderheiten der Kugelgeometrie beachtet werden müssen. So können auf dem Parameterfeld räumlich voneinander getrennte Punkte (Beispiel: linke unterste und linke oberste Position) auf der Kugel in direkter Nachbarschaft liegen.

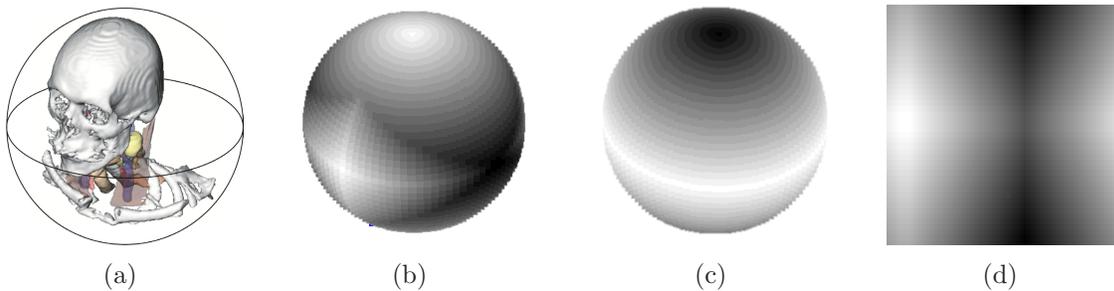


Abb. 3.18: Die Vorzugsblickwinkel in (b) sind so definiert, dass die Visualisierung in (a) vorzugsweise von vorne (*anterior*) oder von oben (*superior*) aus gesehen dargestellt werden soll. Die Vorzugsblickwinkel in (c) orientieren sich entlang des Äquators. In (d) ist zu erkennen, dass die Distanzberechnung nicht von den Grenzen des Parameterfeldes abhängen darf, da dies nicht der Distanz auf der Kugel entsprechen würde.

Dieses Parameterfeld wird auf der Basis von Metainformationen gebildet. Es ist stark anwendungsspezifisch und dementsprechend nicht in der allgemeingültig konzipierten Datenbasis enthalten.

Kameranähe

Die neu generierte Kameraposition sollte, wenn möglich, in der Nähe der aktuellen Kameraposition liegen. Dadurch wird der Aufwand der Neuorientierung, welcher sich für den Nutzer beim Wechsel zur generierten Kameraposition ergibt, verringert. Dies reduziert die Gefahr von Fehleinschätzungen aufgrund einer Falschorientierung. Außerdem wird eine spätere Anwendung in Form von Animationen begünstigt. Durch eine entsprechende Wichtung dieses Parameters kann erreicht werden, dass die für den Animationspfad generierten Kamerapositionen aneinander ausgerichtet sind, wodurch extremen Pfadsprüngen entgegengewirkt werden kann.

Die aktuelle Kameraposition wird im Parameterfeld maximal bewertet, die Wertung der anderen Positionen sinkt mit steigendem Abstand. Hierbei kommt ebenfalls eine modifizierte Distanzfunktion zum Einsatz.

Stabilität

Bei einigen Parameterfeldern, wie z.B. dem der Sichtbarkeit, können Maximalwerte in Form von flächigen Bereichen auftreten. Wird eine Position am Rand eines solchen Bereiches als Ergebnisposition gewählt, ist diese instabil. Am Beispiel der Sichtbarkeit erläutert bedeutet dies, dass eine kleine Änderung dieser Position große Änderungen der Sichtbarkeitsverhältnisse zur Folge haben kann. Aber auch sehr kleine Maximalbereiche liefern solch instabile Positionen („Blick durch ein Nadelöhr“). Da der Nutzer die Visualisierung, ausgehend von der erzeugten Kameraposition, weiter explorieren wird, sollte die Kameraposition möglichst stabil sein. Eine kleine Änderung des Blickwinkels sollte nicht zur Folge haben, dass die Zielstruktur für ihn nicht mehr sichtbar ist (siehe Abb. 3.19). Demzufolge ist es wünschenswert, dass sich die Ergebnisposition möglichst in der Mitte großer maximaler Sichtbarkeitsbereiche befindet und kleine Maximalbereiche gar nicht oder nur abgeschwächt in die Ergebnisfindung einfließen.

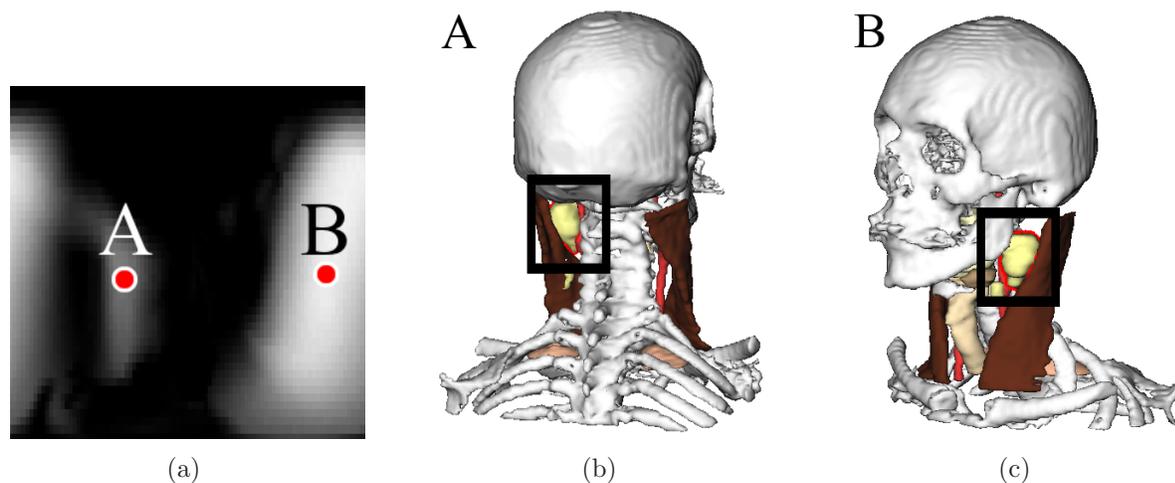


Abb. 3.19: In (a) ist ein Parameterfeld für Sichtbarkeit abgebildet. In (b) und (c) wird dargestellt, wie sich die ausgewählte Struktur (Kasten) von Position *A* und *B* aus präsentiert. Position *A* ist verglichen zu Position *B* instabil, da die Zielstruktur sehr weit hinten liegt und sich ihre Sichtbarkeit dementsprechend aufgrund einer kleinen Kamerabewegung stark ändern würde. Dies zeigt sich durch einen starken Abfall der Sichtbarkeit im Parameterfeld.

Für die Erzeugung eines Stabilitätsmaßes werden die betreffenden Parameterfelder binarisiert. Der dabei verwendete Grenzwert legt fest, welche minimale Güte eine potentielle Region haben muss, damit auf sie das Stabilitätsmaß angewendet wird.

Wird der maximal mögliche Grenzwert gewählt, fließen dementsprechend auch nur Maximalregionen mit ein. Da die stabilste Kameraposition in der Mitte dieser Regionen angenommen wird, dient die Distanz vom Rand der Regionen als Stabilitätsmaß. Die Position innerhalb der Region, welche den größten Abstand von deren Rand besitzt, ist demzufolge die stabilste.

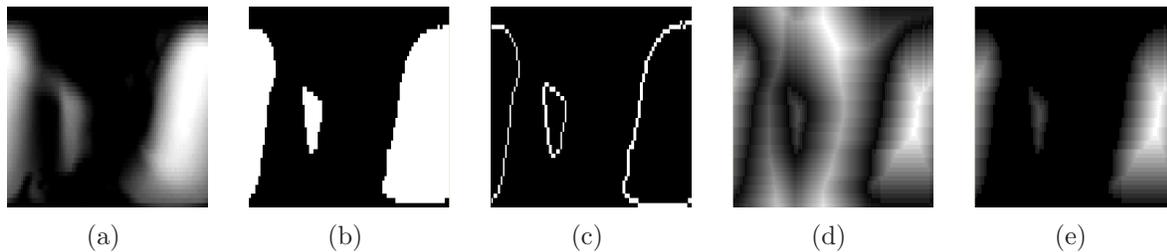


Abb. 3.20: Schritte der Stabilitätsberechnung: (a) Sichtbarkeit; (b) binarisierte Sichtbarkeit; (c) extrahierte Kanten; (d) Distanzmaß, ausgehend von diesen Kanten; (e) binarisierte Sichtbarkeit aus (b) wird als Maske auf (d) angewandt, um das Stabilitätsmaß zu erhalten.

Mittels eines Kantenoperators (Sobel-Filter) werden die Kanten der aus der Binarisierung resultierenden Regionen extrahiert. Die im entstandenen Parameterfeld enthaltenen Kanten dienen als Grundlage für die Distanzberechnung, welche in dieser Form schon bei dem Parameter „Vorzugsblickwinkel“ zum Einsatz gekommen ist. Da nur Distanzen innerhalb der Region von Interesse sind, werden die im daraus resultierenden Parameterfeld enthaltenen Distanzen mit den Werten aus dem ursprünglichen, binären Parameterfeld multipliziert. Je größer eine Region ist, desto höhere Abstände sind möglich. Dies führt dazu, dass darin enthaltene stabile Positionen besser bewertet werden, als stabile Positionen in kleineren Regionen, was der eingangs genannten Forderung an das Stabilitätsmaß entspricht. Die Teilschritte des Verfahrens sind in Abbildung 3.20 dargestellt.

Relativer Abstand

Für die Bewertung der Kameraposition ist der Abstand der Zielstruktur zum Betrachter ebenfalls von Interesse. Liegt eine Struktur sehr weit weg vom Betrachter, muss dieser demzufolge durch die gesamte Visualisierung blicken, um die Struktur zu sehen. Dieser „Tunnelblick“ ist ungeeignet, um Aussagen über das räumliche Umfeld der Zielstruktur zu treffen. Es kann der Fall auftreten, dass die Struktur von der einen Seite betrachtet dicht zum Betrachter liegt und von einer relativ wichtigen Struktur verdeckt wird (opaker) und von der anderen Seite betrachtet zwar weiter weg vom Betrachter liegt, dafür aber von einer unwichtigeren Struktur (transparenter) verdeckt wird. Ohne den Parameter „relativer Abstand“ gibt es keinen Parameter, welcher dazu führen würde, die näher gelegene Position als Ergebnis zu liefern. Diese Unausgeglichenheit würde dazu

führen, dass der Tunnelblick immer als bessere Alternative gewählt werden würde (siehe Abb. 3.21).

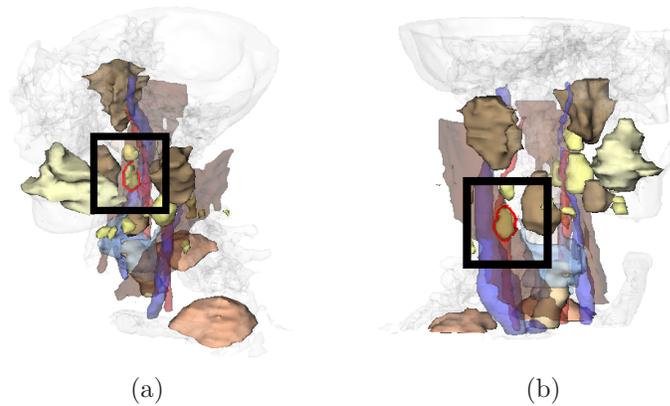


Abb. 3.21: Für die Zielstruktur (Kasten) wird ohne Einbeziehen der Nähe zum Betrachter eine Position gewählt, bei der sie nur durch unwichtige Strukturen (Knochen) verdeckt wird (a). Von dieser Position aus betrachtet, lassen sich keine Aussagen über die Lage der Struktur treffen, da man durch den komplette Schädel blickt. Wird der Abstand mit einbezogen, präsentiert sich die Struktur wie in (b). Sie wird zwar von einer wichtigeren Struktur (Muskel) verdeckt, trotzdem lassen sich bessere Aussagen über ihre Position treffen.

Als Information über den relativen Abstand wird der direkt in der Datenbasis enthaltene minimale z -Wert genutzt. Er ist zwischen 0 (nah) und 1 (fern) normiert und muss somit invertiert werden, damit Positionen, von denen aus gesehen die Struktur nah am Betrachter liegt, hoch bewertet werden (siehe Abb. 3.22).

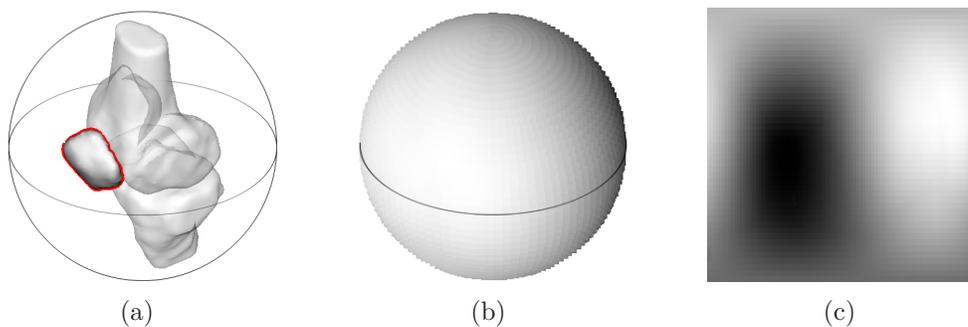


Abb. 3.22: Das Abstands-Parameterfeld für die *Patella* im Kniedatensatz (a). Je näher diese Struktur zum Betrachter liegt, desto höher wird die entsprechende Position im Parameterfeld bewertet (3D (b)/2D (c)).

Sinus-Abwertung

Die Parameterfelder stellen eine aufgefaltete Kugeloberfläche dar. Aus Gründen der einfacheren Berechnung sind die Achsen in Längen- und Breitengrade eingeteilt und daran orientiert. Dies hat zur Folge, dass die Werte von Kamerapositionen an den Polen einer Kugel zu stark repräsentiert werden (siehe Abb. 3.23). Die repräsentierte Fläche

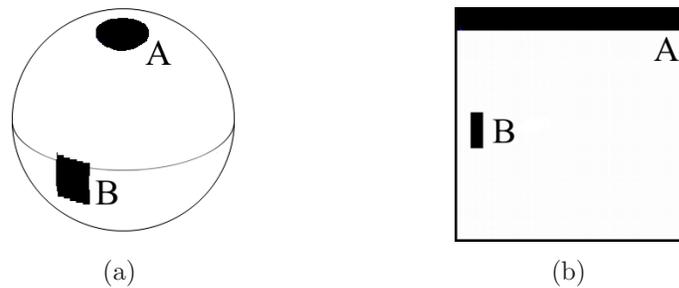


Abb. 3.23: Obwohl die beiden schwarzen Regionen fast die gleiche Fläche auf der Kugel einnehmen (a), wird die polnahe Region A im Parameterfeld durch eine viel größere Fläche repräsentiert.

auf dem Parameterfeld entspricht also nicht der wirklichen Wertigkeit der Informationen, d.h. alle auf der Fläche basierenden Parameter, wie z.B. Stabilitätsinformationen, werden falsch berechnet. Für weiter von den Polen entfernte Positionen gilt dies in abgeschwächter Form. Um die Besonderheit der Kugelform auf die betreffenden Parameterfelder abzubilden, müssen die Informationen zu den Polen hin abgewertet werden. Dies wird durch die Anwendung einer entsprechend skalierten Sinusfunktion entlang der Längengrade erreicht (siehe Abb. 3.24).

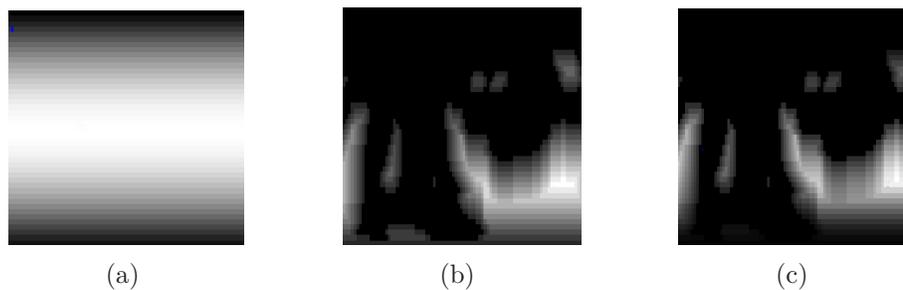


Abb. 3.24: Das Sinus-Parameterfeld (a) wird mit dem Stabilitäts-Parameterfeld (b) multipliziert. Dadurch werden die polnahen Stabilitätsinformationen entsprechend abgewertet (c).

3.3.2 Ergebnisbildung und weiterführende Anwendungen

Die Position des Maximums im durch das gewichtete Summieren der Parameterfelder entstehende Summenfeldes entspricht der Ergebnisposition. Die Berechnung dieses Summenfeldes für eine ausgewählte Struktur stellt sich wie folgt dar:

Name	Bedeutung	Name	Bedeutung
ω_i	Wichtung des Parameters i	p_{imp}	kontextbezogene Wichtigkeit
p_{vis}	Sichtbarkeit	p_{imp_sta}	Stabilität der Wichtigkeit
p_{vis_sta}	Stabilität der Sichtbarkeit	p_{sin}	Sinus-Abwertung für Stabilität
p_{num}	Anzahl verdeckender Strukturen	p_{ent}	Entropie (projizierte Fläche)
p_{pref}	Vorzugsblickwinkel	p_{cam}	Ausgangsposition der Kamera
p_{dis}	relativer Abstand		

$$f_{sum}(\alpha, \beta) = \begin{bmatrix} \omega_{vis} \cdot p_{vis} + \omega_{vis_sta} \cdot p_{vis_sta} \cdot p_{sin} + \\ \omega_{imp} \cdot p_{imp} + \omega_{imp_sta} \cdot p_{imp_sta} \cdot p_{sin} + \\ \omega_{num} \cdot p_{num} + \omega_{ent} \cdot p_{ent} + \omega_{pref} \cdot p_{pref} + \\ \omega_{cam} \cdot p_{cam} + \omega_{dis} \cdot p_{dis} \end{bmatrix} (\alpha, \beta) \quad (3.7)$$

Abbildung 3.28 liefert einen schematischen Überblick über die einzelnen Parameter und die Art und Weise, wie sie für die Ergebnisfindung kombiniert werden.

Zur Ermittlung der Maximalposition wird das Parameterfeld schrittweise abgewandert und der Wert jeder Position überprüft. Der dabei zuletzt gefundene, maximale Wert bestimmt die Ergebnisposition. Durch die Komplexität der durch dieses Parameterfeld beschriebenen Funktion ist häufig nur ein Maximum vorhanden. Dieses repräsentiert eine Kameraposition, welche die sich aus der Parameterwichtung ergebende Anforderung am ehesten erfüllt.

Demzufolge ist das Verfahren nicht „*hard constrained*“, d.h. es gibt keinen starren Qualitätsrahmen, in dem eine Kameraposition liegen muss, um als Ergebnis akzeptiert zu werden. Die Ergebnisposition stellt unter Berücksichtigung der Parameterwichtung das Optimum dar. Allerdings kann nicht festgestellt werden, ob es sich dabei um ein lokales oder globales Maximum in dem sich durch alle möglichen Parameterwichtungen und Kamerapositionen ergebenden Lösungsraum handelt. Die sich durch die Parameterwichtung ergebende Einschränkung des Lösungsraumes ist ein erwünschter Effekt, da in speziellen Anwendungsfällen eine Kameraposition trotz einer schlechteren Bewertung bezüglich der anderen Parameter ein Optimum darstellt.

Neben dem Finden einer einzelnen optimalen Position sind weitere, spezialisierte Anwendungen möglich, welche auf der Basis von Summenfeldern durchgeführt werden können. Im Folgenden werden zwei solche Anwendungen vorgestellt: das Ermitteln weiterer Positionen für eine Struktur und die Anpassung des Summenfeldes, um eine günstige Position für die Analyse minimaler Abstände zwischen Strukturen zu finden. Des Weiteren wird kurz auf die Anwendung der ermittelten Kameraposition im Rahmen von Animationen eingegangen.

Ermittlung von weiteren Positionen für eine Struktur. Es kann dazu kommen, dass weitere, leicht niedriger bewertete Positionen im Summenfeld enthalten sind und somit ebenfalls sehr hochwertige Kamerapositionen repräsentieren (siehe Abb. 3.25). Diese werden bei ausschließlicher Verwendung der Maximalposition trotz ihres hohen Informationsgehaltes nicht beachtet. Befinden sie sich in einem gewissen Abstand von der Maximalposition, können sie verwendet werden, um dem Nutzer einen besseren Überblick über die Lage der Zielstruktur zu liefern, indem diese aus anderen, etwas weniger günstigen Positionen präsentiert wird. Dabei ist zu beachten, dass sie eine vordefinierte Qualitätsstufe nicht unterschreiten.

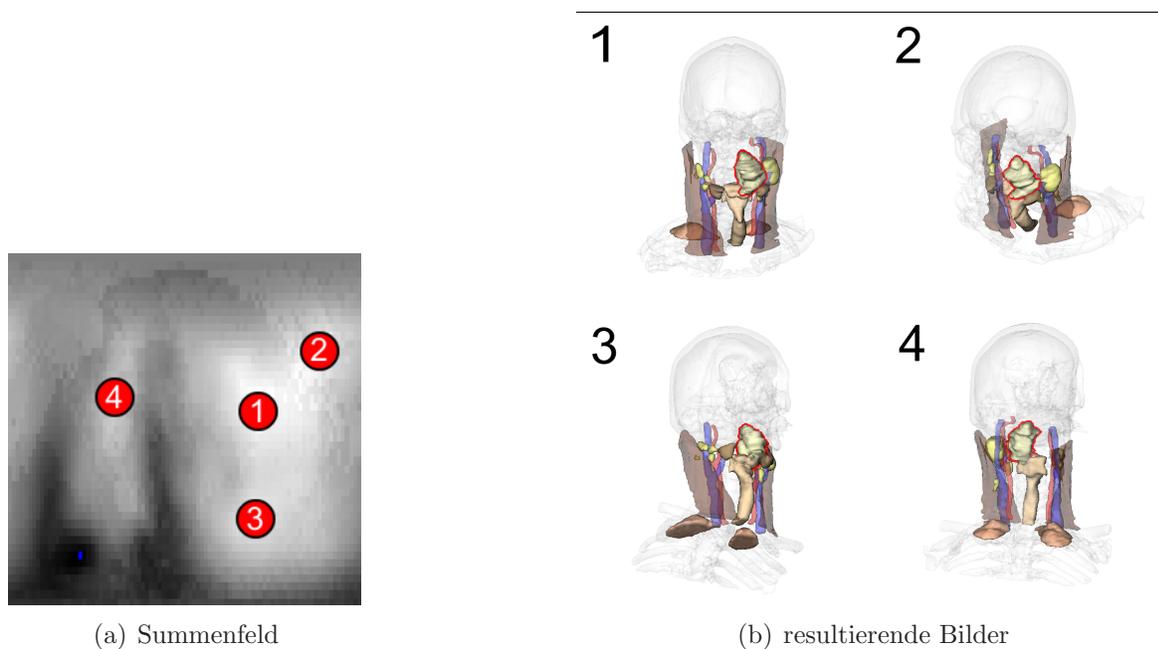


Abb. 3.25: Neben dem globalen Maximum an Position 1, ermöglichen die lokalen Maxima 2 – 4 ebenfalls einen guten Blick auf den Tumor (rot umrandet).

Das Summenfeld wird binarisiert. Der verwendete Grenzwert ist dabei das Maß für die minimal einzuhaltende Qualität der resultierenden Kameraposition. Durch die Berechnung der Stabilität auf der Basis des binarisierten Feldes können mehrere lokale Maxima entstehen. Diese stellen die Kamerapositionen dar, welche für eine Überblicksdarstellung der Zielstruktur geeignet sind. Mithilfe eines horizontal- und vertikal auf das Feld angewandten Sobeloperators lassen sich alle Maxima in Form von Nulldurchgängen identifizieren und als binäres Feld darstellen.

Die Ergebnispositionen dürfen nicht zu dicht beieinander liegen, da sie in diesem Fall redundante Informationen über die Struktur vermitteln und somit für eine Überblickspräsentation ungeeignet sind. Deswegen werden beim Auffinden eines lokalen

Maximums alle lokalen Maxima in einer vordefinierten Umgebung aus der binären Repräsentation entfernt.

Positionen für die Untersuchung minimaler Abstände. Eine häufig auftretende Anwendung bei der Exploration medizinischer Visualisierungen ist die Messung minimaler Abstände. Kommt ein entsprechendes Werkzeug zum Einsatz, wird der Abstand normalerweise durch das Hinzufügen eines metagraphischen Symbols, wie z. B. einer Abstandsmarke visuell verdeutlicht. Solche Abstandsmarken sind im Vergleich zur Gesamtvisualisierung häufig sehr klein. Sowohl das Auffinden der Abstandsmarken als auch das Ausrichten der Kamera, um den Abstand zwischen den Strukturen visuell korrekt einschätzen zu können, erweist sich unter diesen Umständen als kompliziert. Eine automatische Positionierung der Kamera ist demzufolge wünschenswert.

Um die Abstandsmarke ohne perspektivische Verkürzung sehen zu können, muss die Blickrichtung orthogonal dazu ausgerichtet sein. Nur eine Teilmenge der auf der umgebenden Kugel angeordneten Kamerapositionen besitzt eine dementsprechende Blickrichtung. Diese Teilmenge stellt sich als, abhängig von der Ausrichtung der Abstandsmarken orientierter, Umkreis auf der Kugeloberfläche dar (siehe Abb.3.26). Da die Kugeloberfläche durch die Kamerapositionen nur approximiert wird, werden die Kamerapositionen, die den kürzesten Abstand zu diesem Umkreis besitzen, der Teilmenge zugeordnete.

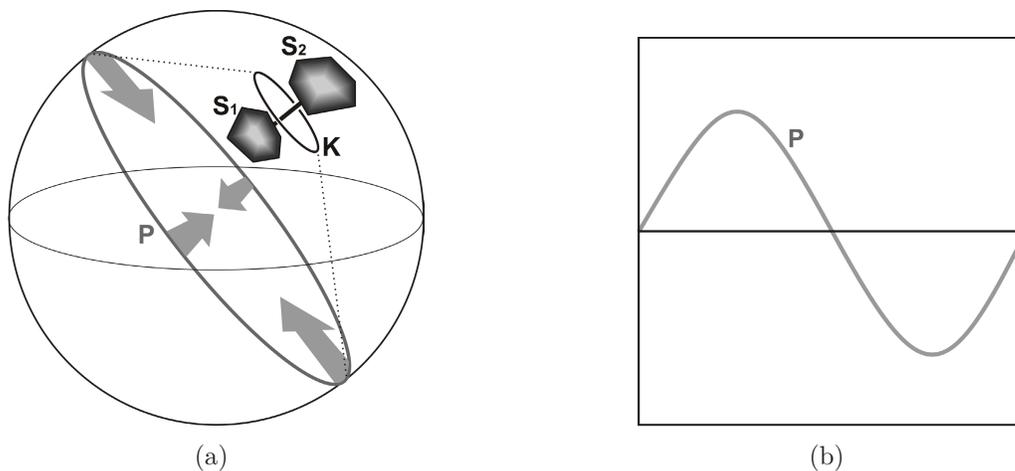


Abb. 3.26: Für die Abstandsmarke zwischen Struktur S_1 und Struktur S_2 in (a) ergeben sich günstige Blickwinkel, wenn sie von Kreis K ausgehend betrachtet wird. Die dazu passenden Kamerapositionen liegen auf der umgebenden Kugel auf dem Pfad P . Dieser Pfad stellt sich im Parameterfeld als eine Kurve dar, wie sie in (b) abgebildet ist.

Aus dieser Teilmenge ergibt sich als Parameterfeld eine binäre Maske (siehe Abb.3.27). Die Summenbilder der beiden an der Abstandsmessung beteiligten Strukturen werden

gemittelt. Darauf wird durch eine Multiplikation die binäre Maske angewendet. Dies hat zur Folge, dass nur noch auf bzw. in der Nähe des Umkreises liegende Kamerapositionen ein Bewertungsmaß besitzen. Durch die Mittelung und die starke Einschränkung des Lösungsraumes kann nicht gewährleistet werden, dass das im Parameterfeld ermittelte Maximum eine für beide Strukturen günstige Kameraposition repräsentiert. Das gemittelte Summenbild sorgt allerdings dafür, dass im Falle einer Verdeckung möglichst günstige Bedingungen für Verfahren zur Behandlung von Verdeckungen herrschen.

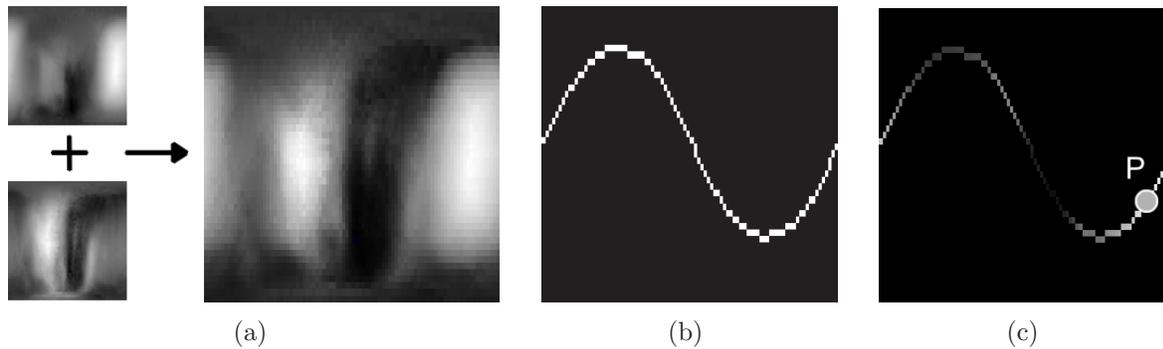


Abb. 3.27: Ablauf der Positionsbestimmung für den minimalen Abstand: Die Summenbilder zweier Strukturen werden zusammengefasst (a). Aus der Teilmenge potentieller Kamerapositionen ergibt sich ein Pfad im Parameterfeld (b). Dieses Parameterfeld wird mit dem gemeinsamen Summenfeld aus (a) multipliziert. Das Maximum im resultierenden Parameterfeld (c) ist die Ergebnisposition.

Die Parameter „Kameranähe“ und „Vorzugsblickwinkel“ sollten nicht in die Summenbilder der jeweiligen Strukturen einfließen, da diese Parameter für das Auffinden und Betrachten der Abstandsmarke keine Bedeutung haben. Durch ihre Einflussnahme wird der Menge günstiger Kamerapositionen unnötig eingeschränkt. Gleiches gilt für den Parameter „Strukturentropie“. Der Nutzer ist bei dieser speziellen Anwendung am Abstand zwischen den beiden Strukturen interessiert. Aus den Strukturen selbst ableitbare Informationen spielen dabei eine untergeordnete Rolle. Zusammenfassend betrachtet gilt: Je weniger Parameter eine zur Sichtbarkeit verschiedene Wertungsverteilung in das Summenfeld einbringen, desto höher ist die Wahrscheinlichkeit, dass die resultierende Kameraposition eine gute Sichtbarkeit beider Strukturen beinhaltet.

Wie oben erwähnt, sind die Abstandsmarken im Vergleich zur Gesamtvisualisierung meist sehr klein. Die resultierende Kameraposition ist aber bezüglich ihrer Blickrichtung auf den Szenenmittelpunkt ausgerichtet. Um die Aufmerksamkeit des Nutzers auf die Abstandsmarke zu lenken, sollte diese möglichst in der Bildmitte und vergrößert abgebildet werden. Eine Verschiebung der Kamera ist somit notwendig. Diese Verschiebung erfolgt auf einer zur Sichtrichtung der Kamera orthogonalen Ebene. Da die Darstellung durch eine orthographische Projektion erfolgt, hat diese parallele Verschiebung, abgesehen von der Darstellung eines anderen Szenenabschnitts, keinen Einfluss auf die sich

dem Nutzer bietenden Sichtbarkeitsverhältnisse. Die Abstandsmarke wird somit auch nach der Verschiebung im gewünschten Neigungswinkel präsentiert.

Automatische Animationen. Eine interessante Anwendung der generierten Kamerapositionen ist das automatische Erzeugen von Animationen. Da dieser Bereich eine eigenständige Anwendung der Kamerapositionen darstellt und nur indirekt mit dem hier vorgestellten Verfahren zur Kamerapositionierung im Zusammenhang steht, wird im Folgenden nur ein kurzer Überblick gegeben.

Automatisch erzeugte Animationen können z. B. genutzt werden, um dem Mediziner einen Überblick über die Verhältnisse in einer medizinischen Visualisierung zu liefern. Welche Strukturen dabei von besonderem Interesse sind, hängt von der Art des präsentierten medizinischen Falles ab. Bei der Planung einer Lymphknotenausräumung sind z. B. alle vergrößerten Lymphknoten von Bedeutung. Für eine Animation wird demzufolge im ersten Schritt für jeden dieser Lymphknoten eine günstige Kameraposition ermittelt. Diese Kamerapositionen dienen als Grundlage für den im nächsten Schritt zu erzeugenden Animationspfad.

Sollen die einzelnen Lymphknoten in einer bestimmten Reihenfolge präsentiert werden, kann der Parameter „Kameranähe“ zur Optimierung der Kamerapositionen genutzt werden. Bei einer entsprechenden Wichtung des Parameters wird dafür gesorgt, dass sich jede nachfolgende Kameraposition möglichst dicht zur vorhergehenden befindet. Dadurch wird extremen Sprüngen im Kamerapfad entgegengewirkt. Spielt die Reihenfolge der an der Animation beteiligten Strukturen keine Rolle, sollte dieser Parameter nicht in die Summenfelder mit einfließen, da keine Aussage über die vorhergehende Kameraposition getroffen werden kann. In diesem Fall sollte der Pfad sich an der räumlichen Verteilung der Kamerapositionen orientieren und die kürzeste Verbindung zwischen ihnen darstellen.

Auch eine zusätzliche Anwendung der zuvor erläuterten Überblickspositionen für eine Struktur ist denkbar. Wenn die Kamera während der Animation die optimale Position für eine bestimmte Struktur erreicht hat, kann von dort ausgehend eine Animation über alle Überblickspositionen gestartet werden. So wird neben der Gesamtübersicht auch noch detailliert auf jede Struktur einzeln eingegangen. Damit es durch das Abweichen vom ursprünglichen Kamerapfad und durch die damit eingeführten Sprünge im Animationspfad zu keiner Irritation des Betrachters kommt, sollten nur Überblickspositionen genutzt werden, die eine gewissen Nähe zum Kamerapfad besitzen. Sollen alle Positionen in den Pfad einfließen, muss dieser Neuberechnet und durch entsprechende Verfahren geglättet werden.

3.4 Zusammenfassung

Es wurde ein bildbasiertes Verfahren entwickelt, um die Sichtbarkeitsverhältnisse in dreidimensionalen Visualisierungen zu analysieren. Die Ergebnisse umfassen nicht nur, ob und zu welchem Grad eine Struktur sichtbar ist, sondern auch die eventuell vorhandenen verdeckenden Strukturen und den Grad der durch sie verursachten Verdeckung. Anhand dieser Informationen können Entscheidungen über die Behandlung der Verdeckungen getroffen werden. Des Weiteren werden Informationen über die projizierte Fläche einer Struktur gewonnen, ohne eventuelle Verdeckungen mit einzubeziehen. Diese Informationen können als einfaches Entropiemaß genutzt werden. Aussagen über relative Abstände der Strukturen zum Betrachter sind ebenfalls in der resultierenden Datenbasis enthalten.

Es wurde analysiert, wie Verteilung der potentiellen Kamerapositionen und die Auflösung der erzeugten Bilder geartet sein müssen, um die geforderte Genauigkeit der Ergebnisse sicherzustellen. Neben Anforderungen an das Sichtbarkeitsverfahren wurden auch die Anforderungen an ein Verfahren zur Kamerapositionierung analysiert. Die durch die Sichtbarkeitsanalyse gewonnenen Informationen sind bezüglich ihrer Mächtigkeit ausreichend, um daraus die notwendigen Parameterinformationen für die automatische Kamerapositionierung zu extrahieren.

An die Problemstellung angepasste Parameter wurden definiert. Des Weiteren wurde die Berechnung zugehöriger Parameterfelder auf der Grundlage der generierten Datenbasis und Metainformationen beschrieben. Ein effektiver Weg um die Ergebnisposition zu ermitteln wurde ebenso erläutert wie spezielle, weiterführende Aufgabenstellungen auf dem resultierenden Summenfeld.

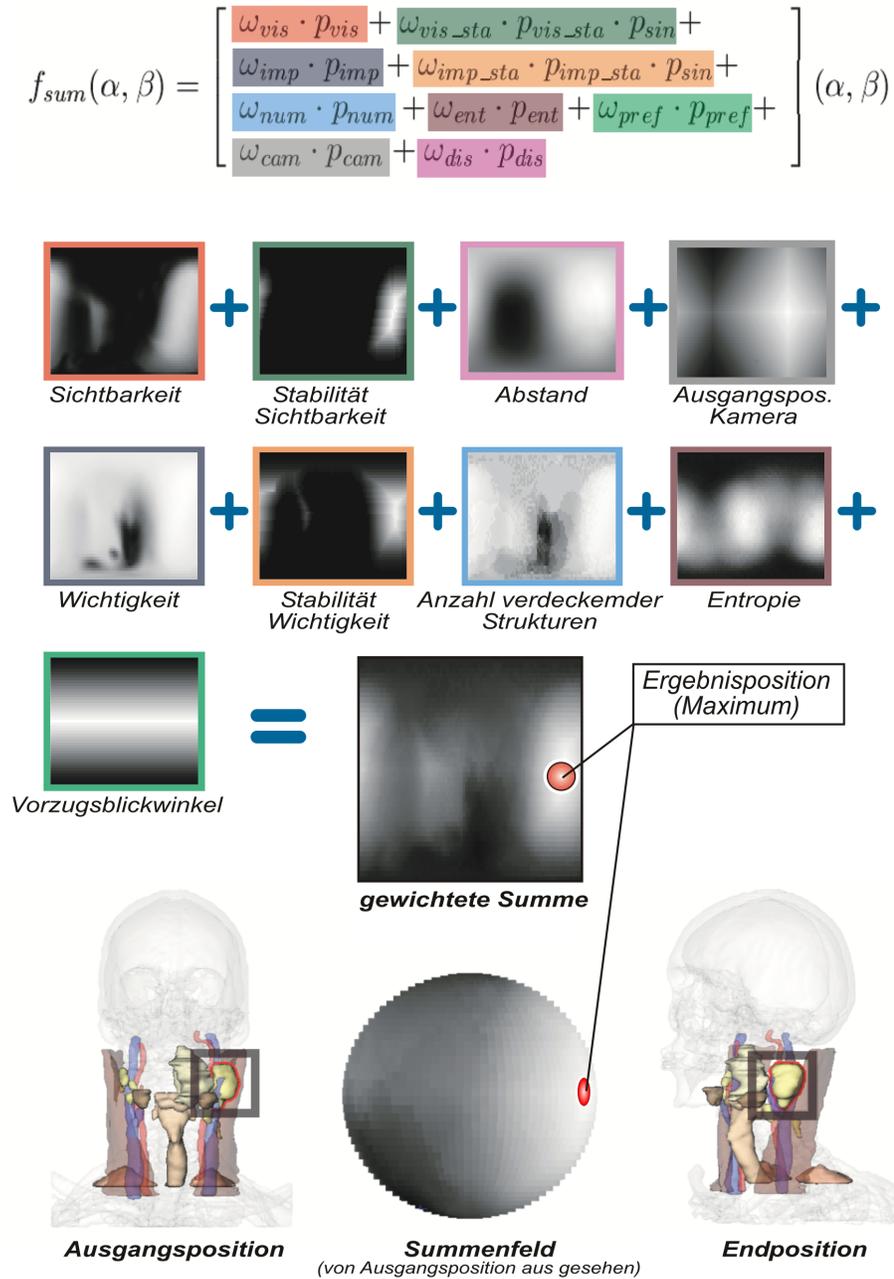


Abb. 3.28: Die einzelnen Parameterfelder werden gewichtet summiert. Das Maximum im daraus resultierenden Summenfeld entspricht der Ergebnisposition.

4 Implementierung

In diesem Kapitel wird die konkrete Umsetzung der Kamerapositionierung beschrieben. Zu Beginn werden die bei der Implementierung verwendete Entwicklungsumgebung und zugehörige Bibliotheken vorgestellt. Des Weiteren wird näher auf das Softwaresystem eingegangen, in welches das Verfahren eingebettet werden soll.

Die Realisierung der beiden Teilschritte „Generierung der Datenbasis“ und „Berechnung der Kamerapositionen“ wird in den darauf folgenden Abschnitten beschrieben. Ferner wird auf während der Umsetzung aufgetretene Probleme und deren Lösung eingegangen und die Laufzeiten der Datenbasisgenerierung werden vorgestellt und diskutiert.

4.1 Werkzeuge zur Implementierung

Die automatische Kamerapositionierung wurde auf der Basis der Entwicklungsplattform MEVISLAB umgesetzt. Es wurde sich für MEVISLAB als Plattform entschieden, da so eine einfache Zusammenarbeit mit dem *Medical Exploration Toolkit* METK ermöglicht wird, welches ebenfalls in MEVISLAB realisiert wurde.

Im Folgenden wird ein Überblick über MEVISLAB und das METK geliefert. Dem vorangestellt werden grundsätzliche Mechanismen von OPEN INVENTOR erläutert. Die grafische Darstellung in MEVISLAB beruht auf dieser *Highlevel*-Grafikbibliothek und spielt in diesem Zusammenhang unter anderem eine wichtige Rolle bei der bildbasierten Sichtbarkeitsanalyse.

4.1.1 Open Inventor

Bei OPEN INVENTOR handelt es sich um eine objektorientierte Grafikbibliothek [WERNECKE, 1993, 1994]. Sie basiert auf den Grafikroutinen von OPENGL. OPEN INVENTOR wurde mit dem Ziel entwickelt, es dem Nutzer auf einfache Weise zu ermöglichen, die Eigenschaften einer dargestellten OPENGL-Szene in Echtzeit zu manipulieren. Diese Manipulationen können z.B. die Kameraposition, die Materialeigenschaften oder die Szenengeometrie betreffen.

Die Darstellungen werden in Form eines Szenengraphen repräsentiert. Die Objekte bzw. Knoten aus denen dieser Graph aufgebaut ist, umfassen nicht nur die dargestellten

Grafikprimitive, sondern unter anderem auch Materialeigenschaften, Kameras oder Transformationen.

Der Szenengraph entspricht vom Aufbau her einem Baum. Ausgehend von der Wurzel wird für die Darstellung der Szene der zugehörige Szenengraph von der Wurzel aus traversiert. Die Knoten des Graphs werden dabei in der Reihenfolge von oben nach unten, von links nach rechts, nacheinander abgearbeitet.

Knoten, welche Informationen über die Grafikprimitive enthalten, stellen spezielle Funktionen bereit, welche bei ihrer Traversierung aufgerufen werden. Je nach Art des Knotens werden diese Funktionen mit `GLRender` bzw `GLRenderBelowPath` bezeichnet. Dort werden für die Darstellung native `OPENGL`-Befehle genutzt.

Abhängig von ihrer Anordnung im Graph können sich Knoten gegenseitig beeinflussen. Knoten, die z.B. Translationen oder Materialeigenschaften beschreiben, wirken sich auf alle nachfolgend traversierten Knoten aus, sofern ihr Einfluss nicht durch weitere derartige Knoten überschrieben oder negiert wird.

Des Weiteren unterstützt `OPEN INVENTOR` das sogenannte Feldkonzept. Alle gängigen Datentypen, wie z.B. Integer oder Float werden in Form von Klassen repräsentiert. Dies ermöglicht eine ständige Überwachung von Variablen dieses Datentyps. Sie können mit Sensoren verbunden werden, welche bei einer Änderung der Variablen bestimmte Funktionen aufrufen, um entsprechend auf die Änderung reagieren zu können. Des Weiteren können Felder untereinander verbunden und auf diese Weise synchronisiert werden (Feldverbindungen), sofern sie vom Datentyp zueinander passen.

`OPEN INVENTOR` stellt sehr viele Möglichkeiten zur Interaktion mit der Szene bereit. Die Transformation einzelner Objekte ist ebenso möglich, wie die ganzer Teilbäume. Es unterstützt die Abfrage von Maus- oder Tastatur-Ereignissen und Funktionen zu deren weiterer Verarbeitung. So wird z.B. auf Wunsch bei einem Mausklick eine Liste sämtlicher Szenenobjekte, die sich hinter dem Cursor befunden haben, zurückgegeben um zu ermitteln, welche Objekte „getroffen“ wurden.

4.1.2 MeVisLab

`MEVISLAB` [HAHN et al., 2003] ist eine Entwicklungsplattform für die Bildverarbeitung und die Visualisierung medizinischer Datensätze. Sie wurde vom Zentrum für medizinische Diagnosesystem und Visualisierung `MEVIS` in Bremen entwickelt.

Wie zuvor erwähnt, basiert die grafische Darstellung in `MEVISLAB` auf `OPEN INVENTOR`. Die in `MEVISLAB` gegebene Möglichkeit, modular abgegrenzte Funktionalitäten in Form eines Netzwerkes zu komplexeren Anwendungen zu kombinieren, orientiert sich dabei stark an der graphenorientierten Architektur, welche in `OPEN INVENTOR` zum Einsatz kommt. `MEVISLAB` stellt hierfür eine grafische Benutzeroberfläche zur Verfügung, in der die funktionellen Knoten in Form von Modulen grafisch repräsentiert werden und

per *Drag&Drop* aus einer Liste ausgewählt und miteinander verknüpft werden können. Es steht dabei eine Vielzahl verschiedener Module zur Auswahl.

Man unterscheidet diese in zwei Hauptklassen: pure INVENTOR-Module, mit denen z.B. ein Szenengraph aufgebaut werden kann und die sogenannten *MeVis Image Processing Library*-Module (ML-Module). Ihr hauptsächliches Einsatzgebiet ist die medizinische Bildverarbeitung.

Um eigene Anwendungen in MEVISLAB umzusetzen, stehen dem Entwickler drei Ebenen zur Verfügung: die C++-Ebene, die Netzwerkebene und die Skriptebene. Die C++-Ebenen bildet die Basis. Dort werden neue Module erzeugt, indem von INVENTOR- oder ML-Klassen abgeleitet wird und sie um entsprechende Funktionalitäten erweitert werden. Des Weiteren können auch auf C++-Ebene, sozusagen in einem Modul, andere Module eingebunden und zu funktionalen Netzwerken verknüpft werden. Alle Module, ob nun von INVENTOR- oder ML-Modulen abgeleitet, unterstützen das schon in Abschnitt 4.1.1 beschriebene Feldkonzept.

Die auf diese Weise neu entwickelten Module stehen dann auf der nächsthöheren Ebene, der Netzwerkebene, zur Verfügung. Dort können sie, abhängig von der Art ihrer Basisklasse und der auf C++-Ebenen definierten Ein- und Ausgänge, mit anderen Modulen zu komplexeren Netzwerken kombiniert werden. Auch das Erzeugen von Feldverbindungen ist möglich.

Sehr komplexe oder funktional abgeschlossene Netzwerke werden in sogenannten Makromodulen zusammengefasst. Dem Nutzer wird eine grafische Oberfläche präsentiert, von der aus er das zugrunde liegende Netzwerk ansteuern kann, indem er z.B. Feldwerte ändert oder Prozesse startet. Über die dritte Ebene, die Skriptebene, wird die Steuerung der Netzwerke automatisiert. Einem Makromodul wird eine Skriptdatei zugeordnet. Diese kann in den Skriptsprachen JAVASCRIPT oder PYTHON realisiert werden. In ihr wird die Verbindung zwischen komplexeren Manipulationen des zugrunde liegenden Netzwerkes und der grafischen Nutzerschnittstelle des Makromodules hergestellt.

4.1.3 METK

In MEVISLAB ist die Verarbeitung und Exploration medizinischer Fälle und aller damit zusammenhängenden Informationen häufig mit dem Aufbau sehr komplexer und umfangreicher Netzwerke verbunden. Das *Medical Exploration Toolkit* ist eine Sammlung von Werkzeugen, welche diesen Verarbeitungsprozess vereinfachen, indem die einzelnen Teilschritte mittels sinnvoll voneinander abgetrennten Modulen realisiert werden. Es ist möglich, auf einfache Weise Fälle und alle zugehörigen Metainformationen zu laden und über eine einheitliche Schnittstelle allen METK-Modulen zugänglich zu machen. Diese Schnittstelle stellt sich in Form des *METKManager*-Moduls dar. In ihm sind alle notwendigen Informationen in Form von Objekten hinterlegt.

Jedes Objekt (*ObjectID*) besitzt verschiedene Informationsschichten (*LayerID*) in denen die einzelnen Informationen vorliegen (*InfoID*). Somit ergeben sich für die Organisation der Daten drei Hierarchieebenen.

Die im Rahmen dieser Arbeit entwickelten Verfahren wurden so realisiert, dass sie den `METKManager` als Schnittstelle nutzen und demzufolge eine einfache Einbindung in das METK ermöglicht wird. Die für die Generierung der Datenbasis notwendigen Szeneninformationen werden aus ihm gewonnen, die aus der Berechnung der Kamerapositionen resultierenden Informationen werden für die weitere Nutzung in ihm hinterlegt.

4.2 Umsetzung der automatischen Kamerapositionierung

Die automatische Kamerapositionierung teilt sich in zwei Schritte: Die Generierung der Datenbasis in Form eines Vorverarbeitungsschrittes und die auf der Datenbasis beruhende Berechnung einer günstigen Kameraposition. Dementsprechend findet die Realisierung der automatischen Kamerapositionierung auch in Form zweier getrennter Implementierungen statt. Trotz der in den folgenden Abschnitten getrennten Betrachtung dieser beiden Bereiche, besteht ein Zusammenhang welcher schematisch in Abbildung 4.1 wiedergegeben wird.

4.2.1 Generierung der Datenbasis

Das in Abbildung 4.2 dargestellte Netzwerk enthält alle Komponenten, welche für die Generierung der Datenbasis notwendig sind. Eine zentrale Rolle spielt das neu entwickelte Modul `SoVisibilitySolver`. Dort findet die eigentliche Sichtbarkeitsanalyse statt, nachdem ihm mit Hilfe des ebenfalls neu entwickelten `METKToSolverConnection`-Moduls die notwendigen Szeneninformationen aus dem `METKManager` übergeben wurden. Für die Durchführung der Sichtbarkeitsanalyse ist der `OffscreenRenderer` und der angeschlossene `Viewer` von Bedeutung.

Im Folgenden wird näher auf die Implementierungsbesonderheiten des Akquirierungsprozesses im Modul `SoVisibilitySolver` eingegangen, wobei auch Bezug auf die Bedeutung der anderen Module im Netzwerk genommen wird.

Der Akquirierungsprozess

Um die für Datenbasis notwendigen Informationen gewinnen zu können, muss ein kontinuierlicher Prozess ausgeführt werden, der sich aus den folgenden Schritten aufbaut:

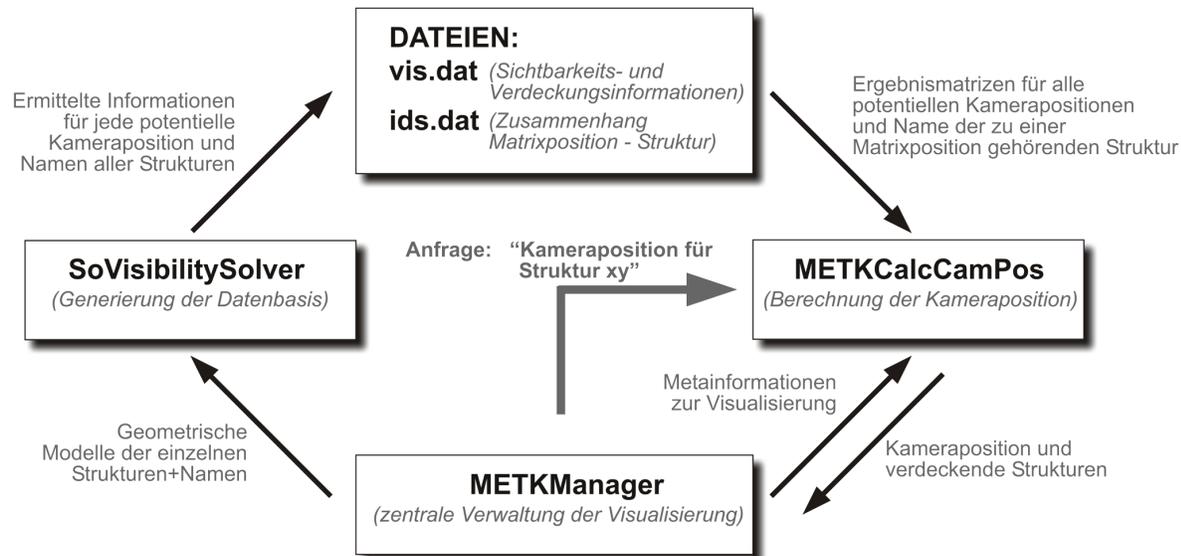


Abb. 4.1: Der schematisch dargestellte Informationsfluss zwischen den Modulen. Der **METKManager** hält alle relevanten Informationen über die Visualisierung. Über werden auch die virtuelle Kamera und diverse Anfragen koordiniert. Die Kommunikation zwischen den Implementierungen der beiden Teilschritte **SoVisibilitySolver** (Generierung der Datenbasis) und **METKCalcCamPos** (Berechnung der Kameraposition) findet nur indirekt über Dateien statt. Trotz dieser Trennung sind alle Komponenten Teil eines Gesamtkonzepts zur automatischen Positionierung der Kamera.

1. Positionieren der Kamera auf der umgebenden Kugel
2. Auslesen der z-Buffer:
 - Darstellen aller Strukturen und Speichern des gemeinsamen z-Buffers (Gesamtbuffer)
 - Darstellen jeder einzelnen Struktur und Speichern des jeweiligen z-Buffers
3. Auswerten der z-Buffer:
 - Aus dem Gesamtbuffer relevante Pixelpositionen bestimmen und für jede dieser Positionen die Reihenfolge der einzelnen Strukturen anhand ihrer z-Buffer-Werte ermitteln
 - Diese Reihenfolge auswerten und die daraus gewonnenen Erkenntnisse in die Ergebnismatrix übertragen
4. Nächste Kameraposition auf der Kugel einnehmen

Auf die Besonderheiten der drei Hauptschritte eins bis drei wird in den folgenden Abschnitten detaillierter eingegangen.

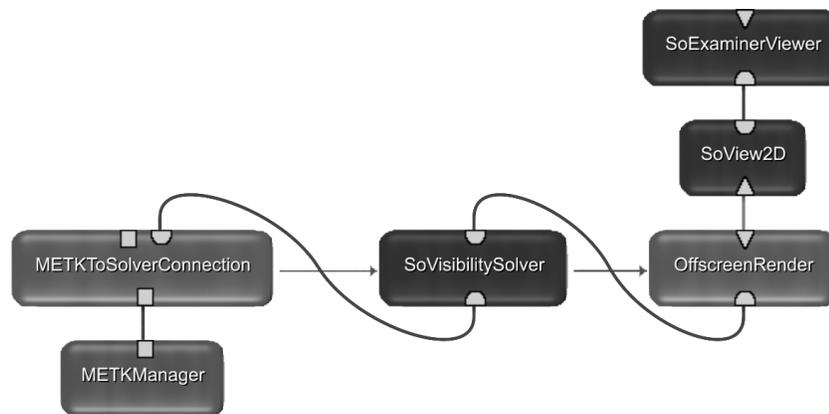


Abb. 4.2: MEVISLAB-Netzwerk für die Generierung der Datenbasis

Positionierung der Kamera. Als Kamera wird eine `SoOrthographicCamera` genutzt, welche auf C++-Ebenen unter das `SoVisibilitySolver`-Modul gehängt wurde (siehe Abb. 4.3). Um die Ausdehnung und Position der Kugel zu berechnen, auf dem die Kamerapositionen während der Datenakquisition verteilt werden, wird die *Bounding Box* aller Strukturen der Szene genutzt. Als Kugeldurchmesser wird die längste diagonale Ausdehnung der *Bounding Box* genutzt, ihr Mittelpunkt entspricht dem Mittelpunkt der Kugel. So wird sichergestellt, dass von allen Kamerapositionen die gesamte Szene sichtbar ist.

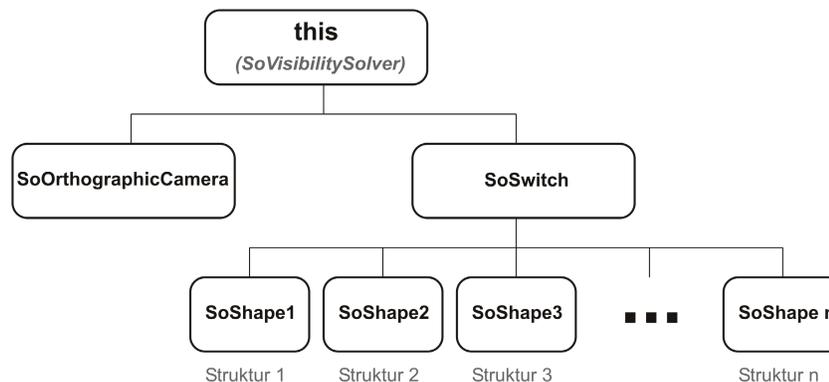


Abb. 4.3: Internes Netzwerk im `SoVisibilitySolver`: Dieses Netzwerk wurde auf C++-Ebene realisiert. Die Strukturmodelle werden in Form eines `SoShape`-Knoten unter den *Switch* gehängt. Dem Ganzen ist eine orthographische Kamera vorangestellt.

Die Kamerapositionen auf der Kugel richten sich nach den Längen- und Breitengraden auf der Kugel. Ein vorher festgelegter Gradabstand bestimmt die Anzahl der Kamerapositionen und die Dichte ihrer Verteilung. Wie in Abschnitt 3.2.2 erläutert ist eine derartige Verteilung der Kamerapositionen nicht optimal, da es zu einer Häufung der

Kameraposition an den Polen kommt. Es wurde sich aufgrund der einfachen Implementierung und der ebenso einfachen interpolationsfreien Erzeugung der Parameterfelder trotzdem für diese Verteilungsmethode entschieden. In Hinblick auf den damit verbundenen Mehraufwand während der Sichtbarkeitsanalyse ist dies jedoch nicht als endgültige Lösung anzusehen.

Auslesen der z-Buffer. Um die z -Buffer der einzelnen Strukturen zu erhalten, müssen diese gerendert werden. Allerdings sind alle Abläufe in MEVISLAB *Demand Driven*, d.h. sie werden erst ausgeführt, wenn die Notwendigkeit dafür besteht. Dies gilt auch für das *Rendering*. Deswegen befindet sich im Netzwerk für die Generierung der Datenbasis ein Viewer (`SoExaminerViewer`). Die durch ihn erfolgende Darstellung der Strukturen sorgt dafür, dass das *Rendering* durchgeführt wird.

Dem Viewer sind die Module `SOVIEW2D` und `OFFSCREENRENDERER` vorangestellt. Dadurch wird ein *Offscreen Rendering* realisiert. Das bedeutet, die während des *Rendering*-Prozesses erzeugten Bilder sind nicht vom Viewport abhängig, in dem sie dargestellt werden. Demzufolge hat eine Änderung der Größe des Viewer-Fensters keinen Einfluss auf die Größe der erzeugten Buffer. Auch wenn ein Teil des Viewer-Fensters durch ein anderes Fenster verdeckt wird, bleibt der Buffer davon unbeeinflusst. Bei einem *Onscreen Rendering* werden nur Bufferinformationen für die sichtbaren Bereiche eines Fensters erzeugt. Der Einsatz des `OffscreenRenderers` stellt somit sicher, dass die Buffer unabhängig von Position und Zustand des Viewers vollständig sind und eine konstante Größe besitzen. Ein weiter Vorteil der dieses Modul liefert, ist die Möglichkeit über eine Feldverbindung gezielt zu steuern, wann ein Bild erzeugt wird.

Wie in Abbildung 4.3 zu erkennen ist, sind die geometrischen Modelle der Strukturen unter einem `SoSwitch`-Modul angeordnet. Durch das Setzen einer entsprechenden Feldvariablen ermöglicht es dieses Modul, gezielt auszuwählen, ob alle oder nur eine spezifische Struktur dargestellt werden soll. Auf diese Art und Weise wird der Gesamtbuffer und für jede Struktur ein eigener z -Buffer erzeugt. Die Reihenfolge in der die n Strukturen unter dem `SoSwitch` angeordnet sind, entspricht ihren IDs. Mit diesen IDs werden sie während der gesamten Akquirierungsprozesses unterschieden.

Die z -Werte der Buffer sind zwischen Null und Eins normiert, abhängig von der Lage der zugehörigen Strukturbestandteile zwischen der *Near-* und *Far Clipping Plane*. Die bei der Darstellung genutzte `SoOrtographicCamera` besitzt diesbezüglich eine interne Optimierung. Die *Near-* und *Far Clipping Plane* richten sich nach der *Bounding Box* der darzustellenden Szene. Dadurch wird die Genauigkeit des z -Buffers erhöht. Ein Vergleich der verschiedenen z -Buffer ist allerdings nicht möglich, da sich die enthaltenen z -Werte nach der räumlichen Ausdehnung der einzelnen dargestellten Strukturen und nicht nach der gesamten Szenen richten. Aus den z -Buffern zweier Strukturen kann somit keine Aussage getroffen werden, wie diese Strukturen, bezogen auf ihren Abstand vom Betrachter, zueinander liegen.

Um dies zu verhindern, müssen vor jedem *Rendering* absolute Werte für die Definition der *Near-* und *Far Clipping Plane* übergeben werden. `SoOrthographicCamera` stellt dafür zwei Variablen zur Verfügung: `nearDistance` und `farDistance`. Der Kameraabstand zur *Near Clipping Plane* wird auf Null und der Abstand der *Far Clipping Plane* auf den Durchmesser der umgebenden Kugel gesetzt. So wird sichergestellt das der durch die *Clipping*-Ebenen aufgespannte Raum minimal sowie konstant ist und trotzdem alle Strukturen der Szene enthält. Die z -Werte aller Buffer können nun miteinander verglichen werden.

Die z -Buffer werden in der `GlRenderBelowPath` des `SoVisibilitySolver`-Moduls mit Hilfe der nativen `OpenGL`-Funktion `glReadPixels` ausgelesen und in ein entsprechend skaliertes `Float-Array` übertragen. Nachdem die gesamte Szene und jede der n Strukturen einzeln dargestellt wurden, liegen somit $n+1$ `Float-Arrays` mit den zugehörigen z -Buffern vor. Diese werden ausgewertet.

Auswerten der z -Buffer. An jeder Pixelposition, an der im Gesamtbuffer ein z -Wert ungleich Eins vorliegt, befinden sich Pixel, die nicht zum Hintergrund gehören. Dort befinden sich demzufolge relevante Informationen über die Reihenfolge, in der die Strukturen bezüglich des Abstands zum Betrachter auftreten. Alle n Buffer werden an dieser Pixelposition untersucht. Wird ein z -Wert ungleich eins gefunden, wird er zusammen mit der ID der Struktur in einer Liste gespeichert.

Die Liste wird mit Hilfe des *Quicksort*-Algorithmus nach den z -Werten aufsteigend sortiert. Je weiter vorn eine Struktur in dieser Liste steht, desto dichter liegt sie zum Betrachter. Somit können aus dieser Liste Sichtbarkeits- bzw. Verdeckungsinformationen für eine Pixelposition extrahiert werden. Dieser Prozess wird in Algorithmus 1 beschrieben.

Algorithmus 1 : Extrahieren der für die Ergebnismatrix wichtigen Informationen

Eingabe : nach z -Werten aufsteigend sortierte Liste L mit allen z -Werten und den zugehörigen StrukturIDs

Ausgabe : aktualisierte Werte in Matrix M für aktuelle Kameraposition

```

1 for  $i \leftarrow 0$  to  $L.length()$  do
2    $ID \leftarrow L[i].ID$ 
3   Erhöhe in Matrix  $M$  die Anzahl der Strukturpixel für Struktur  $ID$  um 1
4   if  $i = 0$  then
5     Erhöhe in Matrix  $M$  Anzahl der sichtbaren Strukturpixel für Struktur  $ID$  um 1
6   for  $j \leftarrow i + 1$  to  $L.length()$  do
7      $ID2 \leftarrow L[j].ID$ 
8     Erhöhe in Matrix  $M$  die Anzahl der Pixel durch die Struktur  $ID2$  von Struktur
       $ID$  verdeckt wird um 1

```

Nachdem alle relevanten Pixelpositionen auf diese Art und Weise ausgewertet wurden, ist die Ergebnismatrix M für die aktuelle Kameraposition p vollständig. Sie enthält für jede Struktur Informationen darüber, an wie vielen Pixelpositionen sie repräsentiert wird, an wie vielen Positionen sie sichtbar ist und welche Strukturen sie im Falle einer Verdeckung durch wie viele Pixel überdecken. Es wird zur nächsten Kameraposition gewechselt und dort erneut mit dem Auslesen und Auswerten der $n + 1$ Buffer begonnen.

Probleme bei der Umsetzung des Akquirierungsprozesses

Fast jeder Teilschritt des Akquirierungsprozesses setzt voraus, dass zuvor ein Bild der entsprechend angepassten Szene (Kameraposition, alle Strukturen - eine Struktur) erzeugt wurde, um die resultierenden Buffer entsprechend auswerten zu können. Dies hat zur Folge, dass der Prozess der Akquirierung sich nicht in Form einer einfachen Schleife auf der C++-Ebene realisieren lässt. Für die Darstellung der Szenen muss der Szenengraph erneut traversiert werden. Dies ist allerdings erst dann möglich, wenn die intern aufgerufenen Funktionen eines im Szenengraph enthaltenen Moduls abgearbeitet sind.

Eine C++-Schleife im Modul `SoVisibilitySolver` würde komplett durchlaufen werden, bevor ein neues Bild und somit die für den korrekten Ablauf des Akquirierungsprozesses notwendige Buffer erzeugt werden würden. Es wurden drei verschiedenen Varianten untersucht, um den korrekten Ablauf des Prozesses zu realisieren: Schleifen auf Skriptebene, das Entkoppeln der Schleife mit Hilfe eines Timers und Framebasierter Ablauf auf der Basis der `GLRenderBelowPath`

- **Skript-Schleife:** Das Problem des Abarbeitens der Schleife, ohne das Bilder erzeugt werden, kann umgangen werden, indem die Schleife auf einer höheren, über dem eigentlichen Netzwerk liegenden Ebene realisiert wird. Dies ist in Form eines Skriptes auf dem Netzwerk möglich. Über Feldvariablen kann das `SoVisibilitySolver`-Modul in den entsprechenden „Modus“ versetzt werden („Positioniere Kamera“, „Stelle Struktur x dar“, „Speichere z -Buffer“ etc.) und der `OffscreenRenderer` aktiviert werden, um das Bild zu erzeugen. Allerdings ist eine solche Skript-Schleife nicht so schnell und effizient, wie dies auf C++-Ebene möglich wäre.
- **Timer:** Mit Hilfe eines Timers kann auf C++-Ebenen eine Entkoppelung der Schleife erreicht werden. Das bedeutet die Schleife wird kurzzeitig unterbrochen, um die Ausführung übergeordneter Prozesse zu ermöglichen. Ein `SoTimerSensor` wird so konfiguriert, dass er in bestimmten Zeitintervallen eine statische Funktion im Modul aufruft. In dieser Funktion kann das Modul abhängig von der aktuell benötigten Funktionalität in den entsprechenden Modus versetzt werden. Um eine hohe Geschwindigkeit des Verfahrens zu gewährleisten, sollten diese Intervalle sehr klein gewählt sein.

Allerdings kann der Fall eintreten, dass der *Rendering*-Prozess oder das Auswerten der *z*-Buffer noch nicht abgeschlossen sind. Dann würde das Modul erst ein vollständiges Zeitintervall später in den nächsten Schritt der Schleife eintreten, was zu kurzen Pausen während der Akquirierung führen würde. Da ein effektiver Ablauf des Verfahrens angesichts der hohen zu verarbeitenden Datenmenge wichtig ist, sind solche Pausen nicht erwünscht.

- **Framebasierter Ablauf:** Wie eingangs erläutert, stehen die Schritte der Akquirierung in direkten Zusammenhang mit dem Erzeugen von Bildern. Die bei jedem *Rendering* aufgerufene Funktion `GLRenderBelowPath` kann also genutzt werden, um die Schleife auf C++-Ebenen zu entkoppeln. Verglichen mit dem Timer-Ansatz können so unnötige Pausen vermieden werden.

Nachdem in der `GLRenderBelowPath` die für den aktuellen Schritt notwendigen Auswertungen durchgeführt worden sind, werden alle für den nächsten Schritt notwendigen Vorbereitungen getroffen (z.B. Neupositionierung der Kamera, Anpassung der `SoSwitch`-Konfiguration) und dann von dort aus über eine Feldverbindung der `OffscreenRenderer` aktiviert. Durch das daraufhin erfolgende *Rendering* kommt es zu einem erneuten Aufrufen der `GLRenderBelowPath`. Der neu erzeugte Buffer wird gespeichert oder auch ausgewertet und alle Vorbereitungen für den nächsten Schritt werden getroffen, bevor der `OffscreenRenderer` erneut aktiviert wird.

So wird ein kontinuierlicher Prozess ermöglicht, dessen Verweildauer in einem Modus direkt von der wirklich benötigten Zeit und nicht von einem festgelegten Intervall abhängig ist. Dieser Ansatz wurde dann letztendlich auch umgesetzt und ist Teil der aktuellen Implementierung.

Wie in Abbildung 4.2 zu erkennen ist, stellt das Modul `METKToSolverConnection` die Verbindung zwischen dem `METKManager` und dem `SoVisibilitySolver` her. Es liest die geometrischen Modelle der Strukturen aus dem `METKManager` ein, welche dort in Form von Adress-Pointern repräsentiert werden und hängt diese unter den schon erwähnten `SoSwitch`, welcher dann an das Modul `SoVisibilitySolver` übergeben wird. Des Weiteren stellt es einen Zusammenhang zwischen den IDs der Strukturen und den Strings her, mit denen sie im `METKManager` bezeichnet werden.

Dieses Modul ist sehr spezialisiert und kann nur in Zusammenhang mit dem `SoVisibilitySolver` genutzt werden. Demzufolge wäre Vereinigung der Funktionalitäten von `SoVisibilitySolver` und `METKToSolverConnection` sinnvoll. Dazu müssen `INVENTOR`- und `ML`-Eigenschaften in einem Modul vereint werden. Nur ein `ML`-Modul kann auf Informationen aus dem `METKManager` zugreifen und die auf `INVENTOR`-Module beschränkte `GLRenderBelowPath` ist essentiell für den Akquirierungsprozess.

Es wurde versucht, ein `INVENTOR`-Modul in einem `ML`-Modul zu kapseln um die notwendigen Funktionalitäten in einem Modul zu vereinen. Allerdings funktionierte in diesem

Fall die Entkopplung der Schleife auf C++-Ebene nicht mehr. Der framebasierte Entkopplungsansatz basiert darauf, dass andere Module über Feldverbindungen angesprochen werden. Das dafür notwendige *Message Handling* wird in ML-Modulen auf andere Weise realisiert als in INVENTOR-Modulen. Während dort die über Feldverbindungen durchgeführten Aufrufe in einer Art Stack angeordnet und demzufolge nacheinander aufgerufen werden, kommt es in ML-Modulen zur sofortigen Ausführung der Aufrufe. Der Effekt der Entkopplung wird damit vollständig aufgehoben.

Da nur das Umsetzen der Schleife auf der langsameren Skriptebene dieses Problem umgehen würde, wurde sich für die Auslagerung der Kommunikationsfunktionen in ein weiteres Modul entschieden. Trotzdem sollte dies nicht als endgültige Lösung angesehen werden, da derartig stark spezialisierte Module dem Konzept einer guten modularen Softwarearchitektur widersprechen.

Repräsentation der Ergebnisse

Die Generierung der Datenbasis ist ein Vorverarbeitungsschritt. Er kann demzufolge ein gewisser Zeitraum zwischen der Erzeugung und der Verwendung der Ergebnisse liegen. Deshalb ist es nahe liegend, die Ergebnisse in Dateien zu speichern. Während des kompletten Prozesses entstehen zwei Dateien: die Datei „ids.dat“ und die Datei „vis.dat“.

Identifikationsdatei „ids.dat“. Diese Datei wird durch das Modul `METKToSolverConnection` erzeugt. Sie ist im lesbaren ASCII-Format formatiert und enthält die IDs der Strukturen sowie die zugehörigen Namen, unter denen sie im `METKManager` hinterlegt sind. Sie stellt somit einen Zusammenhang zwischen den Falldaten des METK und den Ergebnissen aus dem `VisibilitySolver` her. Zusätzlich werden in ihr die Informationen über die Größe und Position der die Szene umgebenden Kugel hinterlegt.

Ergebnisdatei „vis.dat“. Diese Datei ist binär formatiert und enthält alle Ergebnisse aus dem Akquirierungsprozess. Sie wird durch das Modul `SoVisibilitySolver` erzeugt und enthält die Ergebnismatrizen aller Kamerapositionen. Um eine korrekte Auswertung und Einschätzung der Ergebnismatrizen zu ermöglichen, sind im Header Informationen zu der Genauigkeit des Akquirierungsprozesses (Viewport-Auflösung, Anzahl der Kameraposition pro Längen- und Breitengrad) sowie die Anzahl aller Strukturen hinterlegt.

4.2.2 Diskussion der Laufzeit

Die Generierung der Datenbasis wurde auf einem System mit einer INTEL CELERON-CPU (2.8 GHz), 1 GB Ram und einer NVIDIA GEFORCE 6200 TC Grafikkarte durchgeführt. Die Laufzeiten bei verschiedenen Datensätzen werden in Tabelle 4.1 wiedergegeben.

Datensatz	Anzahl Strukturen	Dreiecke der Szene	Dauer der Analyse
Hals1	36	87188	2h 45min
Hals2	20	139536	1h 05min
Leber	16	360524	44min
Knie	10	25224	37min

Tabelle 4.1: Laufzeiten bei der Generierung der Datenbasis. Die verwendeten Datensätze umfassen zwei Visualisierungen der Halsregion sowie jeweils eine Visualisierung der Leber und des Kniegelenks.

Die Viewport-Auflösung während der Akquirierung lag bei $400 * 400$ Pixeln. Die Kamerapositionen wurden an Längen- und Breitengraden orientiert und mit einem Abstand von vier Grad auf der Kugel verteilt. Daraus ergibt sich eine Anzahl von 4050 Kamerapositionen (90 pro Breitengrad, 45 pro Längengrad).

Bei diesen Parametern und der aktuellen Implementierung kommt es, wie in Abbildung 4.1 zu erkennen, zu sehr hohen Laufzeiten. Es handelt sich zwar um einen Vorverarbeitungsschritt, trotzdem sollte die Laufzeit möglichst niedrig sein, um eine Anwendung im alltäglichen Rahmen zu ermöglichen.

Es gibt mehrere Möglichkeiten die Laufzeit zu reduzieren. Die Anzahl der Kamerapositionen haben einen linearen Einfluss auf die Laufzeit des Verfahrens, da für jede Kameraposition die gleiche Auswertung der Buffer stattfindet. Hier zeigt sich deutlich die Notwendigkeit einer gleichmäßigen, ungehäufteten Verteilung der Kamerapositionen. Ausgehend von einem Ikosaeder als Grundkörper, erhält man nach 5 Unterteilungsschritten eine kugelähnliche Struktur mit 2562 Eckpunkten. Diese Eckpunkte, welche mit den Kamerapositionen gleichzusetzen sind, haben einen Abstand von ungefähr 4 Grad.

Dies bedeutet, dass bei etwa gleicher Genauigkeit eine Laufzeitersparnis von 37% erreicht werden kann. Des Weiteren ist der Abstand von vier Grad auf der umgebenden Kugel als sehr hohe Informationsauflösung anzusehen. Bei dem Halsdatensatz, wo häufig nur sehr schmale Sichtbereiche für bestimmte Strukturen vorhanden waren, hat eine Auflösung von zehn Grad ebenfalls vollständig befriedigende Ergebnisse erzielt. Dies reduziert die Zahl von notwendigen Kamerapositionen auf einer nicht gleichverteilten Kugel auf 648 was einer Laufzeitersparnis von 84% entspricht. Bei einem Ikosaeder würden sich nach vierfacher Unterteilung 646 Positionen bei einem Abstand von ca. 8 Grad ergeben, d.h. eine höhere Genauigkeit bei ähnlicher Laufzeitersparnis.

Die Viewport-Auflösung von $400 * 400$ Pixeln ist ebenfalls als relativ hoch anzusehen. Diese Auflösung eignet sich für die Hals-Datensätze, da dort einige Lymphknoten mit einer sehr geringen räumlichen Ausdehnung¹ vorliegen. Bei den anderen Datensätzen sind die Strukturen deutlich größer. Dort können auch mit einer Viewport-Auflösung von $200 * 200$ Pixeln noch ausreichende Aussagen über den Grad ihrer Sichtbarkeit getroffen werden. Dies hat eine Laufzeiterparnis von 75% zur Folge, da anstatt 160.000 Pixelpositionen nur noch 40.000 Pixelpositionen ausgewertet werden müssen. Die erwarteten Laufzeiten nach Umsetzung aller hier beschriebenen Anpassungen werden in Tabelle 4.2 beschrieben.

Datensatz	Anzahl Strukturen	Dreiecke der Szene	Dauer der Analyse
Hals1	36	87188	30min
Hals2	20	139536	14min
Leber	16	360524	6min
Knie	10	25224	5min

Tabelle 4.2: Erwartete Laufzeiten nachdem die Viewport-Auflösung (für Leber und Knie $200 * 200$) angepasst und die Positionsverteilung auf der umgebenden Kugel (vierfach unterteilter Ikosaeder) optimiert wurde.

Die Datensätze haben eine hohe geometrische Komplexität. Trotzdem wird die Laufzeit des Verfahrens aufgrund der effektiven, hardwarebasierten Darstellung der Visualisierungen davon nur zu einem sehr geringen Grad beeinflusst. Die Laufzeit richtet sich primär nach der Anzahl n der Strukturen. Diese hat einen annähernd quadratischen Einfluss. Es entstehen im aufwändigsten Fall pro Pixelposition Listen der Länge n , welche sortiert und ausgewertet werden müssen. Die Sortierung wird durch den *Quicksort*-Algorithmus mit einer mittleren Laufzeit von $O(n * \log(n))$ durchgeführt. Die Auswertung hat ebenfalls einen tendenziell quadratischen Laufzeitcharakter (siehe Algorithmus 1), da bei der Auswertung einer Struktur die durch sie verursachte Verdeckung an den entsprechenden Matrix-Positionen der verdeckten Strukturen repräsentiert werden muss.

Die zurzeit vorliegende Implementierung ist bezogen auf den benötigten Speicher sehr effektiv, da die während der Akquirierung gewonnenen Informationen direkt in die Ergebnismatrizen übertragen werden. Speicherintensivere Zwischenrepräsentationen der gewonnenen Informationen könnten eventuell zur weiteren Reduzierung der Laufzeit genutzt werden.

Der grundsätzliche, quadratische Laufzeitcharakter der Akquirierung bleibt jedoch erhalten, da stark abstrahiert betrachtet die Lage jeder Struktur mit der der anderen Strukturen verglichen werden muss, um Aussagen über ihre Sichtbarkeit und die verdeckenden Strukturen treffen zu können. Im Gegenzug erhält man Sichtbarkeits- bzw.

¹Auf den zugrunde liegenden Volumendatensatz bezogen entspricht deren Ausdehnung ca. 2-3 Voxel

Verdeckungsinformationen, welche einen allgemeinen und globalen Charakter besitzen. Es wird keine Voreinschränkungen der Ergebnisse durchgeführt und die Ergebnisse jeder Struktur stehen mit denen aller anderen Strukturen in Beziehung.

4.2.3 Berechnung der Kamerapositionen

Das Auswerten der Ergebnismatrizen, das darauf aufbauende Erzeugen der einzelnen Parameterfelder und die Berechnung der finalen Ergebnisposition werden im Modul `METKCalcCamPos` realisiert. Abbildung 4.4 zeigt eine typische Netzwerkkonfiguration, in der dieses Modul zum Einsatz kommt. Im Folgenden wird geklärt, woher das Modul alle für die Berechnung notwendigen Informationen erhält. Des Weiteren wird auf die interne Repräsentation der Parameterfelder, ihre Berechnung und die Speicherung der Ergebnisse eingegangen.

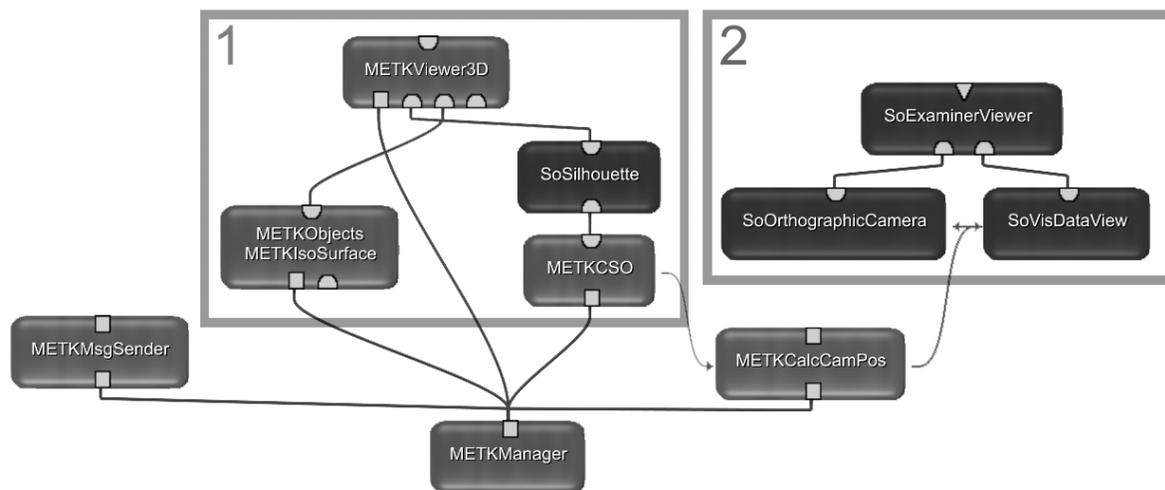


Abb. 4.4: Beispiel-MEVISLAB-Netzwerk für die Verwendung von `METKCalcCamPos`: Der Bereich 1 umfasst Module, die es ermöglichen per Mausklick eine Struktur auszuwählen, worauf diese mit Hilfe einer farbigen Silhouette hervorgehoben wird. Über eine Feldverbindung wird der Name der Struktur an `METKCalcCamPos` übergeben. Bereich 2 enthält Module zur Darstellung der Parameterfelder.

Informationsquellen

Für das Erzeugen der Parameterfelder und die Berechnung der Kameraposition sind viele Informationen aus verschiedenen Quellen von Bedeutung. Die Basis der Berechnung bilden die Ergebnismatrizen aller potentiellen Kameraposition. Wie bereits erläutert liegen diese in der binären Datei „vis.dat“ vor. Wird im METK ein Fall geladen, wird

`METKCalcCamPos` der Pfad der zum Fall gehörenden „vis.dat“ übergeben damit diese geladen werden kann. In den einzelnen Ergebnismatrizen werden die Strukturen in Form von IDs voneinander unterschieden.

Wird über das METK eine Anfrage bezüglich einer günstigen Kameraposition für eine bestimmte Struktur gestellt, wird die Struktur mit dem Namen bezeichnet, unter dem sie auch im `METKManager` vorliegt. Damit die Ergebnismatrizen bezüglich dieser Struktur korrekt ausgewertet werden können, muss ein Zusammenhang zwischen den IDs und den Strukturnamen hergestellt werden. Die geschieht durch die Informationen in der Datei „ids.dat“.

Fallspezifische Informationen werden direkt aus dem `METKManager` gewonnen. Dort ist z.B. die aktuelle Wichtigkeit aller Strukturen des Falls hinterlegt. Diese kann sich während der Laufzeit ändern und ist somit nicht in der Datenbasis aufgrund der Forderung nach Allgemeingültigkeit hinterlegt.

Eine dritte Informationsklasse stellen der Name der ausgewählten Struktur und die Wichtungen der einzelnen Parameterfelder dar. Durch diese Informationen nimmt der Nutzer Einfluss auf die Ergebnisse des Verfahrens zur Kamerapositionierung. Es wurden zwei Arten der Informationsübermittlung an das Modul realisiert.

Zum einen durch ein grafisches Interface, dass es dem Nutzer ermöglicht, den Namen der Struktur einzutragen² und über Regler die einzelnen Wichtungen zu definieren. Soll die Kamerapositionierung nicht manuell durch einen Nutzer, sondern durch eine andere Applikation mit den notwendigen Parametern versorgt werden, ist diese Art der Informationsübermittlung umständlich.

Deswegen wurde in das Modul `METKCalcCamPos` ein `METKMsgReceiver` integriert. Dadurch ist es möglich, auf Nachrichten zu reagieren, welche durch einen `METKMsgSender` angesandt wurden zu reagieren. Die Kommunikation findet dabei über den `METKManager` statt. Auf diese Weise können der Strukturname und die Parameterwichtungen in Form eines Strings an `METKCalcCamPos` übergeben werden. Ein weiterer Vorteil dieser Form der Informationsübermittlung ist die Möglichkeit, durch ein *Flag* dem Sender mitzuteilen, wann die gesendete Information verarbeitet wurde, d.h. wann eine Ergebnisposition vorliegt.

Repräsentation der Parameterfelder

Für die Berechnung der Kameraposition spielen die Parameterfelder eine zentrale Rolle. Sie können als Bilddaten interpretiert werden. Die ML-Module bieten viele Möglichkeiten Bilddaten zu repräsentieren und weiterzuverarbeiten. Allerdings stellen Parameterfelder spezielle und gut eingrenzbare Anforderungen an die Art ihrer

²In Kombination mit dem *Picking Tool* `METKCSO` kann die Auswahl in Form eines Mausklicks auf die gewünschte Struktur in der Visualisierung noch weiter vereinfacht werden

Repräsentation. Die Entwicklung einer eigenen Bildklasse `CmnImage` erschien demzufolge einfacher und sinnvoller, als die generisch konzipierten Bildklassen der ML-Bibliothek an die auftretenden Anforderungen anzupassen.

In der Klasse `CmnImage` liegen Parameterfelder als Float-Arrays vor. Es werden Addition und Multiplikation als arithmetische Operationen auf diesen Arrays zur Verfügung gestellt. Weitere arithmetische Operationen werden für die Berechnung der Kamerapositionen nicht benötigt. Neben typischen Funktionen wie dem Normieren und Invertieren der Parameterfelder wurden speziell an die zugrunde liegende Kugelgeometrie angepasste Distanztransformationen für die Erzeugung der Stabilitäts- und Vorzugsblickwinkelfelder implementiert. Dies geschieht durch eine wiederholte Dilatation und eine abstandsbezogene Wertung der neu hinzugefügten Punkte. Der Kugelgeometrie erfordert es, dass an einem Rand des Parameterfeldes die Dilatation nicht stoppt, sondern, wenn möglich, auf der gegenüberliegenden Seite fortgeführt wird.

Auch wenn effektiverer Implementierungen denkbar sind, hat sich diese Art der Distanzberechnung auch in Sonderfällen als stabil erwiesen. Solch ein Sonderfall tritt z.B. auf, wenn die Region für die ein Stabilitätsmaß errechnet werden muss die Kugel in einige Richtungen vollständig umschließt. Auf dem Parameterfeld präsentiert sich eine solche Region als nicht abgeschlossen (siehe Abb. 4.5). Verfahren die für die für die Berechnung der inneren Distanz den Abstand vom Rand der Region in x - und y -Richtung benötigen, können in diesem Fall kein korrektes Distanzmaß ermitteln.

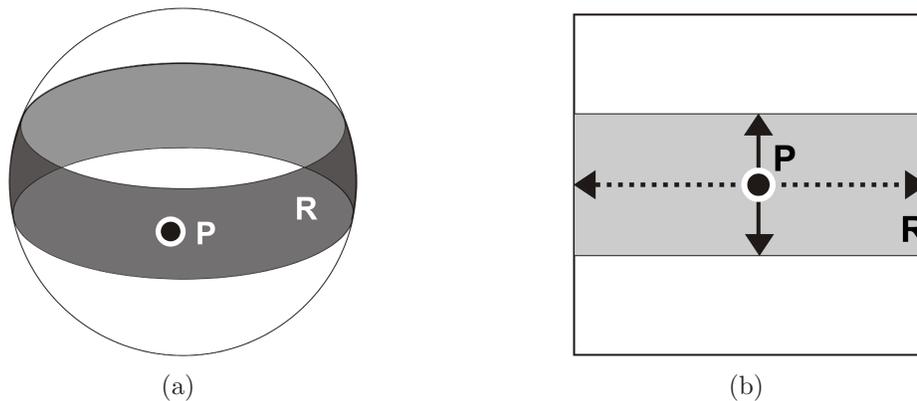


Abb. 4.5: Der Punkt P wird als stabiler bewertet, je weiter er von allen Rändern der Region R entfernt liegt (a). Wird das Stabilitätsmaß jedoch so berechnet, dass von der Existenz von Rändern in allen Richtungen ausgegangen wird, können für P keine korrekten Stabilitätsinformationen ermittelt werden (b). In äquatorialer Richtung ist Region R nicht abgeschlossen, d.h. dort liegen keine Abstandsinformationen zu einem Rand vor (gestrichelte Linie).

Berechnung der Parameterfelder und der Ergebnisposition

Die verschiedenen Parameterfelder können in zwei Klassen unterteilt werden. Zum einen in Parameterfelder, deren Erscheinungsbild sich direkt nach den Szenengegebenheiten richtet (Szenenfelder) und zum anderen in auf der Basis von Metainformationen berechnete Parameterfeldern (Metafelder).

Das Parameterfeld für die Sichtbarkeit ist ein typisches Beispiel für ein Szenenfeld. Die Informationen für dieses Parameterfeld werden direkt aus der im Vorverarbeitungsschritt generierten Datenbasis gewonnen. Die darin enthaltenen Ergebnismatrizen enthalten die notwendigen Informationen für alle Positionen auf dem Parameterfeld. Jede Ergebnismatrix wird an den für die ausgewählte Struktur relevanten Stellen ausgewertet. Das Ergebnis ist ein skalarer Wert, der an die zur Matrix gehörende Position im Parameterfeld übergeben wird. Im Falle der Sichtbarkeit errechnet sich dieser skalare Wert aus den in der Matrix enthaltenen Informationen über die Anzahl aller Strukturpixel und der Anzahl der sichtbaren Pixel einer Struktur. Es gibt so viele Ergebnismatrizen wie es Positionen auf einem Parameterfeld gibt. Da alle relevanten Informationen über die Verteilung der Kameraposition auf der Kugel bekannt sind, kann der Zusammenhang zwischen einer Ergebnismatrix und einer Position im Parameterfeld einfach hergestellt werden.

Ein Beispiel für ein Metafeld ist das Parameterfeld Kameraabstand. Die skalaren Werte dieses Parameterfeldes stammen nicht aus der Datenbasis, sondern werden auf der Basis der Metainformation über die Ausgangsposition der Kamera berechnet. Über ein Distanzmaß werden die Positionen im Parameterfeld mit zunehmendem Abstand von dieser Ausgangsposition entsprechend abgewertet.

Es gibt auch Mischformen. Das Parameterfeld für die kontextbezogene Wichtigkeit basiert auf Metainformationen über die aktuelle Wichtigkeit einer Struktur und auf Informationen über die Anzahl der Pixel mit denen sie eine andere Struktur verdeckt, welche aus der Datenbasis stammen. Die Parameterfelder für das Stabilitätsmaß stellen ebenfalls eine Sonderform dar. Sie benötigen ein berechnetes Parameterfeld damit darauf aufbauend die Stabilität berechnet werden kann. Hier werden also schon ausgewerteten Informationen aus der Datenbasis in Form eines Parameterfeldes und Metainformationen in Form eines Grenzwertes für die Mindestgüte der resultierenden Stabilitätsregionen genutzt.

Um die Ergebnisposition zu ermitteln, werden alle Parameterfelder entsprechend der Wichtungsfaktoren skaliert und addiert. Die Position des im resultierenden Summenfeld enthaltenen Maximums entspricht dem Längen- und Breitengrad der als am günstigsten bewerteten Kameraposition auf der umgebenden Kugel.

Alle für die weitere Verwendung relevanten Informationen werden in `METKManager` hinterlegt. Dazu wird ein Datenobjekt mit der `ObjectID` „CameraSolver“ angelegt. In ihm sind unter anderem die berechnete Kameraposition und Orientierung, die ausgewählte

Struktur sowie alle verdeckenden Strukturen und der Grad der durch sie verursachten Verdeckung gespeichert (siehe Abb. 4.6).

Object/Layer/Info	Value
ApplicationObject	
CameraSolver	
Calculation	
Success	TRUE
Camera	
Height	1.23
Orientation	0.096306 0.784348 0.612800 2.991391
Position	-8.419730 -128.001083 -223.108154
Global	
Id	CameraSolver
Info	
BestAlpha	256
BestBeta	88
Current	Structure_TumorPart_1
Occluder	Structure_Bones 0.999123 Structure_Pharynx 0.022807 Structure_Trachea 0.022807
WorstAlpha	64
WorstBeta	156
CaseObject	
Collections	
Communication	

Abb. 4.6: Eine Ausschnittsdarstellung aus dem `ObjInspector`. Damit ist es möglich sich einen Überblick über alle im `METKManager` hinterlegten Informationen zu verschaffen. Deutlich sind die einzelnen Informations-*Layer* des „CameraSolver“-Datenobjektes zu erkennen.

Update-Verhalten der Parameterfelder

Wird ein neuer Fall geladen und eine bestimmte Struktur ausgewählt, um für sie eine günstige Kameraposition zu ermitteln, müssen alle Parameterfelder berechnet werden. Im weiteren Verlauf der Anwendung ist es je nach Art der sich verändernden Bedingungen meist nicht mehr notwendig, alle Parameterfelder neu zu berechnen. Die nachfolgende Tabelle beschreibt, bei welchen Ereignissen welche Parameterfelder neu berechnet werden müssen.

Ereignis	zu aktualisierende Parameterfelder
Neuer Fall wird geladen	Alle Parameterfelder; Summenfeld
Wichtung eines Parameterfeldes wird geändert	Summenfeld
Grenzwert eines Stabilitätsmaßes wird geändert	entsprechendes Stabilitätsfeld; Summenfeld
Die Ergebnisposition entspricht nicht mehr der Ausgangsposition der Kamera	Kamerafeld
Die Wichtigkeit mind. einer Struktur ändert sich	Wichtigkeitsfeld; Stabilitätsfeld für Wichtigkeit; Summenfeld
Der Vorzugsblickwinkel ändert sich	Blickwinkelfeld; Summenfeld
Eine andere Zielstruktur wird gewählt	Alle Parameterfelder außer Blickwinkelfeld und Kamerafeld; Summenfeld

Wie zu erkennen ist, wird bei fast jedem Ereignis das Summenfeld neu berechnet. Dank der effektiven Realisierung der Parameterrepräsentation durch die Klasse `CmnImage` ist dies in Echtzeit möglich. Das Verfahren kann spontan auf sich ändernde Bedingungen reagieren und ist somit für den interaktiven Einsatz geeignet. Natürlich kann diese Interaktivität nur bis zu einer gewissen Größe und Anzahl der Parameterfelder gewährleistet werden. Bei der schon als hoch zu betrachtende Zahl von 18 Parameterfeldern mit je 4050 darin enthaltenen Kamerapositionen war eine interaktive Anpassung der Wichtungparameter auf einem durchschnittlichen System³ immer noch problemlos möglich.

Exploration der Parameterfelder

Mithilfe des INVENTOR-Moduls `SoVisDataView` ist es möglich, die Parameterfelder grafisch darzustellen. Für die Berechnung der Ergebnisposition hat dies keine Bedeutung. Dieses Modul kann aber z.B. genutzt werden, um beim Einbinden neuer Parameter deren Einfluss auf die Ergebnisposition zu untersuchen oder auch eventuelle Zusammenhänge zwischen verschiedenen Parametern zu erkennen.

Die Darstellung der Parameterfelder auf drei verschiedene Arten möglich: zweidimensional, dreidimensional und in Form einer Kugel (siehe Abb. 4.7). Gerade die letztgenannte Darstellungsform ist gut geeignet, um die Ausdehnung eines Bereichs im Parameterfeld korrekt einschätzen zu können. Als zusätzliche Informationen wird die Ergebnisposition auf jedem Parameterfeld hervorgehoben dargestellt. So lässt sich optisch abschätzen, inwieweit ein bestimmter Parameter durch diese Position befriedigt wird. Mit Hilfe optional darstellbarer Achsen kann bei der Darstellung in Form einer Kugel der räumliche Zusammenhang zwischen der Visualisierung und den Parameterfeldern verdeutlicht werden.

³INTEL CELERON CPU mit 2.8 Ghz, 1GB RAM, NVIDIA GEFORCE 6200 TC Grafikkarte

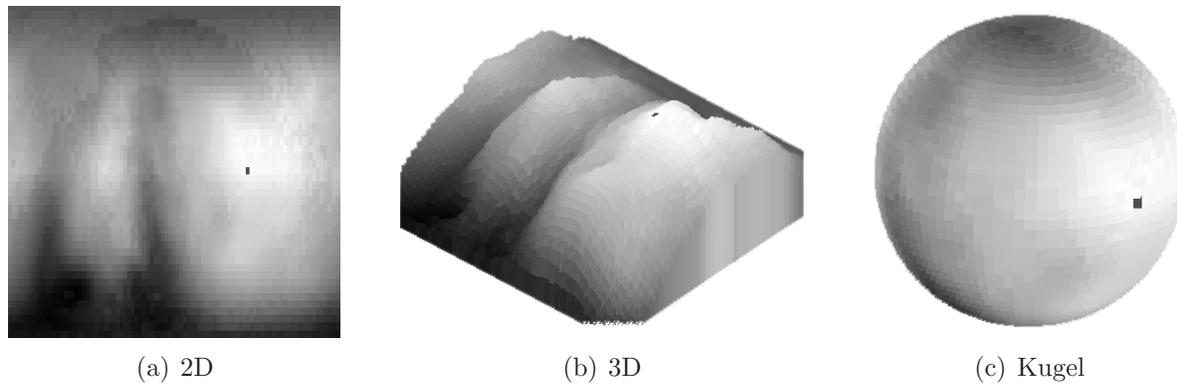


Abb. 4.7: Darstellungsformen für die Exploration von Parameterfeldern. Die Markierung im Feld gibt die als maximal bewertete Position an.

4.3 Zusammenfassung

Das Verfahren zur automatischen Kamerapositionierung gliedert sich in zwei große Teilschritte: die Generierung der Datenbasis und die auf diesen Informationen aufbauende Berechnung einer Ergebnisposition. Für beide Teilschritte wurden in MEVISLAB Module entwickelt. Dabei wurde besonderen Wert auf die Integration in das METK gelegt, um die Ergebnisse für weiterführende Anwendungen auf einfache Weise zur Verfügung stellen zu können.

Sich aus den Besonderheiten der Entwicklungsumgebung ergebende Probleme konnten gelöst werden. Zusammenfassend kann gesagt werden, dass bezogen auf die im Konzept formulierten Anforderungen die Teilschritte des Verfahrens erfolgreich umgesetzt werden konnten.

5 Ergebnisse

In diesem Kapitel werden die Ergebnisse der automatischen Kamerapositionierung präsentiert. Es wird anhand einiger Beispiele auf den Einfluss der verschiedenen Parameter eingegangen. Des Weiteren werden die Resultate einer informellen Evaluierung, basierend auf den Ergebnissen verschiedener Parameterkonfigurationen, vorgestellt und ausgewertet.

5.1 Anwendungsbeispiele

Das Verfahren wurde auf die drei in der Anforderungsanalyse vorgestellten Visualisierungen der Halsregion, des Knies und der Leber angewandt. Der Fokus wird auf typische Anwendungsszenarien bei der Exploration dieser Visualisierungen gelegt.

Bei der Visualisierung der Halsregion sind der darin enthaltene Tumor und die Lymphknoten von Interesse. Die Gelenkknorpel und der Meniskus spielen bei der Exploration der Knie-Visualisierung eine große Rolle und bei der Leber-Visualisierung wird besonderes Augenmerk auf den darin enthaltenen Tumor gelegt. Alle drei Visualisierungen stellen spezielle Anforderungen an das Verfahren zur Kamerapositionierung.

Initial wird das Verfahren auf die drei Visualisierungen mit einer gleichmäßigen Wichtung aller Parameter angewendet, d.h. es gilt:

$$p_{vis} = p_{vis_sta} = p_{imp} = p_{imp_sta} = p_{num} = p_{ent} = p_{pref} = p_{cam} = p_{dis} = 1$$

Als Vorzugsblickwinkel wurde der Äquatorialbereich auf der umgebenden Kugel definiert. Des Weiteren wird an einigen Beispielen gezeigt, wie durch eine unterschiedliche Parameterwichtung die Ergebnisse beeinflusst werden können.

5.1.1 Hals-Visualisierung

Abbildung 5.2 zeigt die aus den Ergebnispositionen resultierenden Bilder (Ergebnisbilder) für den Tumor und zufällig ausgewählte Lymphknoten. Die resultierenden Kamerapositionen stellen im Sinne der Anforderungsanalyse sehr gute Ergebnisse dar. Die

ausgewählten Strukturen sind gut sichtbar und vom Vorzugsblickwinkel wird nur abgewichen, wenn dies aufgrund der Sichtbarkeitsverhältnisse notwendig ist.

Auch wenn sich die kontextbezogenen Wichtigkeiten und somit die Darstellung der Strukturen ändert, liefert das Verfahren bei gleichmäßiger Wichtung aller Parameter Ergebnisse, die sich den veränderten Sichtbarkeitsverhältnissen anpassen und den Blick auf die Zielstrukturen ermöglichen (siehe Abb. 5.3).

Für die Beurteilung des Tumors sind strukturbezogene Informationen von Interesse. Die Entropie gibt an, von welchem Blickwinkel aus die meisten derartigen Informationen präsentiert werden. In der aktuellen Implementierung wird die Entropie mit der projizierten Fläche einer Struktur gleichgesetzt. Wie in Abbildung 5.4 verdeutlicht wird, kann durch die höhere Wichtung der Entropie eine bessere Darstellung erreicht werden.

Wird die Entropie allerdings zu hoch gewichtet, kann der Effekt ins Gegenteil umschlagen, wenn die ausgewählte Struktur aus einem sehr ungewöhnlichen Blickwinkel präsentiert wird. Des Weiteren ist die projizierte Fläche keine Garantie dafür, dass auch die informativste Seite einer Struktur präsentiert wird. Auch wenn ein komplexeres Entropiemaß im Endeffekt wahrscheinlich bessere Ergebnisse liefern würde, war es mit diesem einfachen Entropiemaß in vielen Fällen möglich, durch die Kombination mit anderen Parametern gute bis sehr gute Ergebnisse zu erzielen.

5.1.2 Knie-Visualisierung

Die Gleichwichtung aller Parameter liefert bei der Knie-Visualisierung teilweise akzeptable Ergebnisse für die Gelenkknorpel, jedoch nur unzureichende Ergebnisse für den Aussenmeniskus (siehe Abb. 5.5). Optimalere Ergebnisse sind möglich, wenn von der Forderung nach einem möglichst unverdeckten Blick (Parameter „Anzahl verdeckender Strukturen“ und „Sichtbarkeit“) und dem äquatorialen Vorzugsblickwinkel abgewichen wird.

Die Gelenkknorpel und der Meniskus sind auf engem Raum zwischen den Gelenkknöcheln angeordnet. Um bessere Aussagen über ihren Zustand treffen zu können, muss eine Blickrichtung gewählt werden, bei der der Nutzer durch die Knochenstruktur blickt. Hierbei spielen die Parameter „kontextbezogene Wichtigkeit“ und „Strukturentropie“ eine wichtige Rolle und werden dementsprechend hoch gewichtet.

Des Weiteren können die durch das Verfahren zur automatischen Kamerapositionierung zusätzlich gelieferten Informationen über die verdeckenden Strukturen genutzt werden, um diese gezielt auszublenden und so einen freien Blick auf die Zielstrukturen zu ermöglichen. Im Fall der Knie-Visualisierung wird der entsprechende Gelenkknöchel (*Tibula*) ausgeblendet, was in einem unverdeckten Blick auf den zugehörigen Gelenkknorpel resultiert. Wird dieser auch noch ausgeblendet, ist der Meniskus unverdeckt sichtbar (siehe Abb. 5.6).

5.1.3 Leber-Visualisierung

Die besondere Eigenschaft der Leber-Visualisierung besteht darin, dass der Tumor sich innerhalb der Leber befindet und fast vollständig von Lebergewebe verdeckt wird. Der Parameter „Sichtbarkeit“ liefert im Fall einer vollständigen Umschlossenheit keinen Beitrag zur Endposition.

Kommt es jedoch, wie in dieser Visualisierung der Fall ist, zu einer teilweisen Durchdringungen des umgebenden Gewebes durch den Tumor, kann die Kameraposition nachteilig durch den Parameter „Sichtbarkeit“ beeinflusst werden. Dann wird die Kameraposition bei gleichmäßiger Parameterwichtung so gewählt, dass dem Nutzer der Durchdringungsbereich präsentiert wird (siehe Abb. 5.7(a)). Dies ist nur sinnvoll, wenn das Lebergewebe sehr opak dargestellt wird (siehe Abb. 5.7(b)).

Da jedoch der Tumor bei der Exploration im Vordergrund steht, wird das Lebergewebe als weniger wichtig eingestuft und demzufolge abgeschwächt dargestellt. Damit das Verfahren zur Kamerapositionierung in diesem Fall eine günstige Position ermitteln kann, muss der Parameter „Sichtbarkeit“ abgewertet werden. Durch eine höhere Wertung des Vorzugsblickwinkels und der Entropie wird eine weitere Verbesserung der Tumordarstellung erreicht (siehe Abb. 5.7(c)).

5.1.4 Zusammenfassung

Bei der Hals-Visualisierung könnten ohne eine speziell angepasste Parameterwichtung zufrieden stellende Ergebnisse bei der automatischen Kamerapositionierung erzielt werden, wobei sich diese Wertung primär auf die Sichtbarkeit der jeweilig gewählten Zielstruktur bezieht. Durch eine Anpassung der Parameterwichtung konnten für den Tumor mehr Details präsentiert werden, was bei der Bewertung mit einem höheren Informationsgehalt der dargestellten Szene gleichgesetzt wird.

Die Visualisierungen des Knies und der Leber stellen aufgrund ihrer internen Struktur (Knie: Zielstrukturen auf engen Raum angeordnet, kaum sichtbar; Leber: Zielstruktur fast vollständig umschlossen) besondere Anforderungen an das Verfahren. Durch gezieltes Anpassen der Parameterwichtung und dem Einbeziehen der Verdeckungsinformationen ließen sich dort ebenfalls zufrieden stellende Kamerapositionen ermitteln.

Allerdings ist diese Bewertung, da sie vom Author vorgenommen wurde, als subjektiv anzusehen. Sie ist nur auf Sichtbarkeit und den Detailgrad bezogen. Da in der medizinischen Anwendung noch weitere Aspekte von Bedeutung sein können, müssen die Ergebnisse des Verfahrens in Form einer Evaluierung untersucht werden.

5.2 Evaluierung

Die automatische Kamerapositionierung soll Mediziner bei alltäglichen Explorationsaufgaben unterstützen. Wie sich allerdings im vorherigen Abschnitt gezeigt hat, gestaltet sich eine korrekte Einschätzung des Einflusses der verschiedenen Parameter auf das Endergebnis als sehr komplexe Aufgabe. Die Parameter beschreiben sehr abstrakte, informationstheoretische Inhalte und stehen in keinem direkten medizinischen Zusammenhang mit der zugehörigen Visualisierung. Das manuelle Auffinden einer günstigen Parameterkonfiguration ist für den Mediziner demzufolge eine unzumutbare Aufgabe, die den unterstützenden Charakter der automatischen Kamerapositionierung stark abschwächen würde.

Deshalb sollten dem Mediziner vordefinierte Parameterkonfigurationen (*Templates*) zur Verfügung stehen, die sich nach den jeweiligen Gesichtspunkten richten, unter denen er die Visualisierung exploriert. Derartige *Templates* würden in diesem Fall manuell erzeugt und müssten anhand von Evaluierungen überprüft werden. Eine erste derartige, informelle Evaluierung wurde durchgeführt und wird im Folgenden genauer beschrieben.

Der Anwendungsbereich wurde auf die Planung einer Hals-Operation eingeschränkt. Die zum Einsatz kommenden Visualisierungen richten sich in erster Linie an Mediziner, die mit der Anatomie der Halsregion vertraut sind. Da auch Nichtmediziner an der Evaluierung teilgenommen haben, werden die Aussagen der Mediziner einzeln betrachtet.

Es wird eine Parametrisierung vorgenommen, welche als günstig für die Anwendung im Bereich einer Hals-Visualisierung erscheint. Vergleichend dazu werden Parametrisierungen verwendet, die für eine Leber- und eine Knievisualisierung vielversprechende Ergebnisse liefern und eine Parametrisierung, deren Ergebnisse sich an der in [VÁZQUEZ et al., 2001] beschriebenen *Viewpoint Entropy* (VPE) orientieren (siehe Tabelle 5.1).

Parameter	Hals	Leber	Knie	VPE
Sichtbarkeit	0.3	0.0	0.0	0.5
Stabilität der Sichtbarkeit	0.3	0.0	0.0	0.5
Vorzugsblickwinkel	0.5	0.0	0.7	0.0
Ausgangsposition der Kamera	0.0	0.0	0.0	0.0
Strukturentropie	0.5	1.0	1.0	1.0
Anzahl verdeckender Strukturen	0.05	0.0	0.0	0.0
kontextbezogene Wichtigkeit	0.5	1.0	0.0	0.0
Stabilität der kontextbezogenen Wichtigkeit	0.5	0.5	0.0	0.0

Tabelle 5.1: Parameterkonfiguration für die Evaluierung. Alle Parameterwichtungen sind auf den Bereich $[0.0, 1.0]$ skaliert.

Die Annahme ist, dass aus dieser repräsentativen Sammlung von *Templates* das an der Halsvisualisierung orientierte die höchsten Bewertungen erhalten wird.

5.2.1 Fragebogen

Die Evaluierung wird mithilfe eines web-basierten Fragebogens durchgeführt. Bei den präsentierten Strukturen handelt es sich, ausgehend von zwei Datensätzen der Halsregion, um zufällig ausgewählte Lymphknoten und Tumoren. Auf diese werden die oben beschriebenen Parametrisierungen angewendet. Die Befragten erhalten zu den daraus resultierenden Bildern keine Informationen über die Parameter oder die ihre Wichtigkeit. In Abbildung 5.1 werden derartige Bilder für zwei ausgewählte Strukturen präsentiert.

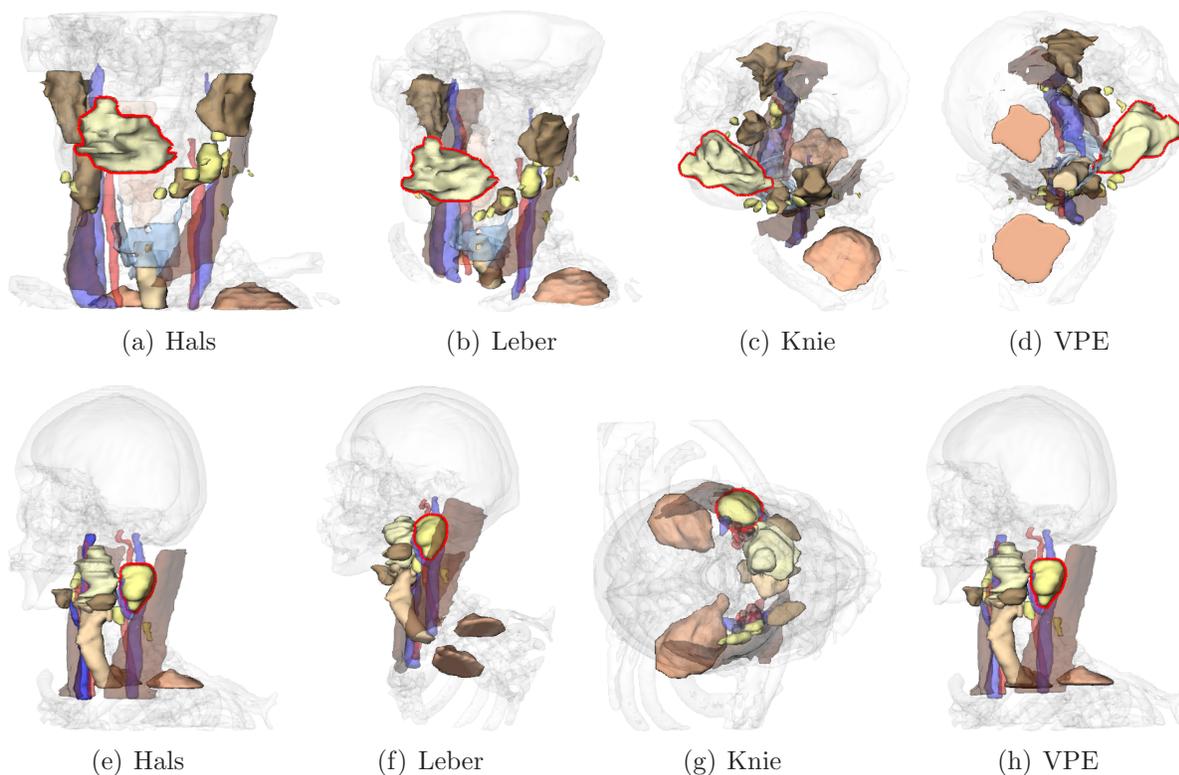


Abb. 5.1: Ergebnisbilder, nachdem die vier verschiedenen Parameterwichtungen auf Strukturen aus der Hals-Visualisierung (rot umrandet: oben - Tumor, unten - Lymphknoten) angewandt wurden.

Um die Befragten später klassifizieren zu können, werden zu Beginn des Fragebogens einige persönliche Informationen abgefragt: das Alter, das Geschlecht, der Bildungsstand sowie eine persönliche Einschätzung der Fähigkeit mit Computern und 3D-Anwendungen

umzugehen. Des Weiteren muss der Befragte Angaben zu seinem medizinischen Wissen, speziell im Bereich der Hals-Anatomie, machen.

Jede Seite des Fragebogens ist nach demselben Muster aufgebaut. Pro Seite werden die aus den unterschiedlichen Parametrisierungen resultierenden Bilder zu jeweils einer Struktur präsentiert. Der Befragte muss angeben, wie gut die Struktur sichtbar ist und für wie aussagekräftig er den gewählten Blickwinkel hält. Für die Antwort steht ihm eine Fünf-Werte-Skala (- -, -, 0, +, ++) zur Verfügung, wobei „fünf“ der schlechtesten und „eins“ der besten Bewertung entspricht. Des Weiteren hat der Befragte die Möglichkeit, Kommentare zu jedem Bild zu hinterlegen.

5.2.2 Ergebnisse und Interpretation

Insgesamt lagen 44 komplett ausgefüllte Fragebogen vor, die evaluiert werden konnten. Für eine speziellere Analyse standen die Antworten von 18 (vier weibliche, 14 männliche) Befragten mit einem guten medizinischen Wissen zur Verfügung. Ihr mittleres Alter betrug 30 Jahre und sie hatten mittelmäßige Fähigkeiten in den Bereichen Computernutzung und 3D-Anwendung.

Die Ergebnisse führen zu der Erkenntnis, dass das an der Hals-Visualisierung orientierte *Template* adäquate Ergebnisse für alle entscheidenden Strukturen liefert (siehe Tabelle 5.2). Im direkten Vergleich zwischen diesem *Template* und dem VPE erweist sich das Hals-*Template* als überlegen. Seine Bewertung liegt mindestens 0.39 Punkte über der des VPE-*Templates*. Das Leber-*Template* liefert ebenfalls bessere Ergebnisse als das der VPE, obwohl es nicht für die Anwendung in der Hals-Visualisierung ausgelegt ist. Es wird angenommen, dass dies aus der Berücksichtigung der „kontextbezogenen Wichtigkeit“ resultiert.

Alle Teilnehmer	Hals	Leber	Knie	VPE
Sichtbarkeit	2.12	2.41	3.71	2.51
Einschätzbarkeit der Verhältnisse	2.43	2.88	3.52	3.01
Medizinische Experten				
Sichtbarkeit	1.78	2.16	3.48	2.18
Einschätzbarkeit der Verhältniss	2.07	2.58	3.19	2.63

Tabelle 5.2: Ergebnisse der Evaluierung: 44 Personen haben den Fragebogen vollständig ausgefüllt. 18 von ihnen besaßen ein gutes medizinisches Wissen. Der Bewertungsbereich lag zwischen 1.0 (sehr gut) und 5.0 (schlecht).

Es lässt sich somit zusammenfassend sagen, dass das im Rahmen dieser Arbeit entwickelten Positionierungsverfahren gute Ergebnisse liefert. Sogar wenn die Parametrisierung sich nicht nach den Gegebenheiten der Visualisierung richtet, ist es auf klassischen Positionierungsmaßen (VPE) basierenden Verfahren überlegen.

Diese Evaluierung kann als erster Schritt zur Verfahrensbewertung betrachtet werden. Sie beschränkt sich auf die Anwendbarkeit im Rahmen der Hals-Visualisierung. Des Weiteren werden aufgrund der relativ geringen Teilnehmerzahl zwar Tendenzen erkennbar, statistisch signifikante Ergebnisse benötigen jedoch eine höhere Anzahl an Bewertungen.

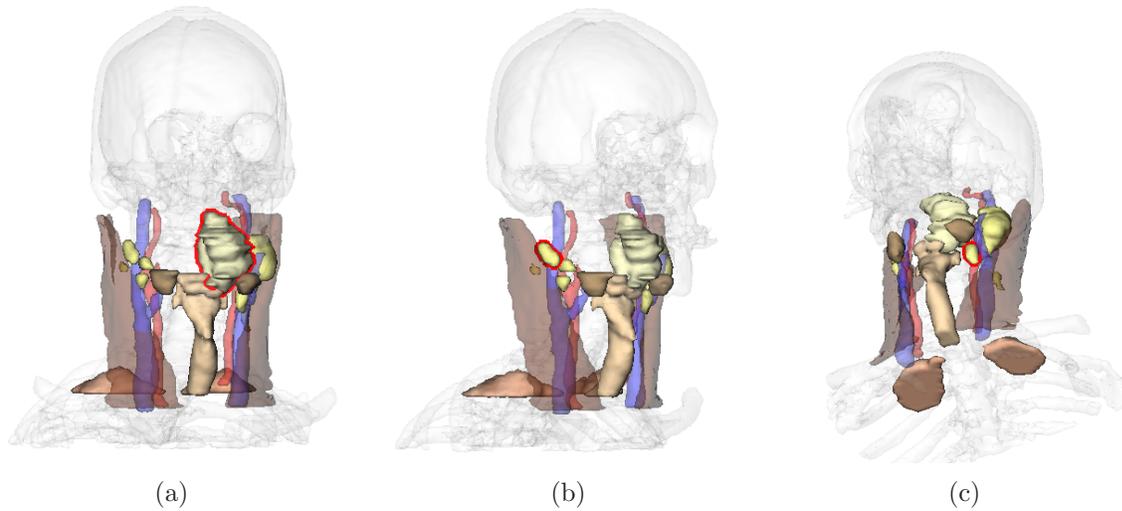


Abb. 5.2: Ergebnisbilder für die Hals-Visualisierung bei gleicher Parameterwichtung: In (a) wurde der Tumor ausgewählt, bei(b) und (c) liegt der Fokus auf zwei zufällig ausgewählten Lymphknoten. In (c) ist zu sehen, dass aufgrund der Sichtbarkeitsverhältnisse von der Vorzugsblickrichtung abgewichen wird, da andernfalls der Blick auf den Lymphknoten durch eine Struktur verdeckt wäre.

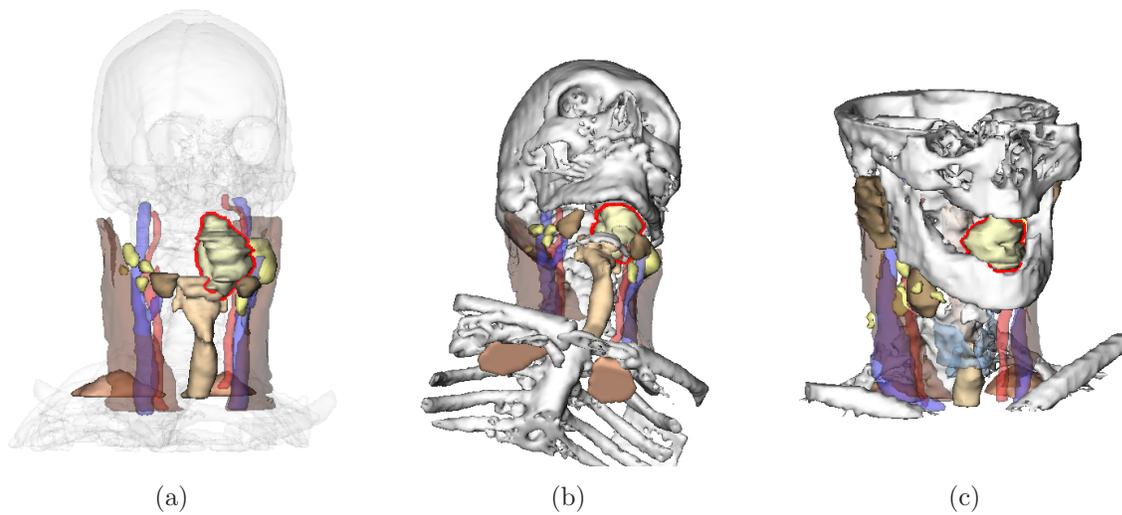


Abb. 5.3: Ergebnisbilder für die Hals-Visualisierung bei geänderten Wichtigkeitsverhältnissen: In (a) haben die Knochen eine sehr geringe Wichtigkeit. Allerdings sind sie für die Positionseinschätzung des Tumors von Bedeutung. Wird ihre Wichtigkeit erhöht und somit auch ihre optische Präsenz verstärkt (opakere Darstellung) ändert dies die Sichtbarkeitsverhältnisse der Szene. Der Tumor ist nur noch von unten sichtbar (b) (Hals2) oder wie in (c) durch den geöffneten Mund des Patienten (Hals1).

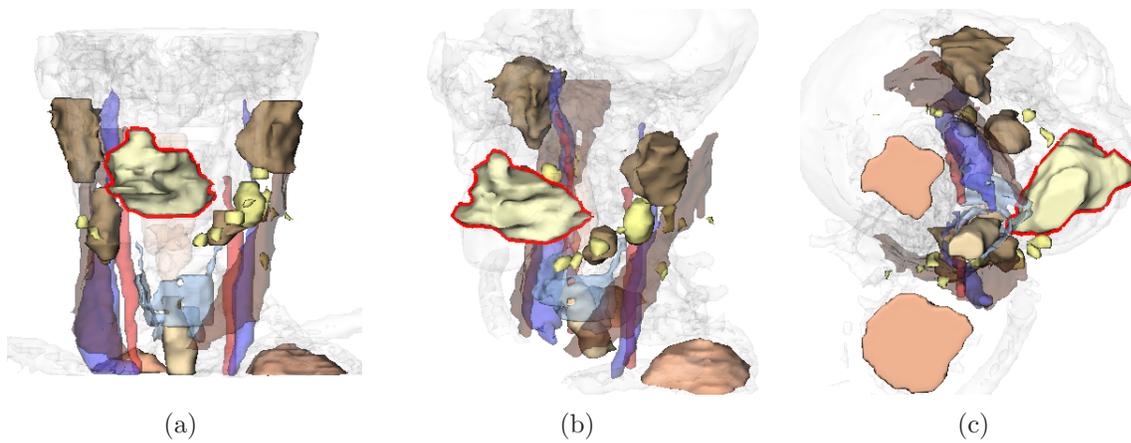


Abb. 5.4: Ergebnisbilder für die Hals-Visualisierung bei höher gewichteter Entropie: Durch eine leicht höhere Wichtung der Entropie wird der Tumor besser präsentiert (b). Seine Größe und Form sind besser einschätzbar als in der Ausgangsposition (a). Wird die Entropie jedoch zu hoch gewichtet, wird die Kameraposition nicht mehr durch den Parameter „Vorzugsblickwinkel“ stabilisiert und die Szene wird aus einem sehr ungewohnten Blickwinkel präsentiert.

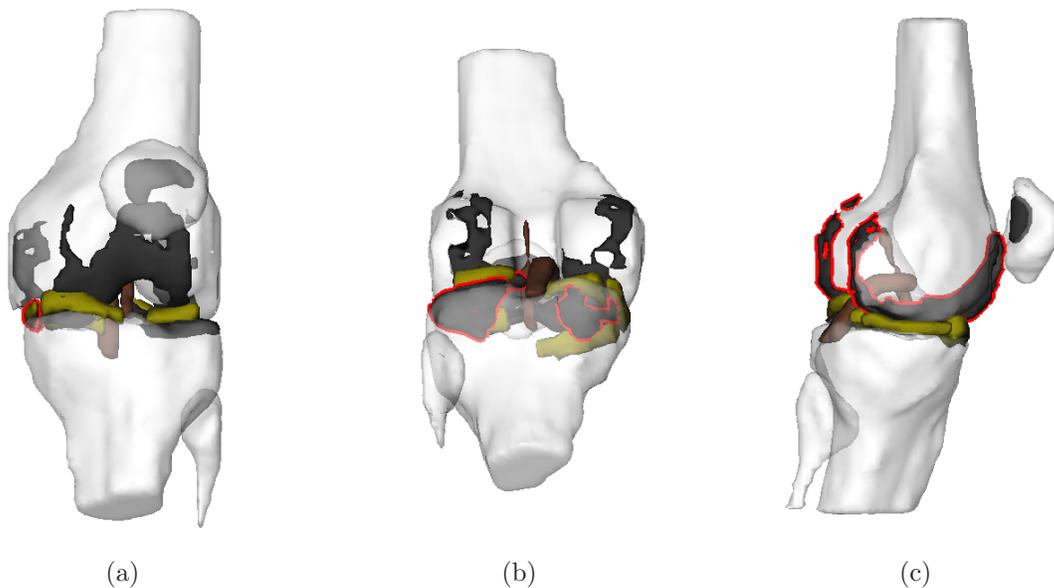


Abb. 5.5: Ergebnisbilder für die Knie-Visualisierung bei gleicher Parameterwichtung: Der Aussenmeniskus ist kaum erkennbar (a). Die Gelenkknorpel der *Tibula* (b) und des *Femur* sind zwar besser erkennbar, aber in allen drei Fällen kann durch eine veränderte Wichtung der Parameter eine deutliche Verbesserung erreicht werden.

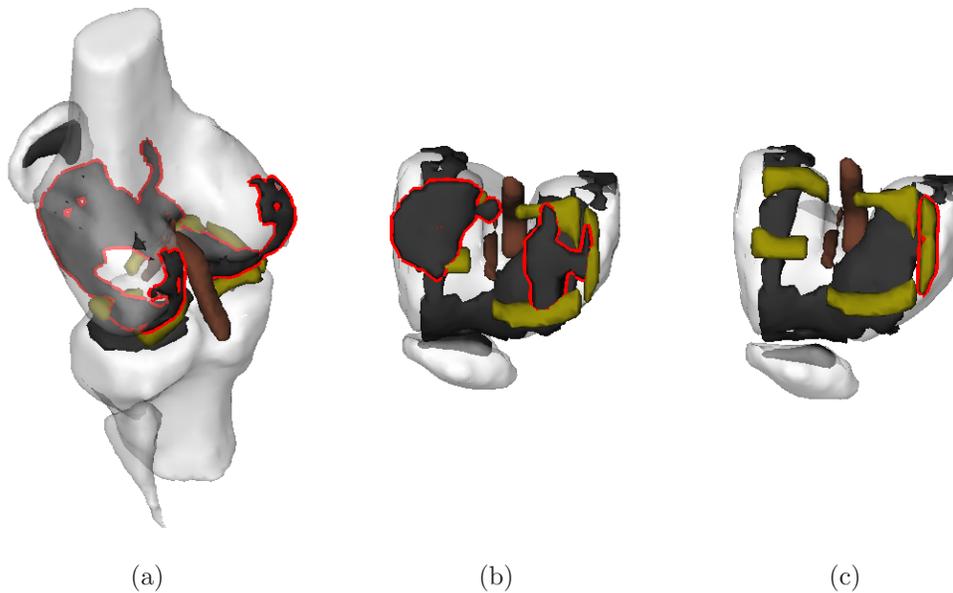


Abb. 5.6: Ergebnisbilder für die Knie-Visualisierung bei angepasster Parameterwichtung: Die Gelenkknorpel des *Femur* werden bei höherer Wichtung der Entropie besser erkennbar (a). Eine weitere Steigerung kann erreicht werden, wenn die aus dem Verfahren resultierenden Informationen über verdeckende Strukturen genutzt werden. Die Gelenkknorpel der *Tibula* sind sehr gut sichtbar, wenn diese vollständig ausgeblendet wird (b). Werden auch ihre Gelenkknorpel ausgeblendet, ist der Blick auf den Aussenmeniskus frei (c).

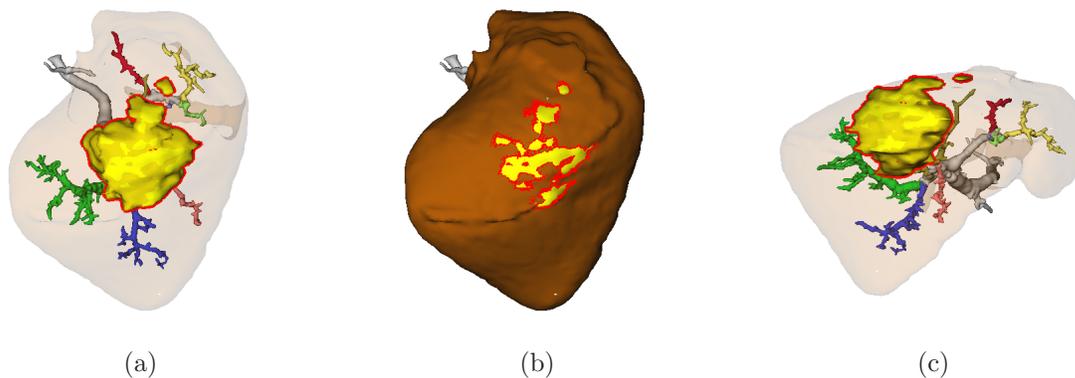


Abb. 5.7: Ergebnisbilder für die Leber-Visualisierung: Bei gleichmäßiger Parameterwichtung ergibt sich für den Tumor eine Ergebnisposition (a) welche nur bei opaker Darstellung des Lebergewebes (b) sinnvoll wäre. Durch eine Anpassung der Parameterwichtungen kann ein besseres Ergebnis erzielt werden (c). Wie dort zu erkennen ist, wurde die Kameraposition so gewählt, dass der Tumor nicht durch die ebenfalls als wichtig eingeschätzten Lebergefäße verdeckt wird.

6 Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein Verfahren zur automatischen Kamerapositionierung in dreidimensionalen, medizinischen Visualisierungen entwickelt und implementiert. Dabei wurde die Problemstellung in zwei Hauptbereiche differenziert: die Erzeugung aller notwendigen Informationen über die Sichtbarkeitsverhältnisse der dreidimensionalen Szene und parametrisierbare, automatische Positionierung der virtuellen Kamera auf der Basis dieser Informationen.

Es wurde eine umfangreiche Analyse anhand dreier Visualisierungen aus unterschiedlichen Domänen durchgeführt, um so die Anforderungen an die Datenbasis und die darauf aufbauende automatische Kamerapositionierung formulieren zu können. Die Generierung der Datenbasis wurde derart konzipiert, dass daraus resultierend ein breites Spektrum von Informationen über die Sichtbarkeitsverhältnisse und Verdeckungen in der Visualisierung vorliegt.

Das Ergebnis ist ein bildbasiertes Verfahren, das für eine Menge diskreter Kamerapositionen diese Informationen ermittelt. Dabei wurde eingehend untersucht, welche Parameter die Genauigkeit der Ergebnisse auf welche Weise beeinflussen und daraus entsprechende Schlussfolgerungen über die Wahl dieser Parameter getroffen.

Anhand der Ergebnisse der Anforderungsanalyse wurde eine Menge an Parametern definiert, an denen die Güte einer Kameraposition gemessen werden kann. Es wurde gezeigt, wie aus der generierten Datenbasis und anderen Quellen alle notwendigen Informationen gewonnen werden können, um diese Parameter quantitativ beschreiben zu können.

Darauf aufbauend wurde ein echtzeitfähiges Verfahren entwickelt, das es ermöglicht eine optimale Kameraposition auf der Basis der gewichteten Parameter zu ermitteln. Das Verfahren wurde auf die Datensätze aus der Anforderungsanalyse angewendet, wodurch der Einfluss der Parameter getestet und eine gute Qualität der Ergebnisse in erster Instanz bestätigt werden konnten. Zusätzlich wurde eine Evaluierung durchgeführt, um objektivere Aussagen über die Ergebnisqualität treffen zu können.

Beide Teilschritte des Verfahrens, die Generierung der Datenbasis und die automatische Positionierung der Kamera konnten entsprechend der an sie gestellten Anforderungen bezogen auf Genauigkeit und Laufzeiten in MEVISLAB umgesetzt werden. Die gewünschte Integration in das METK wurde vollständig erreicht, wodurch die

Ergebnisse für weiterführende Anwendungen im Bereich der Exploration medizinischer Visualisierungen effektiv zu Verfügung gestellt werden.

6.1 Weiterführende Ansätze und Verbesserungsvorschläge

Die aktuelle Implementierung umfasst bisher nur eine uniforme, an Längen- und Breitengraden orientierte Verteilung der Kamerapositionen. Da ein umfassendes Konzept für eine Gleichverteilung der Kamerapositionen beschrieben wurde, stellt deren Umsetzung keine komplizierte, aber im Sinne der Laufzeitoptimierung notwendige Verbesserung der Implementierung dar.

Auch wenn das Verfahren durch diese Verteilung der Kamerapositionen beschleunigt wird, werden die Szenengegebenheiten dabei nicht beachtet. Es ist denkbar, dass eine sich nach dem Aufbau der dreidimensionalen Szenen ausgerichtete Verteilung eine sehr hohe Genauigkeit bei gleichzeitig niedriger Laufzeit ermöglichen kann. Des Weiteren müsste keine Vorabschätzung über die Dichte, mit der die Kamerapositionen auf der Kugel verteilt werden, durchgeführt werden.

Ein möglicher Ansatz lässt sich folgendermaßen beschreiben: Es wird mit einer relativ geringen Anzahl an Kamerapositionen begonnen. Die sich von dort aus bietenden Sichtbarkeitsverhältnisse müssen mittels eines Maßes beschrieben werden. Auf der Basis dieser Maßzahl können Vergleiche zwischen den einzelnen Positionen vorgenommen werden. Unterschreiten sie bezogen auf ihre Ähnlichkeit einen vordefinierten Grenzwert, haben sich die Sichtbarkeitsverhältnisse von einer Position zur nächsten zu stark geändert. Das Hinzufügen weiterer Positionen zwischen den beiden Ausgangspositionen ist notwendig, um diese Änderungen zu erfassen.

Allerdings muss ein aussagekräftiges Maß gefunden werden, um Vergleiche zwischen verschiedenen Kamerapositionen durchführen zu können. In [BORDOLOI und SHEN, 2005] wird ein Maß für diese sogenannte „*View Liklyhood*“ vorgestellt. Ausgehend von der dort beschriebenen Verfahrensweise muss untersucht werden, inwieweit die dafür notwendigen Informationen aus der Datenbasis gewonnen werden können und, wenn dies nicht der Fall ist, um welche Informationen die Datenbasis erweitert werden muss.

Bisher enthält die Datenbasis nur statistische Informationen, was ausreichend für deren Anwendung bei der automatischen Kamerapositionierung ist. Jedoch wäre es möglich, während der Generierung der Datenbasis weitere Informationen zu gewinnen. So können beispielsweise Viewport-bezogene Informationen (z.B. die 2D-Bounding Box einer Struktur) gespeichert werden, um diese für spätere Anpassungen der Visualisierung während der Exploration (z.B. *Cut-Away*, Vergrößerungsgrad während der Darstellung, etc.) nutzbar zu machen.

Das Maß für die Entropie wird sehr einfach aus der Größe der projizierten Fläche einer Struktur gewonnen. Die Umsetzung eines komplexeren Maßes, welches beispielsweise Besonderheiten auf der Oberfläche einer Struktur einbezieht, wäre eine wünschenswerte Verbesserung.

Die Anordnung potentieller Kamerapositionen richtet sich nach der Steuerungsmetapher, die der Nutzer verwendet um die virtuelle Kamera durch die Szenen zu navigieren. Aktuell beschränkt sich das Verfahren auf die *Scene in Hand*-Metapher. Für spezielle Aufgaben, wie die Gefäßexploration, ist diese Steuerungsmetapher ungeeignet. Eine Anpassung des Verfahrens, um auch im Zusammenhang mit anderen Steuerungsmetaphern einsetzbar zu sein, erscheint als eine sinnvolle Weiterentwicklung.

Derzeit wird, mit Ausnahme der Positionierung für die Minimale-Distanz-Betrachtung, immer nur eine Struktur ausgewählt und für deren Darstellung eine Kameraposition ermittelt. Eine sinnvolle Erweiterung wäre das Einbeziehen mehrere Strukturen, wobei der kontextuelle Zusammenhang zwischen den Strukturen bei der Kamerapositionierung eine wichtige Rolle spielt. Es muss untersucht werden, inwieweit die derzeit über eine einzelne Struktur vorliegenden Informationen dabei genutzt werden können und inwieweit das Verfahren zur Generierung der Datenbasis an diese erweiterte Problemstellung eventuell angepasst werden muss.

Die für die fallbezogene Parametrierung genutzten *Templates* werden manuell erzeugt. Dieser Prozess könnte durch eine Automatisierung einfacher gestaltet werden. Die Szene könnte analysiert werden und daraus in Kombination mit Metainformationen eine korrekte Wichtung der Parameter automatisch bestimmt, oder zumindest tendenzielle Vorschläge geliefert werden. So würde es weniger erfahrenen Nutzern erleichtert werden, neue Templates für einen speziellen Anwendungsfall zu erzeugen.

Unabhängig davon, auf welche Weise die Parameterwichtung vorgenommen wird, so richtet sie sich nach der Aufgabenstellung, nicht jedoch nach den persönlichen Vorlieben des Mediziners. Hier wäre die Entwicklung eines Lernsystems denkbar. Während der Mediziner die Visualisierung exploriert wird analysiert, auf welche Weise er dies tut. Die daraus gewonnenen Erkenntnisse können in die Parameterwichtung einfließen und so kann dem Mediziner eine Kamerapositionierung vorgeschlagen werden, die nicht nur der Aufgabenstellung entspricht sondern zusätzlich seine persönlichen Vorlieben berücksichtigt.

Die automatische Positionierung der Kamera erleichtert die Exploration komplexer medizinischer Visualisierungen. Das im Rahmen dieser Arbeit entwickelte Verfahren liefert Ergebnisse, die in praktischen Anwendungen einsetzbar sind. Dies und die vielfältigen Möglichkeiten einer weiterführenden Entwicklung machen deutlich, dass ein wichtiger Beitrag im Bereich der Mensch-Computer-Interaktion geschaffen wurde.

Abbildungsverzeichnis

2.1	Vergleich zwischen CT und MRT	7
2.2	Gegenüberstellung von direkter und indirekter Volumenvisualisierung . .	9
2.3	Einfluss der Auflösung auf die Qualität von IDV und DV	10
2.4	Visualisierung im NECK DISSECTION PLANNER	12
2.5	Beispiele für Visualisierungen in medizinischen Atlanten	14
2.6	CAT: Eingabegerät für die Kamerasteuerung	15
2.7	Regionale Hervorhebungstechniken	17
2.8	Projektionsarten	20
2.9	Steuerungsmetaphern	23
2.10	Sichtbarkeitsanalyse des VISIBILITY SOLVERS	27
2.11	Sichtbarkeitsanalyse nach CHU	28
2.12	Arten der Szenenunterteilung bei Radiosity-Verfahren	29
2.13	Sichtbarkeitsanalyse im VISIBILITY SKELETON	31
2.14	<i>Mid-Level</i> -Ansatz für Viewpointquality	33
2.15	<i>High-Level</i> -Anwendungen bei der Positionsbewertung	34
2.16	Klassifizierung von Systemen zur Kamerapositionierung	36
3.1	Beispiele für Hals-,Leber- und Knievisualisierungen	43
3.2	Größenvergleich zwischen einem Lymphknoten und der vollständigen Halsvisualisierung	43
3.3	opake Visualisierung	44
3.4	Approximation durch einfache Hüllkörper	45
3.5	Abweichung vom Standardblickwinkel	46
3.6	Minimale Auflösung	51

3.7	Abstand der Kamerapositionen auf der Kugel	52
3.8	Häufung der Positionen an den Polen	53
3.9	Annäherung an eine Kugel	54
3.10	Beispiel für ein z -Buffer-Bild	55
3.11	Bufferauswertung pro Pixelposition	56
3.12	Matrix der Datenbasis	57
3.13	Schema: Ablauf der Szenenanalyse	59
3.14	Zusammenhang zwischen Geometrie und Parameterfeld	62
3.15	Einfluss der Wichtigkeit	63
3.16	Verdeckende Strukturen	64
3.17	Einfaches Entropiemaß	65
3.18	Vorzugsblickwinkel	66
3.19	Stabile und instabile Positionen	67
3.20	Schritte der Stabilitätsberechnung	68
3.21	Einfluss der Parameters Abstand	69
3.22	Beispiel für ein Abstands-Parameterfeld	69
3.23	Überbewertung der Flächen an den Polen	70
3.24	Sinus Parameter	70
3.25	Alternative Betrachterpositionen	72
3.26	Pfad auf der umgebenden Kugel für ein „Minimaler Abstand“-Positionierung	73
3.27	Ablauf der Positionsbestimmung für den minimalen Abstand	74
3.28	Ablauf der Berechnung der Ergebnisposition	77
4.1	Informationsfluss zwischen den Modulen	83
4.2	MEVISLAB-Netzwerk für die Generierung der Datenbasis	84
4.3	Internes Netzwerk im <code>SoVisibilitySolver</code>	84
4.4	Beispiel-MEVISLAB-Netzwerk für die Verwendung von <code>METKCalcCamPos</code>	92
4.5	Sonderfall für Stabilitätsberechnung	94
4.6	Datenobjekt für die Speicherung der Ergebnisse	96
4.7	Darstellungsformen für die Exploration von Parameterfeldern	98
5.1	Ergebnisbilder für verschiedenen Parameterkonfigurationen	103
5.2	Ergebnisbilder für die Hals-Visualisierung bei gleicher Parameterewichtung	106

5.3	Ergebnisbilder für die Hals-Visualisierung bei geänderten Wichtigkeitsverhältnissen	106
5.4	Ergebnisbilder für die Hals-Visualisierung bei höher gewichteter Entropie	107
5.5	Ergebnisbilder für die Knie-Visualisierung bei gleicher Parameterwichtung	107
5.6	Ergebnisbilder für die Knie-Visualisierung bei angepasster Parameterwichtung	108
5.7	Ergebnisbilder für die Leber-Visualisierung	108

Literaturverzeichnis

- Edward Angel. *Interactive Computer Graphics. A Top-Down Approach Using OpenGL*. Addison Wesley, Reading, Massachusetts, 4. Aufl., 2005.
- Ragnar Bade, Ivonne Riedel, Lars Schmidt, Karl J. Oldhafer, und Bernhard Preim. Combining Training and Computerized Planning of Oncologic Liver Surgery. In *Bildverarbeitung für die Medizin*, Seiten 409–413, Berlin, Heidelberg, New York, 2006. Springer Verlag.
- Ragnar Bade, Felix Ritter, und Bernhard Preim. Usability Comparison of Mouse-Based Interaction Techniques for Predictable 3d Rotation. In *Proceedings of the 5th International Symposium on Smart Graphics*, Lecture Notes in Computer Science, Seiten 138–150, Berlin, Heidelberg, New York, August 2005. Springer Verlag.
- William Bares, Scott McDermott, Christina Boudreaux, und Somying Thainimit. Virtual 3D Camera Composition from Frame Constraints. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, Seiten 177–186, New York, NY, USA, 2000. ACM Press.
- William H. Bares, Joel P. Gregoire, und James C. Lester. Realtime Constraint-Based Cinematography for Complex Interactive 3D Worlds. In *AAAI/IAAI*, Seiten 1101–1106, Menlo Park, CA, 1998. AAAI Press.
- Jiří Bittner und Peter Wonka. Visibility in Computer Graphics. *Environment and Planning B: Planning and Design*, 30(5):729–756, September 2003.
- Jim Blinn. Where Am I? What Am I Looking At? *IEEE Computer Graphics and Applications*, 8(4):76–81, 1988.
- Udepta Bordoloi und Han-Wei Shen. View Selection for Volume Rendering. In *IEEE Visualization*, Seite 62, Washington, DC, USA, 2005. IEEE Computer Society.
- Andreas Butz. Animation with CATHI. In *AAAI/IAAI*, Seiten 957–962, Menlo Park, CA, 1997. AAAI Press.
- David B. Christianson, Sean E. Anderson, Li wei He, David Salesin, Daniel S. Weld, und Michael F. Cohen. Declarative Camera Control for Automatic Cinematography. In *AAAI/IAAI*, Seiten 148–155, Menlo Park, CA, 1996. AAAI Press.

- Marc Christie und Eric Langu  nou. A Constraint-Based Approach to Camera Path Planning. In *Smart Graphics*, Seiten 172–181, Berlin, Heidelberg, New York, 2003. Springer Verlag.
- Marc Christie, Rumesh Machap, Jean-Marie Normand, Patrick Olivier, und Jonathan Pickering. Virtual Camera Planning: A Survey. In *Smart Graphics*, Seiten 40–52, Berlin, Heidelberg, New York, 2005. Springer Verlag.
- Gil W. Chu und Myung J. Chung. Autonomous selection and modification of camera configurations using visibility and manipulability measures. In *Journal of Robotic Systems*, Seiten 219–230, Chichester, UK, UK, 2002. John Wiley and Sons Ltd.
- Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, und Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. In *SIGGRAPH ’88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, Seiten 75–84, New York, NY, USA, 1988. ACM Press.
- Daniel Cohen-Or, Yiorgos L. Chrysanthou, Claudio T. Silva, und Durand Durand. A Survey of Visibility for Walkthrough Applications. In *IEEE Transactions on Visualization and Computer Graphics*, Bd. 09, Seiten 412–431, Los Alamitos, CA, USA, 2003. IEEE Computer Society.
- Frank Deinzer, Joachim Denzler, und Heinrich Niemann. Viewpoint Selection - A Classifier Independent Learning Approach. In *IEEE Southwest Symposium on Image Analysis and Interpretation*, Seiten 209–213, Washington, DC, USA, 2000. IEEE Computer Society.
- Joachim Diepstraten, Daniel Weiskopf, und Thomas Ertl. Interactive Cutaway Illustrations. In *Computer Graphics Forum*, Bd. 22, Seiten 523–532, Oxford, UK, 2003. Blackwell.
- Christina D  rge. Techniken zur interaktiven Hervorhebung von Objekten in medizinischen 3d-Visualisierungen. Diplomarbeit, Universit  t Bremen, 2002.
- Steven M. Drucker und David Zeltzer. Intelligent Camera Control in a Virtual Environment. In *Proceedings of Graphics Interface ’94*, Seiten 190–199, Banff, Alberta, Canada, 1994. Morgan Kaufmann.
- Steven M. Drucker und David Zeltzer. CamDroid: a system for implementing intelligent camera control. In *SI3D ’95: Proceedings of the 1995 symposium on Interactive 3D graphics*, Seiten 139–144, New York, NY, USA, 1995. ACM Press.
- Fr  do Durand, George Drettakis, und Claude Puech. 3D Visibility Made Visibly Simple: An Introduction to the Visibility Skeleton. In *Symposium on Computational Geometry*, Seiten 475–476, New York, NY, USA, 1997a. ACM Press.

- Frédo Durand, George Drettakis, und Claude Puech. The Visibility Skeleton: a Powerful and Efficient Multi-Purpose Global Visibility Tool. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, Bd. 31, Seiten 89–100, New York, NY, USA, 1997b. ACM Press.
- Martin Hachet, Pascal Guitton, und Patrick Reuter. The cat for efficient 2d and 3d interaction as an alternative to mouse adaptations. In *VRST '03: Proceedings of the ACM symposium on Virtual reality software and technology*, Seiten 225–112, New York, NY, USA, 2003. ACM Press.
- Horst K. Hahn, Florian Link, und Heinz-Otto Peitgen. Concepts for Rapid Application Prototyping in Medical Image Analysis and Visualization. In *SimVis*, Seiten 283–298, Magdeburg, 2003. SCS Publishing House e.V.
- Nicolas Halper. *Supportive Presentation for Computer Games*. Dissertation, Otto-von-Guericke-Universität Magdeburg, Germany, 2003.
- Pat Hanrahan, David Salzman, und Larry Aupperle. A rapid hierarchical radiosity algorithm. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, Seiten 197–206, New York, NY, USA, 1991. ACM Press.
- Karl Heinz Höhne, Bernhard Pflesser, Andreas Pommert, Kay Priesmeyer, Martin Riemer, Thomas Schiemann, Rainer Schuber, Ulf Tiede, Hans Frederking, Sebastian Gehrman, Stefan Noster, und Udo Schumacher. *VOXELMAN 3D-Navigator: Inner Organs. Regional, Systemic and Radiological Anatomy*. Springer-Verlag Berlin, Heidelberg, New York, 2003. DVD-ROM.
- K. Hillesland, B. Salomon, A. Lastra, und D. Manocha. Fast and simple occlusion culling using hardware-based depth queries. Forschungsbericht, University of North Carolina, 2002.
- Jana Hintze, Jeanette Cordes, Bernhard Preim, Ilka Hertel, Gero Strauß, und Uta Preim. Bildanalyse für die präoperative Planung von Neck Dissections. In *Bildverarbeitung für die Medizin*, Informatik aktuell, Seiten 11–15, Berlin, Heidelberg, New York, 2005. Springer Verlag.
- Arno Krüger, Christian Tietjen, Jana Hintze, Bernhard Preim, Ilka Hertel, und Gero Strauß. Interactive Visualization for Neck Dissection Planning. In *IEEE/Eurographics Symposium on Visualization (EuroVis)*, Seiten 295–302, Berlin, Heidelberg, New York, 2005. Springer Verlag.
- Philippe Lacroute und Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. *Computer Graphics*, 28(Annual Conference Series):451–458, 1994.
- Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

- Herbert Lippert. *Lehrbuch Anatomie*. Urban & Fischer, München, 5. Aufl., 2000.
- William E. Lorensen und Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, Seiten 163–169, New York, NY, USA, 1987. ACM Press.
- Thomas H. Massie und J. K. Salisbury. The PHANTOM Haptic Interface: A Device for Probing Virtual Objects. In *Proceedings of the ASME Winter Annual Meeting. Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Chicago, IL, November 1994.
- Hou Miao-le. The Basic Topology Model of Spherical Surface Digital Surface. In *IAPRS*, Bd. 25, Istanbul, 2004. ISPRS Congress, ISPRS.
- Heinz Morneborg. *Bildgebende Systeme für die medizinische Diagnostik*. Publicis MCD Verlag, Erlangen, 3. Aufl., 1995.
- Konrad Muehler, Ragnar Bade, und Bernhard Preim. Skriptbasierte Animationen fuer die Operationsplanung und Ausbildung. In *Bildverarbeitung für die Medizin, Informatik aktuell*, Seiten 296–300, Berlin, Heidelberg, New York, 2006. Springer Verlag.
- Bernhard Preim, Andreas Raab, und Thomas Strothotte. Coherent Zooming of Illustrations with 3D-Graphics and Text. In *Graphics Interface*, Seiten 105–113, Halifax, Nova Scotia, 1997. A K Peters.
- Bernhard Preim und Felix Ritter. Techniken zur interaktiven Hervorhebung von Objekten in medizinischen 3d-Visualisierungen. In *SimVis*, Seiten 187–200, Magdeburg, 2002. SCS Publishing House e.V.
- Andrew W. Rogers. *Textbook of Anatomy*. Churchill Livingstone, Oxford, 1992.
- Leonard J. Y. Schulman. *Communication in the presence of noise*. Dissertation, Cambridge University, Cambridge, MA, USA, 1992.
- J. Schulze und U. Lang. The parallelization of the perspective shear-warp volume rendering algorithm. In *Proceedings of the 4th Eurographics Workshop on Parallel Graphics and Visualization*, Seiten 61–69, 2002.
- Seemann, Luboldt, Bode, Haferkamp, Schäfer, Plinkert, Dammann, und Claussen. Hybride 3D-Visualisierung und virtuelle Endoskopie von Cochlea Implantaten. *RöFo: Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren*, 172: 238–243, 2000.
- Dmitry Sokolov und Dimitri Plemenos. Viewpoint quality and scene understanding. In *International Eurographics Symposium on Virtual Reality, Archaeology and Cultural Heritage*, Berlin, Heidelberg, New York, 2005. Springer Verlag.

- Dmitry Sokolov, Dimitri Plemenos, und Karim Tamine. Viewpoint quality and global scene exploration strategies. In *GRAPP 2006*, Setúbal, Portugal, February 2006. GRAPP.
- Kristina Stampe. Haptische Interaktion für die Planung endoskopischer nnh-Eingriffe. Diplomarbeit, Otto-von-Guericke-Universität - Dept. of Computer Science, Magdeburg, 2006.
- Wolfgang Stuerzlinger. Imaging all visible surfaces. In *Proceedings of the 1999 conference on Graphics interface '99*, Seiten 115–122, San Francisco, CA, USA, 1999. Morgan Kaufmann.
- Andreas Tappenbeck, Bernhard Preim, und Volker Dicken. Distanzabhängige Transferfunktionen für die medizinische Volumenvisualisierung. In *Bildverarbeitung für die Medizin*, Informatik aktuell, Seiten 307–311, Berlin, Heidelberg, New York, 2005. Springer Verlag.
- Seth J. Teller und Carlo H. Séquin. Visibility preprocessing for interactive walkthroughs. *Computer Graphics*, 25(4):61–68, 1991.
- TeraRecon. Volumepro® 1000: Real-Time 3D Volume Rendering Single Board Special Purpose Computer. Internetquelle: zuletzt besucht am 12.09., 2006. URL http://www.terarecon.com/downloads/products/datasheet_vp1000.pdf.
- Ulf Tiede, Thomas Schiemann, und Karl Heinz Höhne. Visualizing the visible human. *IEEE Comput. Graph. Appl.*, 16(1):7–9, 1996.
- Christian Tietjen, Jana Dornheim, Arno Krüger, Bernhard Preim, Ilka Hertel, und Gero Strauß. Computer Assisted Surgery Planning for Neck Dissections. In Stefan Weber, Frank Langlotz, Jürgen Bier, und Tim C. Lüth, Hrsg., *Computer Aided Surgery around the Head – 3rd International Symposium*, Bd. 17 of *Fortschrittberichte VDI*, Seiten 32–33, Düsseldorf, August 2005a. VDI Verlag.
- Christian Tietjen, Tobias Isenberg, und Bernhard Preim. Illustrative Rendering-Techniken für die medizinische Ausbildung und Therapieplanung. In *Bildverarbeitung für die Medizin*, Informatik aktuell, Seiten 282–286, Berlin, Heidelberg, New York, 2005b. Springer Verlag.
- Robert F. Tobler, Alexander Wilkie, Martin Fedá, und Werner Purgathofer. A Hierarchical Subdivision Algorithm for Stochastic Radiosity Methods. In Julie Dorsey und Philipp Slusallek, Hrsg., *Eurographics Rendering Workshop 1997*, Seiten 193–204, St. Etienne, France, June 1997. Eurographics, Springer Wien.
- Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, und Wolfgang Heidrich. Viewpoint Selection using Viewpoint Entropy. In *VMV '01: Proceedings of the Vision Modeling and Visualization Conference 2001*, Seiten 273–280, Berlin, Amsterdam, 2001. Aka GmbH.

- Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, und Antoni Llobet. Realtime automatic selection of good molecular views. *Computers & Graphics*, 30(1):98–110, 2006.
- Ivan Viola, Miquel Feixas, Mateu Sbert, und Meister Eduard Gröller. Importance-Driven Focus of Attention. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):933–940, October 2006.
- Ivan Viola, Armin Kanitsar, und Meister Eduard Groller. Importance-Driven Volume Rendering. In *VIS '04: Proceedings of the conference on Visualization '04*, Seiten 139–146, Washington, DC, USA, 2004. IEEE Computer Society.
- C. Ware und D. R. Jessome. Using the bat: a six dimensional mouse for object placement. In *Proceedings on Graphics interface '88*, Seiten 119–124, Toronto, Ont., Canada, Canada, 1988. Canadian Information Processing Society.
- Colin Ware und Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. *SIGGRAPH Comput. Graph.*, 24(2):175–183, 1990.
- Josie Wernecke. *The Inventor Mentor: Programming Object-Oriented 3d Graphics with Open Inventor, Release 2*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1993.
- Josie Wernecke. *The Inventor Toolmaker: Extending Open Inventor, Release 2*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994.