
Interner Bericht

Schattenberechnung für die Visualisierung
hybrider Datensätze

Rolf H. van Lengen

Mathias Matzat

Universität Kaiserslautern

258/94

Fachbereich Informatik

Universität Kaiserslautern · Postfach 3049 · D-67653 Kaiserslautern

Schattenberechnung für die Visualisierung hybrider Datensätze

Rolf H. van Lengen

Mathias Matzat

Universität Kaiserslautern

258/94

Universität Kaiserslautern
AG Computergraphik
Postfach 3049
67653 Kaiserslautern

Oktober 1994

Herausgeber: AG Graphische Datenverarbeitung und Computergeometrie
Leiter: Professor Dr. H. Hagen

Schattenberechnung für die Visualisierung hybrider Datensätze

Rolf H. van Lengen und Mathias Matzat
Universität Kaiserslautern
FB Informatik
Postfach 3049
D-67653 Kaiserslautern

Zusammenfassung

Die dreidimensionale Darstellung hybrider Datensätze hat sich in den letzten Jahren als ein wichtiger Teilbereich der wissenschaftlichen Visualisierung etabliert. Hybride Datensätze enthalten sowohl diskrete Volumendaten als auch durch geometrische Primitive definierte Objekte. Bei der visuellen Verarbeitung einer gegebenen Szene spielen Schatteninformationen eine wichtige Rolle, indem sie die Beziehungen von Objekten untereinander verständlich machen. Wir beschreiben ein einfaches Verfahren zur Berechnung von Schatteninformation, das in ein bestehendes System zur Visualisierung hybrider Datensätze integriert wurde. An einem Beispiel aus der klinischen Anwendung werden die Ergebnisse illustriert.

1 Einleitung

Schatteninformationen spielen eine wichtige Rolle bei der visuellen Verarbeitung einer gegebenen Szene. In vielen Fällen wird die räumliche Beziehung von Objekten untereinander erst durch die Lage und Orientierung des Schattenwurfs verständlich (siehe Abb. 1.1). Darüberhinaus verleihen Schatten der graphischen Darstellung einer Szene einen realistischeren Eindruck und erhöhen zugleich deren Tiefenwirkung.

Die Berechnung des Schattens, der von einem Objekt erzeugt wird, setzt sich aus der Bestimmung der Schattenkontur und der Intensität des von der beschatteten Stelle in die Szene reflektierten Lichts zusammen. Beide Operationen erfordern in der Regel einen hohen Rechenaufwand, der durch die Verwendung einer aufwendigen Flächenrepräsentation oder insbesondere durch die Berücksichtigung des von anderen Objekten reflektierten Lichts nochmals erheblich ansteigen kann. Das zuletzt genannte Problem wird u. a. von sogenannten *Globalen Beleuchtungsmodellen*, wie beispielsweise dem *Radiosity* Verfahren, gelöst, auf die aber in diesem Zusammenhang nicht weiter eingegangen werden soll.

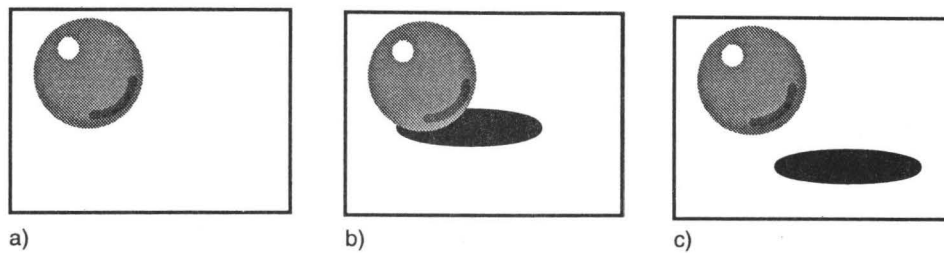


Abbildung 1.1: (a) Die Position der Kugel ist ohne Schatteninformation nicht erkennbar. (b) Die Kugel liegt auf einer Ebene. (c) Die Kugel schwebt über einer Ebene.

Die Mehrheit der Algorithmen generiert in einem Vorverarbeitungsschritt die sichtunabhängige Schatteninformation, die dann während der sichtabhängigen Visualisierung zur eigentlichen Schattenberechnung benötigt wird. Die zur Zeit gebräuchlichsten Verfahren lassen sich nach [1] in fünf verschiedene Klassen einteilen, deren generelle Vorgehensweise jeweils an Hand von Beispielen im folgenden kurz erläutert werden soll.

1.1 Schattenberechnung durch Transformation

Der 1978 von Atherton, Weiler und Greenberg [2] vorgestellte Algorithmus zur Schattenberechnung basiert auf einem von Weiler und Atherton [3] publizierten Hidden-Surface-Verfahren. Zunächst werden die sichtbaren Teile einer mit Polygonen modellierten Szene aus Sicht der Lichtquelle bestimmt. Die sichtbaren Polygone bzw. deren Teilstücke entsprechen dem beleuchteten Teil der Szene und werden dementsprechend gekennzeichnet. Anschließend werden sie in die jeweilige Betrachtersicht zurücktransformiert und bieten zusammen mit den nicht markierten Polygonen eine Möglichkeit, die beschatteten Bereiche der Szene zu bestimmen. Da alle Berechnungen im Objektraum ausgeführt werden, zeichnet sich dieser Algorithmus durch eine äußerst genaue Darstellung der Schattengebiete aus.

1.2 Schattenvolumina

Das von Crow [4] vorgestellte und von Bergeron [5] verbesserte Konzept der Schattenvolumina (sog. *Shadow Volumes*) ist eine relativ alte Technik, die in erster Linie auf polygonal definierten Szenen mit punktförmigen Lichtquellen arbeitet. Dies ist jedoch keine Einschränkung, denn das Verfahren kann im Prinzip auch für kompliziertere Flächenrepräsentationen eingesetzt werden.

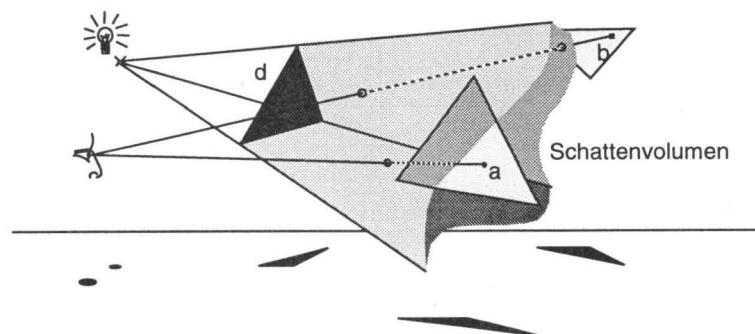


Abbildung 1.2: Das von Dreieck *d* erzeugte Schattenvolumen.

Jedes Polygon der Szene definiert zusammen mit der Lichtquelle ein — eventuell an der Sichtpyramide geclipptes — Schattenvolumen. Ein Punkt p der Szene liegt genau dann im Schatten eines Polygons, falls eine vom Betrachter zum Punkt p verlaufende Gerade über eine ungerade Anzahl von Schnittpunkten mit dem Schattenvolumen verfügt. Der Punkt b beispielsweise (s. Abb. 1.2) wird dagegen von der Lichtquelle beleuchtet, weil die Anzahl der Schnittpunkte mit dem Schattenvolumen gerade ist.

1.3 Scanlinie-orientierte Schattenberechnung

Appel [6] integrierte die Schattenberechnung in seinen scanline-orientierten Algorithmus zur Eliminierung von verdeckten Flächen. In einem Vorverarbeitungsschritt wird eine punktförmige Lichtquelle mit einer geeigneten Kugel umgeben, auf die alle Polygone der Szene der Reihe nach aus Sicht der Lichtquelle projiziert werden. Für jedes Polygon wird eine Liste aller *Schattenpolygone* angelegt, die das Polygon in der Projektion vollständig oder nur teilweise überdecken. Jedes Polygon dieser Liste ist ein möglicher Kandidat für einen Schattenwurf auf das zugehörige Polygon.

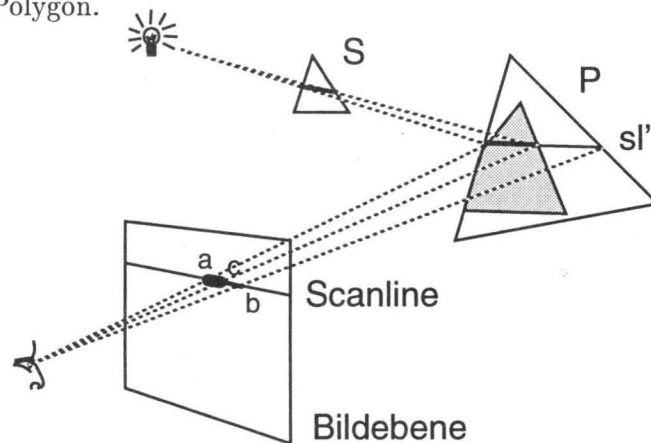


Abbildung 1.3: Nach Bestimmung des sichtbaren Abschnitts \overline{ab} von Polygon P wird getestet, ob es ein Schattenpolygon S gibt, das sl' überdeckt (\overline{ac}).

Bei der Visualisierung wird zunächst das Sichtbarkeitsproblem innerhalb der aktuellen Scanline gelöst. Jeder Teilbereich der Scanline stellt die sichtbare Region eines Polygons oder den Hintergrund der Szene dar. Zur Schattenberechnung wird die sichtbare Region des Polygons auf alle Schattenpolygone der entsprechenden Liste projiziert. Kommt es dabei zu einer Überdeckung, so wirft das Schattenpolygon einen Schatten, und der entsprechende Abschnitt der Scanline wird dunkler dargestellt (s. Abb. 1.3).

Der ausschließliche Gebrauch punktförmiger Lichtquellen und der extrem hohe Speicherbedarf für die Schattenpolygonlisten komplexer Szenen, sowie die rechenintensiven Operationen sind die signifikantesten Nachteile dieses Verfahrens.

1.4 Rekursive Strahlverfolgung

Das Strahlverfolgungsverfahren (*Ray-Tracing*, *Ray-Casting*) bietet eine weitere Möglichkeit der Schattenberechnung. Es lehnt sich eng an die Strahlenoptik der Physik an und simuliert das Verhalten eines Lichtstrahls im dreidimensionalen Raum. In seiner 1964 veröffentlichten

Arbeit beschreibt Appel [6] die Grundlagen des Strahlverfolgungsverfahrens, wie es im Prinzip noch heute in der Praxis zur Anwendung kommt.

Die Position des Betrachters und die Festlegung der rasterförmigen Bildebene bestimmen den Teil der Szene, der abgebildet wird. Für jeden Bildpunkt wird ein Strahl vom Augenpunkt durch die Mitte des Bildpunktes in die dreidimensionale Szene geschickt, um seinen Verlauf anschließend zu verfolgen.

Für diesen Strahl wird der erste Schnittpunkt mit einem Objekt der Szene ermittelt. Verfehlt der Strahl alle Objekte, so erhält der entsprechende Bildpunkt die Hintergrundfarbe. Anderenfalls hängt die Farbgebung von den Eigenschaften des getroffenen Objektes ab. Ist das getroffene Objekt als spiegelnd charakterisiert, so wird gemäß dem Reflexionsgesetz der Physik der reflektierte Strahl rekursiv weiterverfolgt. Ist der Körper außerdem transparent, so kommt zusätzlich das Brechungsgesetz zur Anwendung. Zu diesem Zweck wird der Sehstrahl am Schnittpunkt in einen reflektierten und einen transmittierten Anteil aufgespalten. Anschließend verfolgt man den Verlauf beider Teilstrahlen getrennt weiter. Auf diese Weise kommt es zum Aufbau eines Strahlenbaums, der mit der Rekursionstiefe exponentiell anwächst (s. Abb. 1.4).

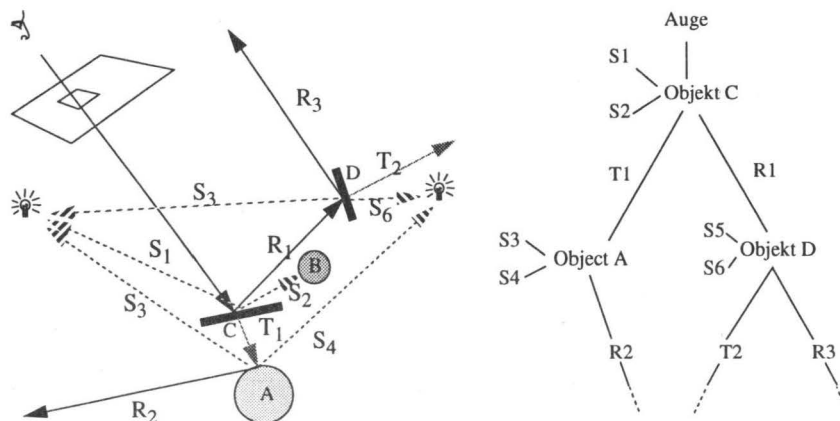


Abbildung 1.4: Prinzip der rekursiven Strahlverfolgung [7].

Zur Bestimmung von Schatteninformation wird zusätzlich für jeden Auftreffpunkt eine Verbindung zu jeder Lichtquelle der Szene gezogen. Diese sogenannten *Schattenfühler* ertasten, ob sich zwischen dem Schnittpunkt und der Lichtquelle ein blockierender Körper befindet. Ist dies der Fall, so liegt der Schnittpunkt bezüglich der Lichtquelle im Schatten und demzufolge wird die Intensität der Lichtquelle bei der Berechnung der Gesamtintensität des Bildpunktes nicht berücksichtigt. Die Lichtintensität wird dabei mit Hilfe eines beliebigen Beleuchtungsmodells für jeden Auftreffpunkt des Strahls getrennt berechnet. Die Gesamtintensität des Bildpunktes ergibt sich abschließend durch aufsummieren der nach dem Dämpfungsgesetz der Physik gewichteten Teilintensitäten.

Das Strahlverfolgungsverfahren liefert photorealistic Bilder von sehr hoher Qualität, deren Berechnung allerdings, trotz der Entwicklung diverser Beschleunigungsansätze [7] [1], sehr zeitintensiv ist.

1.5 Der Schatten-Z-Puffer

Der *Schatten-Z-Puffer* erlaubt die einfache Berechnung von Schatteninformation für beliebig komplexe Szenen. Analog zum bekannten Z-Puffer-Algorithmus, der das Sichtbarkeitspro-

blem löst, wird aus Sicht der Lichtquelle ein Bild mit Tiefeninformation (sog. *Shadow Map*) der Szene generiert, indem die z -Koordinaten der sichtbaren Objekte in die Shadow Map eingetragen werden (s. Abb. 1.5).

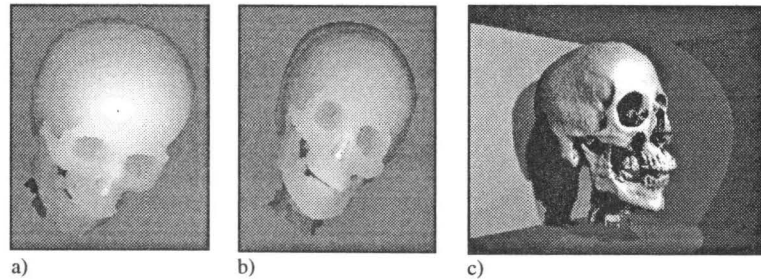


Abbildung 1.5: (a)(b) Die Shadow Maps der beiden Lichtquellen.
(c) Hybrider Datensatz mit Schatten.

Im Verlauf der Visualisierung wird jeder sichtbare Oberflächenpunkt zunächst in das Koordinatensystem der Lichtquelle transformiert. Die Projektion des Punktes in die Shadow Map liefert den Index eines Tiefenwertes, der mit der transformierten z -Koordinate des Punktes verglichen wird. Der Oberflächenpunkt liegt im Schatten, falls der betreffende Tiefeneintrag kleiner als die z -Koordinate ist.

Mit Hilfe des Schatten-Z-Puffers lassen sich beliebig komplexe Szenen in akzeptablen Rechenzeiten verarbeiten. Die verfahrensbedingten Aliasing-Effekte in den Shadow Maps und der hohe Speicherbedarf sind gravierende Nachteile dieser Art der Schattenberechnung, auf die wir in Abschnitt 3.2.2 noch detailliert eingehen werden.

2 Effiziente Visualisierung hybrider Datensätze

Begünstigt durch die fortschreitende Entwicklung auf den Gebieten der Computergraphik und der digitalen Bildverarbeitung finden Algorithmen zur Visualisierung medizinischer Daten eine immer breitere Anwendung. Die in der medizinischen Diagnostik zum Einsatz kommenden modernen bildgebenden Verfahren (z. B. Kernspintomographie (MRT), Computertomographie (CT) und Ultraschall) liefern in der Regel diskrete Meßwerte als Ergebnis einer Untersuchung, die schichtweise in Form einer zweidimensionalen Matrix abgelegt werden. Die räumliche Anordnung dieser Schichten auf einer regelmäßigen Gitterstruktur wird dabei als Volumendatensatz bezeichnet. Die in den einzelnen Gitterpunkten vorliegenden Daten repräsentieren Skalare, Vektoren oder Funktionen mit ihren Parametern (s. Abb. 2.1).

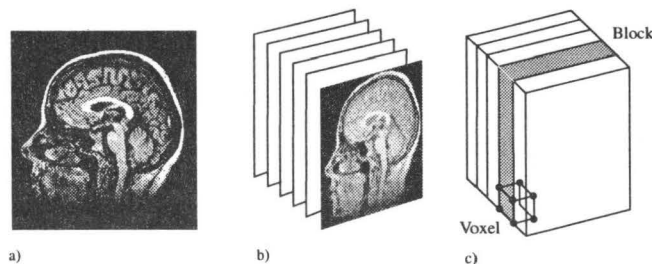


Abbildung 2.1: Kernspintomographie und deren räumliche Anordnung.

Bei der Visualisierung komplexer Volumendatensätze finden herkömmliche Algorithmen aus der Computergraphik keine unmittelbare Anwendung. Spezielle volumenorientierten Visualisierungsalgorithmen wurden erstmals Mitte der achtziger Jahre veröffentlicht (z. B. [8], [9], [10]). Sie basieren im wesentlichen auf dem traditionellen *Ray-Casting*-Verfahren und bieten eine direkte Visualisierung der Volumendaten ohne Generierung geometrischer Zwischenstrukturen.

Analog zum Ray-Tracing werden zunächst vom Projektionszentrum ausgehend durch jeden Bildpunkt Sichtstrahlen berechnet, die anschließend durch den Datensatz verfolgt werden. An diskreten Abtastpunkten entlang des Sichtstrahls wird in Abhängigkeit der gegebenen Daten jeweils ein Farb- und ein Transparenzwert ermittelt. Farbintensität und Transparenz werden miteinander verrechnet und bestimmen zusammen mit den vorher berechneten Werten einen Farbwert, der dem aktuellen Bildpunkt zugewiesen wird. Eine rekursive Strahlverfolgung wie beim Ray-Tracing findet in der Regel jedoch nicht statt.

Im Rahmen der Krebsbehandlung kamen Anfang der neunziger Jahre zum ersten Mal sogenannte *hybride* Visualisierungsalgorithmen zum Einsatz. Das zu bestrahlende Gewebe wurde durch planare Schnittflächen bzw. kegelförmige Elemente geometrisch repräsentiert und zusammen mit den Volumendaten visualisiert. Spezielle hybride Ray-Casting-Verfahren behalten dabei die jeweilige Datenrepräsentationen bei und stellen sie unabhängig voneinander dar. Die verschiedenen Resultate der Strahlanfrage werden in einer nach dem Tiefenwert des Abtastpunktes sortierten Liste gespeichert und anschließend in einem gewöhnlichen Z-Puffer miteinander vermischt. Diese Vorgehensweise garantiert einerseits eine hohe Bildqualität, führt andererseits aber zu erheblich längeren Rechenzeiten.

Das in der Arbeitsgruppe „Graphische Datenverarbeitung und Computergeometrie“ entwickelte Visualisierungssystem zur Darstellung hybrider Datensätze ist für den Einsatz auf kleineren Rechnern und daraus gebildeten heterogenen Netzwerken konzipiert. Im Gegensatz zu den herkömmlichen hybriden Verfahren wird die Visualisierung der verschiedenen Datenrepräsentationen aus Gründen der Rechengeschwindigkeit und der begrenzten Speicherkapazität parallel vorgenommen. Zum besseren Verständnis der Schattenberechnung soll die grundlegende Vorgehensweise bei der Visualisierung hybrider Datensätze im folgenden kurz erläutert werden. Eine vollständige Beschreibung des Algorithmus findet sich in [11], [12] und [13].

2.1 Die Visualisierung hybrider medizinischer Datensätze

Das in der Arbeitsgruppe entwickelte hybride Visualisierungssystem gestattet die Darstellung medizinischer Volumendaten in Verbindung mit geometrisch definierten Daten. Als Geometriedaten werden in diesem Zusammenhang alle Objekte bezeichnet, die durch Polygone (z. B. Dreiecke) oder andere geometrische Primitive definiert werden.

Eine unmittelbare Visualisierung der Volumendaten ist im allgemeinen schwer möglich. Um eine aussagekräftige Visualisierung zu erhalten, ist in einem Vorverarbeitungsschritt die Segmentierung und anschließende Klassifikation des kompletten Datensatzes erforderlich. Segmentierung bezeichnet dabei die Definition zusammenhängender Gebiete innerhalb des Datensatzes, während unter Klassifikation die Zuordnung eines spezifischen Gewebetyps zu einzelnen Gebieten zu verstehen ist, die in Form eines eindeutigen Materiallabels für jedes Datum abgespeichert wird. Dieser Prozeß ist sehr aufwendig, da er vom Anwender interaktiv mit einem speziell dafür entwickelten Editor vorgenommen werden muß. Eine universell einsetzbare automatische Segmentierung aller Gewebetypen mit anschließender Klassifikation gibt es noch nicht. Allerdings erlaubt unserer Visualisierungssystem die Definition zweidimen-

sionaler Filter, wie beispielsweise Schwellwertoperatoren, die eine verhältnismäßig einfach zu realisierende automatische Segmentierung und Klassifikation von Computertomographien zum Zeitpunkt der Visualisierung gestatten.

Nach der Vorverarbeitung startet der eigentliche Visualisierungsprozeß, der zur Zeit über einen Interpretier gesteuert wird. Volumendatensätze aus dem medizinischen Bereich können eine beachtliche Größe erreichen. Da unser Visualisierungssystem für den Einsatz auf kleineren Rechnern konzipiert ist, muß der Speicherplatzbedarf zur Laufzeit möglichst niedrig gehalten werden. Aus diesem Grund werden in Abhängigkeit vom gewählten Visualisierungsverfahren zu jedem Zeitpunkt nur maximal vier Schichten des Datensatzes im Hauptspeicher gehalten. Die beiden inneren Schichten bilden einen sogenannten *Block*, der sich wiederum aus *Voxeln* zusammensetzt (s. Abb. 2.1c).

Zunächst wird durch eine Vorsortierung der Datenschichten die Abarbeitungsfolge der einzelnen Blöcke festgelegt. Sie ergibt sich aus der fortschreitenden Entfernung der Blöcke zu der in der Szene positionierten Kamera. Liegt die Kamera zwischen den Blöcken, so werden zunächst alle Blöcke links der Kamera von rechts nach links abgearbeitet und anschließend die verbleibenden Blöcke von links nach rechts (s. Abb. 2.2a).

Von jedem Block B_i wird die Bounding Box B_i^V der zu visualisierenden Volumendaten berechnet. Bestimmte Eckpunkte von B_i^V werden auf die Bildebene projiziert und definieren dort die Eckpunkte eines geschlossenen Polygons P_i^V . Das Polygon P_i^V ist die konvexe Hülle der Projektion aller in der Bounding Box B_i^V enthaltenen Volumendaten. Die Eckpunkte von P_i^V werden in Abhängigkeit von der Kameraposition mit Hilfe einer Tabelle ermittelt. Analog wird mit den eventuell in Block B_i vorhandenen Geometriedaten verfahren, so daß sich eine weitere Bounding Box B_i^G und somit auch ein Polygon P_i^G ergeben (s. Abb. 2.2b).

Durch einen modifizierten Scanline-Algorithmus werden alle Bildpunkte, die innerhalb der beiden Polygone P_i^V und P_i^G liegen, zeilenweise generiert. Für jeden Bildpunkt werden in Abhängigkeit vom gewählten Antialiasing-Verfahren ein oder mehrere Strahlen in die Szene geschickt und durch die aktuellen Boxen B_i^V und B_i^G verfolgt.

An diskreten Abtastpunkten entlang des Strahls wird jeweils ein Farb- und ein Transparenzwert berechnet. Bei der Berechnung werden sowohl die im Voxel vorhandenen Volumendaten als auch die Geometriedaten berücksichtigt. In Abhängigkeit vom ausgewählten Visualisierungsverfahren können auch die Strahlintervalle zwischen den jeweiligen Abtastpunkten in die Berechnung miteinbezogen werden (s. Abb. 2.3a).

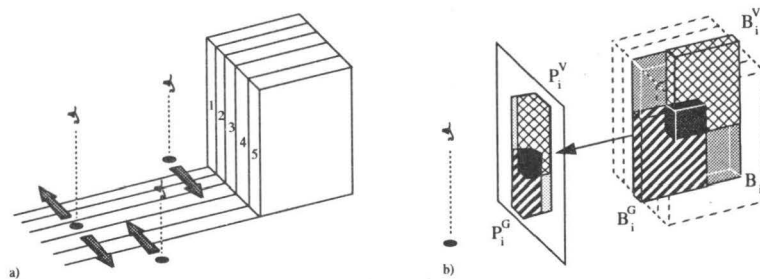


Abbildung 2.2: (a) Abarbeitung der einzelnen Blöcke in Abhängigkeit von der Kameraposition. (b) Projektion der Boxen B_i^V und B_i^G in den Z-Puffer.

Der jeweilige Farb- und Transparenzwert wird anschließend mit dem Z-Puffer-Eintrag des korrespondierenden Bildpunktes verrechnet. Der Z-Puffer entspricht der zweidimensionalen

Bounding Box der Polygone P^V und P^G , wobei P^V und P^G jeweils der konvexen Hülle der Projektion aller zu visualisierenden Volumen- bzw. Geometriedaten entsprechen. Der Z-Puffer wird zur Verrechnung der Teilresultate der einzelnen Blöcke B_i benötigt. Um den hierfür benötigten Speicherplatzbedarf relativ klein zu halten, wurden spezielle Routinen zur Verwaltung des Z-Puffers implementiert. Dabei wurde ausgenutzt, daß im Gegensatz zum konventionellen Z-Puffer einzelne Einträge des Puffers aufgrund der Blocksortierung frühzeitig als fertig markiert werden können (s. Abb. 2.3b). Dies reduziert zum einen den Speicherplatzbedarf und vermindert zum anderen rechenintensive Strahlanfragen.

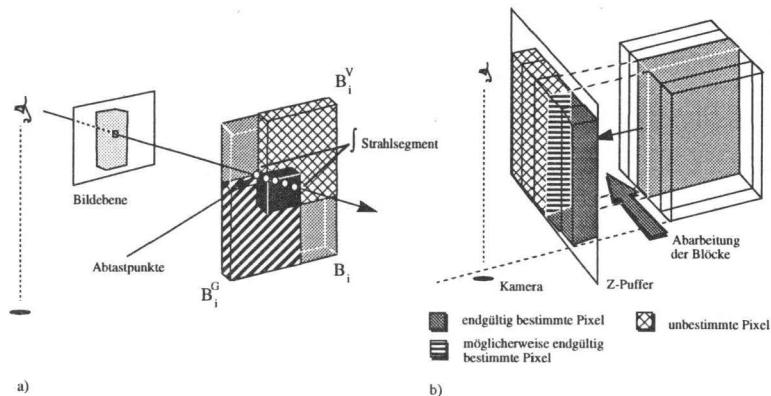


Abbildung 2.3: (a) Strahlverfolgung durch den aktuellen Block B_i . (b) Z-Puffer Belegung.

Eine weitere Beschleunigung läßt sich durch die Verwendung einer Raumhierarchie erreichen. Die hybride Bounding Box H_i der beiden Einzelboxen B_i^V und B_i^G definiert den Objektraum, der durch einen BSP-Baum gleichmäßig unterteilt wird. Der Aufbau der Hierarchie kann dabei auf ein einfaches zweidimensionales Problem zurückgeführt werden [12].

Bei der Strahlverfolgung wird der Strahl von der Kamera aus rekursiv durch den BSP-Baum verfolgt. Die Blätter des BSP-Baums enthalten jeweils zweidimensionale Gitterzellen, die mit Hilfe eines schnellen DDA-Algorithmus der Reihe nach vom Strahl durchlaufen werden. Jede Gitterzelle definiert einen dreidimensionalen Unterraum der hybriden Bounding Box H_i , der aus Voxeln aufgebaut ist und Geometriedaten enthalten kann.

3 Die Berechnung von Schatteninformation

Die in der Einleitung beschriebenen Algorithmen zur Schattenberechnung sind unter Berücksichtigung des vorgestellten Visualisierungskonzeptes nur zum Teil verwendbar. Als wesentliche Nachteile einzelner Verfahren sind in diesem Zusammenhang der einschränkende Gebrauch spezieller Datenstrukturen, die aufwendige Vorverarbeitung der zugrundeliegenden Daten und der gewaltige Speicherplatzbedarf hervorzuheben. Herkömmliche Schattenfühler (s. Abschnitt 1.4), wie sie beim Ray-Tracing Verwendung finden, sind bei dem von uns entwickelten Visualisierungsalgorithmus aufgrund der blockweisen Verarbeitung weniger geeignet und darüberhinaus sehr zeitaufwendig.

Aufgrund der einfachen Integration in das bestehende Konzept und des im Vergleich zu anderen Algorithmen moderaten Speicherplatzbedarfs bietet sich der Schatten-Z-Puffer hier als sinnvoller Kompromiß an. Außerdem ermöglicht dieses Verfahren die sichtunabhängige Generierung von Schatteninformation, was insbesondere bei der Erzeugung von Animationssequenzen von Vorteil sein kann.

3.1 Vorverarbeitung

Wie in der Einleitung schon erwähnt, generiert die Mehrheit der Algorithmen in einem Vorverarbeitungsschritt zunächst die sichtunabhängige Schatteninformation, die dann während der sichtabhängigen Visualisierung zur eigentlichen Schattenberechnung herangezogen wird. Der Schatten-Z-Puffer-Algorithmus generiert zu diesem Zweck Shadow Maps, deren Berechnung in den folgenden Abschnitten beschrieben werden soll.

3.1.1 Der Marching-Cube-Algorithmus

Die Schattenberechnung für hybride Datensätze mit Hilfe des Schatten-Z-Puffers ist nicht direkt möglich. Die Oberflächeninformation einzelner Objekte, die zur Erzeugung der Shadow Maps benötigt wird (s. Abschnitt 1.5), liegt bei Volumendaten nur in impliziter Form vor. Aus diesem Grund muß die diskrete Information in den Gitterpunkten des Datensatzes zunächst in eine einfache geometrische Struktur überführt werden. Zu diesem Zweck verwenden wir den von Lorensen und Cline [14] entwickelten Marching-Cube-Algorithmus, der speziell auf die hier vorliegenden Datensätze angepaßt wurde (s. Abschnitt 3.1.2).

Der Marching-Cube-Algorithmus generiert aus einem diskreten Datensatz durch Vorgabe eines Schwellwerts eine Isofläche, die durch einfache Dreiecksflächen approximiert wird. In Abhängigkeit vom Schwellwert werden zunächst die Eckpunkte jedes Voxels binär klassifiziert. Dabei unterscheidet man zwischen Eckpunkten, die innerhalb bzw. außerhalb der zu erzeugenden Isofläche liegen. Es ergeben sich somit für jedes Voxel 256 mögliche Eckpunkt-kombinationen, die sich allerdings durch Spiegelung, Rotation und Symmetriebetrachtungen auf 15 unterschiedliche Fälle reduzieren lassen. Für jeden Fall läßt sich eine spezielle Dreieckstopologie in einer Tabelle speichern. Sie trennt jeweils die klassifizierten Eckpunkte von den nicht klassifizierten Eckpunkten und gewährleistet dabei die Generierung einer zusammenhängenden Oberfläche. Bei der Konstruktion der Isofläche werden maximal vier Dreiecke pro Voxel erzeugt. Die Dreiecke sind immer so konstruiert, daß sich mindestens eine Dreieckskante in einer Voxelwand befindet und die Eckpunkte jeweils auf einer Voxelkante liegen (s. Abb. 3.1). Die genaue Position der Dreiecksecken wird durch lineare Interpolation des Schwellwerts entlang der Voxelkante ermittelt.

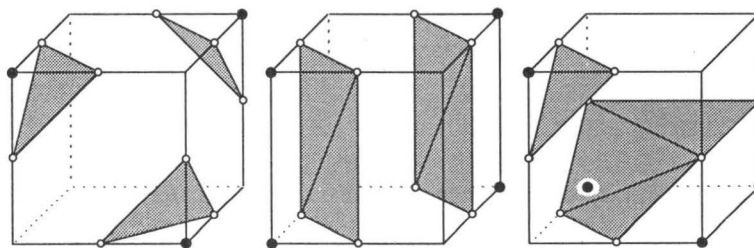


Abbildung 3.1: Drei der 15 möglichen Kombinationen einzelner Voxel Eckpunkte und die daraus resultierende Dreieckstopologie.

Falls die zugrundeliegenden Daten lokal ein annähernd lineares Verhalten zeigen, sind damit gute Ergebnisse bei der Rekonstruktion der zugrundeliegenden Isofläche zu erzielen.

3.1.2 Der adaptive Marching-Cube-Algorithmus

Das allgemeine Marching-Cube-Verfahren, wie es im letzten Abschnitt vorgestellt wurde, ist nicht für die Rekonstruktion von Isoflächen auf der Basis medizinischer Daten geeignet.

In der Hauptsache liegt das in den Eigenschaften der verwendeten Daten begründet. Im allgemeinen ist es unrealistisch, anzunehmen, daß durch die Vorgabe eines Schwellwerts ein bestimmter Gewebetyp vom restlichen Gewebe eindeutig separiert werden kann. Betrachtet man beispielsweise das Grauerthistogramm einer MRT-Aufnahme, so stellt man fest, daß völlig verschiedene Gewebetypen durch identische Meßwerte repräsentiert werden, und damit eine Klassifikation auf Grundlage eines Schwellwerts allein unmöglich ist.

Allerdings steht aus dem der Visualisierung vorgeschalteten Klassifikationsprozeß (s. Abschnitt 2.1) für jeden Meßwert ein Materiallabel zur Verfügung, das bei der Oberflächenrekonstruktion zusätzlich zum Meßwert herangezogen werden kann. Im einfachsten Fall wird die Entscheidung, ob eine Voxelecke innerhalb bzw. außerhalb der Isofläche liegt, nur vom Materiallabel abhängig gemacht. Die Dreieckseckpunkte werden in Ermangelung eines Schwellwerts jeweils auf die Mitte der entsprechenden Voxelkanten gelegt. Nachteil dieser einfachen Vorgehensweise sind unschöne Treppeneffekte, die durch das Auftreten immer gleicher Winkel zwischen den einzelnen Dreiecken entstehen (s. Abb. 3.2a).

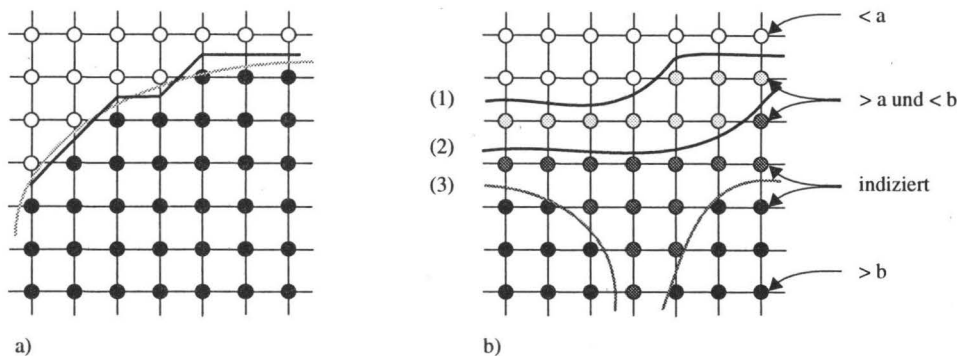


Abbildung 3.2: (a) Berechnung der Isofläche durch Vorgabe des Materiallabels. (b) Berechnung der Isofläche durch Vorgabe des Schwellwerts a (1), des Materiallabels und Schwellwerts a (3) und mit Hilfe des adaptiven Marching-Cube-Algorithmus (2).

Eine Verbesserung läßt sich durch die Einführung eines konstanten Schwellwerts erreichen, der bei der Eckpunktklassifikation zusätzlich berücksichtigt wird. Eine Voxelecke liegt genau dann innerhalb der Isofläche, falls das gewünschte Materiallabel im Eckpunkt vorliegt und der entsprechende Meßwert größer als der Schwellwert ist. Die Position der Dreiecksecken wird analog zum herkömmlichen Marching-Cube-Algorithmus durch lineare Interpolation des Schwellwerts entlang der Voxelkante ermittelt. Die resultierende Isofläche verläuft bei diesem Verfahren nicht unbedingt an der Oberfläche der indizierten Region. Ihr Verlauf wird im wesentlichen durch die benachbarten Datenpunkte bestimmt, deren Meßwerte um den konstanten Schwellwert herum alternieren. Dieses Verhalten kann in extremen Fällen zu Löchern in der Isofläche führen (s. Abb. 3.2b).

Aus diesem Grund sollte die Lage der rekonstruierten Isofläche nur von den Materiallabels abhängig gemacht werden. Durch die Berechnung eines lokalen Schwellwerts, der sich adaptiv an die Umgebung anpaßt, kann zusätzlich eine glatte Oberfläche erzielt werden. In einem Puffer wird für jede Voxelkante, die einen gemeinsamen Schnittpunkt mit der Isofläche besitzt, der berechnete Schwellwert abgelegt. Zur Berechnung eines neuen Schwellwerts wird der gewichtete Durchschnitt der Schwellwerte benachbarter und im Puffer gespeicherter Voxelkanten gebildet. Das Gewicht eines Schwellwerts errechnet sich dabei jeweils aus der Distanz

der entsprechenden Voxelkante zur aktuellen Voxelkante. Dieser Durchschnitt wird in den Puffer eingetragen und interpoliert den Schnittpunkt mit der aktuellen Voxelkante (s. Abb. 3.3). Für den seltenen Fall, daß der Durchschnittsschwellwert nicht innerhalb des Wertebereichs der Voxelkante fällt, wird der Durchschnitt der Kanteneckwerte als neuer Schwellwert angenommen.

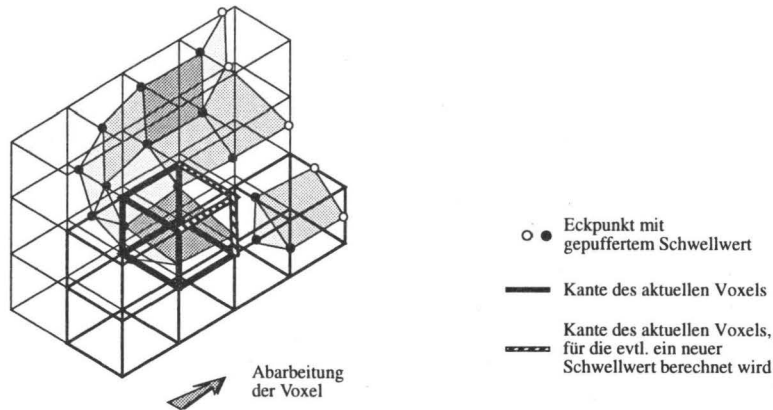


Abbildung 3.3: Zur Berechnung des adaptiven Schwellwerts der aktuellen Voxelkanten werden die gepufferten Schwellwerte der mit • gekennzeichneten Eckpunkte herangezogen.

Das Resultat ist eine Oberfläche, die sich über die gesamte indizierte Region erstreckt und einen relativ glatten Verlauf aufweist (s. Abb. 3.2b).

3.1.3 Berechnung der Shadow Maps

Für jede Lichtquelle der Szene wird vor dem Visualisierungsprozeß eine Shadow Map berechnet. Die Projektion der Bounding Box des Datensatzes in die Shadow Map wird so gewählt, daß die Szene dabei möglichst formatfüllend abgebildet wird. Die Größe der Shadow Map ist variabel und kann dem gewünschten Auflösungsgrad der Szene angepaßt werden. Jeder Eintrag wird mit einem maximal zu erwartenden Tiefenwert vorinitialisiert.

Die Abbildung des Datensatzes in die Shadow Maps erfolgt analog dem späteren Visualisierungsprozeß (s. Abschnitt 2.1). Für jeden Block des Datensatzes wird zunächst die darin enthaltene Geometrie in die Shadow Maps abgebildet. Zur Zeit werden lediglich Dreiecke unterstützt, die jeweils der Reihe nach folgende Operationen durchlaufen:

- Transformation der Eckpunkte in das Koordinatensystem der aktuellen Lichtquelle
- Projektion der Eckpunkte in die Shadow Map der aktuellen Lichtquelle
- Berechnung eines Tiefenwertes für jedes Pixel innerhalb der Projektion. Der Tiefenwert ergibt sich im einfachsten Fall aus dem Schnittpunkt des durch den Mittelpunkt des Pixels verlaufenden Lichtstrahls mit dem Dreieck. Die z -Koordinate des Schnittpunktes wird mit dem entsprechenden Eintrag in der Shadow Map verglichen und ggf. gespeichert.

Im Anschluß daran werden die im aktuellen Block vorhandenen Volumendaten verarbeitet. Für jedes in Betracht kommende Voxel des Blocks werden mit Hilfe des adaptiven Marching-Cube-Algorithmus bis zu vier Dreiecke pro darzustellendem Material erzeugt, die analog zur

Vorgehensweise bei den Geometriedaten der Reihe nach in jede Shadow Map abgebildet werden (s. Abb. 3.4). Die Anzahl der betrachteten Voxel wird durch die Bounding Box B_i^V (vgl. Abschnitt 2.1) der zu visualisierenden Volumendaten erheblich eingeschränkt, was die Berechnung der Shadow Maps beachtlich beschleunigt.

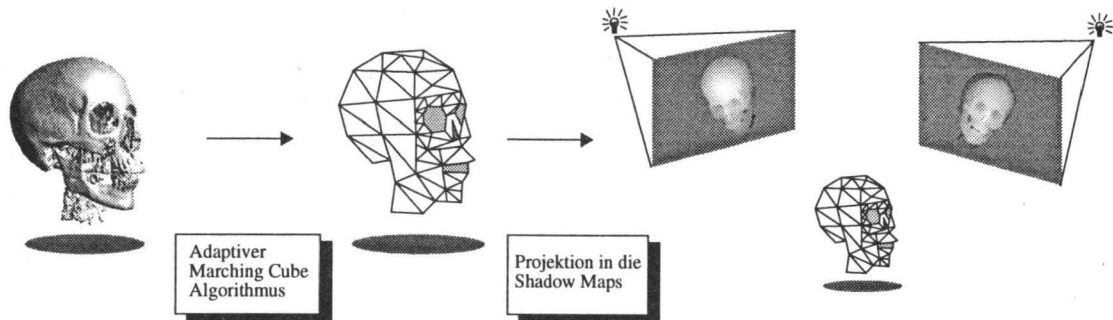


Abbildung 3.4: Prinzip der Shadow-Map-Berechnung für Volumendaten.

Befindet sich die Lichtquelle innerhalb der Szene, so ist es nicht möglich, die komplette Szene in eine Shadow Map abzubilden. In diesem Fall wird um die entsprechende Lichtquelle ein orthogonal zu den Raumachsen des Weltsystems ausgerichteter Würfel aus Shadow Maps (sog. *Shadow Cube*) gelegt. Die Abbildung einzelner Punkte in eine der Shadow Maps kann dann über eine einfache Koordinatenvertauschung erfolgen. Für die Seiten des Shadow Cubes, in die keine Projektion fällt, wird aus Effizienzgründen kein Speicherplatz angelegt.

3.2 Visualisierung

Die Berechnung hybrider Schatteninformation basiert also in der Regel auf einer anderen Datenbasis als die sich ihr anschließende Visualisierung. Diesem Umstand muß bei der Auswahl des Visualisierungsalgorithmus Rechnung getragen werden, damit ein hinreichend genauer Schattenwurf erzielt werden kann. Im folgenden Abschnitt soll ein von uns entwickeltes Modell zur Darstellung von Isoflächen kurz erläutert und die damit erzielten Ergebnisse an Hand eines Beispiels aus der klinischen Diagnostik illustriert werden. Dem schließt sich eine Diskussion der wesentlichen Artefakte, die bei der Verwendung von Shadow Maps auftreten können, an.

3.2.1 Ein Modell zur Visualisierung von Isoflächen

Ziel bei der Visualisierung von Volumendaten ist es, einen Bereich $Q \subseteq \mathbb{R}^3$ des dreidimensionalen Raumes mit den darin enthaltenen Objekten darzustellen. Der Inhalt des Raumbereichs Q läßt sich beschreiben, indem man eine Funktion z auf Q definiert, die für jeden Punkt des Raumbereichs Q angibt, wie sich das Volumen dort zusammensetzt.

Zu diesem Zweck sei $M = \{M_0, \dots, M_{n_m}\}$ die Menge der Materialien, welche in dem Volumenbereich vorkommen. Das Material M_0 wird als *Hintergrundmaterial* ausgezeichnet. M_0 faßt alle Materialien zusammen, die nicht zu Objekten der Volumenszene gehören und wird als *Luft* bezeichnet. Geometriedaten bilden eine separate Klasse von Objekten, die unabhängig davon betrachtet wird.

Dann läßt sich die Funktion $z: Q \rightarrow \mathbb{R}^{n_m+1}$ so definieren, daß die i -te Komponente des Vektors $z(\vec{x})$ angibt, welchen Anteil das Material M_i im Punkt \vec{x} hat. Die i -te Komponente von z sei mit $z_i(\vec{x})$ bezeichnet. Die Funktion z heißt (*Material-*)*Zusammensetzung* auf dem

Raubereich Q . Die Auffistung M der Materialien wird als vollständig angenommen. Daher gilt stets

$$\forall \vec{x} \in Q: \sum_{i=0}^{n_m} z_i(\vec{x}) = 1.$$

Faßt man den Begriff des Materials als Konstruktionselement auf, so liegt in jedem Punkt des Raumes genau ein Material M_i vor. Es gibt also zu jedem Punkt $\vec{x} \in Q$ stets ein i mit $z_i(\vec{x}) = 1$ und $z_j(\vec{x}) = 0$ für alle $j \neq i$. Eine solche Zusammensetzung sei als *diskrete* Zusammensetzung bezeichnet. Bei medizinischen Daten liegt diese Betrachtungsweise vor, wenn man unter Material einen Gewebetyp wie beispielsweise Haut, Wirbel oder Schädelknochen versteht.

Im einfachsten Fall ergibt sich die Zusammensetzung $\tilde{z}_m(\vec{x})$ als Interpolation der durch den Klassifikationsprozeß vorgegebenen Materiallabel $M_{i,j,k}$ in der Umgebung des betrachteten Punktes \vec{x} . Der Gradient der Zusammensetzung, der zur Erzeugung der Isofläche ebenfalls benötigt wird, kann dann einfach als Gradient $\nabla \tilde{z}_m(\vec{x})$ der verwendeten Interpolationsfunktion bestimmt werden.

Ein Nachteil dieser Vorgehensweise ist, daß die Interpolation der Zusammensetzung lediglich auf der vorgegebenen Klassifikation basiert und somit wertvolle Information verschenkt wird. Dies führt zu den gleichen Effekten, wie sie gewöhnlich bei der Visualisierung von binären Volumendaten auftreten. Da für die Umgebung eines Abtastpunktes relativ wenige Konfigurationen möglich sind, scheint das Volumen im berechneten Bild aus quaderförmigen Bausteinen zusammengesetzt zu sein. Aus diesem Grund ist die zusätzliche Berücksichtigung der Meßwerte bei der Interpolation der Materiallabel sinnvoll.

Diese Überlegungen führen zur Berechnung einer verbesserten Zusammensetzungsfunktion \tilde{z}_k . Bei diesem Verfahren wird zunächst eine Interpolation \tilde{g} der Meßwerte bestimmt. Der Gradient von \tilde{g} kann dann direkt als Gradient der Meßwerte verwendet werden. Auf \tilde{g} wird dann eine Klassifikationsfunktion angewendet, um eine Zusammensetzung \tilde{z}_g zu ermitteln.

Die so erzeugte Zusammensetzungsfunktion wird aufgrund der Beschaffenheit der medizinischen Daten in der Regel weit von den tatsächlich vorliegenden Verhältnissen abweichen. Für einen Punkt, der an der Oberfläche eines Knochens liegt, wird die Interpolation der Meßwerte beispielsweise einen Wert liefern, der zwischen dem typischen Wert für Knochen und dem für Muskelgewebe liegt. Dieser Wert ist dann möglicherweise charakteristisch für Fettgewebe, so daß die Funktion \tilde{z}_g zwischen Muskel und Knochen eine Schicht Fettgewebe einschiebt, die im Originalvolumen nicht vorkommt.

Diese Problematik wird gelöst, indem die Zusammensetzung \tilde{z}_m aus der Klassifikation mit der Zusammensetzung \tilde{z}_g aus den Meßwerten zu einer kombinierten Zusammensetzung \tilde{z}_k verrechnet wird. Durch \tilde{z}_m wird dabei die Menge der für die Klassifikation verfügbaren Materialien eingeschränkt, sodaß weniger Mehrdeutigkeiten entstehen. Im obigen Beispiel wird etwa das Material Fettgewebe von vornherein ausgeschlossen, wenn kein Gitterpunkt in der Umgebung des betrachteten Punktes als Fettgewebe klassifiziert ist. Für eine detaillierte Beschreibung der Gewinnung einer kombinierten Zusammensetzung \tilde{z}_k sei auf [15] verwiesen. Der Berechnung von Isoflächen liegt die Vorstellung zugrunde, daß sich das betrachtete Volumen in Regionen zerlegen läßt, die aus einem einheitlichen Material bestehen, daß es also eine diskrete Verteilungsfunktion besitzt. Um aus der interpolierten Materialverteilung $\tilde{z}(\vec{x})$, die ja eine stetige Zusammensetzung ist, eine diskrete Verteilungsfunktion zu machen, wird angenommen, daß in jedem Punkt das Material vorliegt, für welches $\tilde{z}_i(\vec{x})$ den größten Wert besitzt.

Um nun diese Regionen und ihre Grenzflächen darzustellen, wird wieder die Materialzusammensetzung in äquidistanten Abtastpunkten entlang des Sehstrahls ermittelt. Besitzt in zwei aufeinanderfolgenden Abtastpunkten das gleiche Material den höchsten Anteil, so wird für dieses Intervall nur der Volumeneffekt dieses Materials verrechnet. Wechselt dagegen von einem Abtastpunkt zum nächsten das Material mit dem größten Anteil, so muß zwischen diesen beiden Abtastpunkten mindestens eine Grenzfläche zweier Regionen liegen.

Besitze etwa im Abtastpunkt \vec{x}_i das Material M den höchsten Anteil und im Punkt \vec{x}_{i+1} das Material M' . Um die Position des Schnittes mit der Grenzfläche exakt zu ermitteln, wird in der Mitte zwischen den beiden Abtastpunkten ein weiterer Punkt \vec{x}' bestimmt. Liegt in diesem Punkt Material M vor, so muß der Strahl im Intervall zwischen \vec{x}' und \vec{x}_{i+1} eine Grenzfläche passieren, sonst liegt ein Schnittpunkt im Intervall \vec{x}_i, \vec{x}' . Diese Intervallhalbierung wird solange wiederholt, bis der Abstand der Intervallgrenzen unter eine Schranke ε_t gefallen ist. Der auf diese Weise bestimmte Punkt liegt nun auf der Grenzfläche zwischen dem Material M und einem Material M'' . M'' braucht dabei nicht identisch mit M' zu sein, wenn der Strahl im betrachteten Intervall mehrere Oberflächen schneidet. Die Oberfläche zwischen M und M'' ist gegeben durch die Gleichung

$$\tilde{z}_M(\vec{x}) = \tilde{z}_{M''}(\vec{x}).$$

Sie ist also eine Isofläche der Funktion $\tilde{z}_{M''}(\vec{x}) - \tilde{z}_M(\vec{x})$. Somit läßt sich ihre Oberflächennormale als Gradient dieser Funktion gewinnen.

$$\vec{N} = \frac{\nabla \tilde{z}_{M''}(\vec{x}) - \nabla \tilde{z}_M(\vec{x})}{|\nabla \tilde{z}_{M''}(\vec{x}) - \nabla \tilde{z}_M(\vec{x})|}$$

Damit kann der Oberflächeneffekt der Grenzfläche zwischen Material M und M'' mit Hilfe eines oberflächenorientierten Beleuchtungsmodells berechnet werden. Ist nun M'' von M' verschieden, so liegen im Strahlintervall von \vec{x}' bis \vec{x}_{i+1} weitere Grenzflächen, die wie oben beschrieben behandelt werden. Ist $M'' = M'$, so wird davon ausgegangen, daß keine weiteren Grenzflächen vorliegen. Die Berechnung der Volumeneffekte und die anschließende Verrechnung mit den einzelnen Teilintervallen des Strahlabschnitts kann [15] entnommen werden.

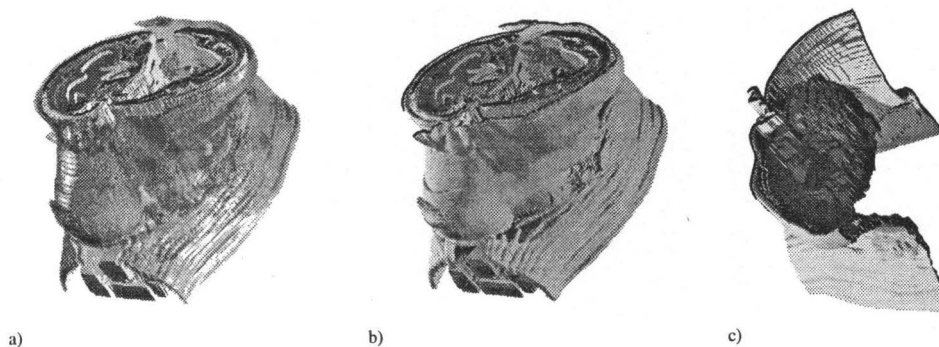


Abbildung 3.5: Hals- und Kinnpartie mit pathologischem Gewebe auf der rechten Halsseite (a) mit transparenter Haut, (b) mit opaker Darstellung der Haut. (c) Detailansicht des pathologischen Gewebes mit Schatten.

Eine schnellere Variante dieses Verfahrens ist ebenfalls verfügbar. Sie kann angewendet werden, wenn nur die Grenzflächen zwischen den Materialien angezeigt werden oder der Volumeneffekt aller Materialien nur einen ambienten, jedoch keinen streuenden Anteil hat. Für

solche Szenen ist der Volumeneffekt nur von der Materialart, jedoch nicht von der Position innerhalb des Materials abhängig. Daher kann bei der Darstellung einer solchen Szene Rechenzeit gespart werden, indem der Volumeneffekt nicht sofort in jedem Abtastpunkt, sondern erst beim Erreichen eines Schnittpunktes, verrechnet wird.

Abbildung 3.5 zeigt die Hals- und Kinnpartie eines Patienten mit pathologischem Befund auf der rechten Halsseite. Bei den zugrundeliegenden Daten handelt es sich um Schichtaufnahmen eines Computertomographen, die mit 4 mm Vorschub abgetastet wurden. Die in Abb. 3.5c erkennbaren Artefakte sind nicht auf das Visualisierungsverfahren zurückzuführen (vgl. Abb. 3.10c), sondern auf die in der klinischen Diagnostik übliche niedrige Abtastrate. Dieses Beispiel veranschaulicht, daß es notwendig ist, geeignete Algorithmen zur Interpolation künstlicher Zwischenschichten einzuführen.

3.2.2 Diskussion einiger Artefakte

Das vorgestellte Konzept zur Generierung hybrider Schatteninformation ist nicht frei von Problemen, die zu sichtbaren Artefakten im berechneten Bild führen können. Die wichtigsten Artefakte und deren Beseitigung sind Gegenstand der folgenden Abschnitte.

3.2.2.1 Aliasing Die Verwendung unterschiedlicher Abtastraten bei der Berechnung der Shadow Maps bzw. des Bildes verursachen insbesondere im Randbereich der generierten Schatten starke Aliasingeffekte, die sich u. a. als störende Treppenstufen im Bild bemerkbar machen (s. Abb. 3.6). Dies liegt vor allem daran, daß verschiedene Sehstrahlen beim Schattentest einen identischen Eintrag in der Shadow Map abfragen bzw. mehrere Einträge der Shadow Map auf ein Pixel der Bildebene projiziert werden.

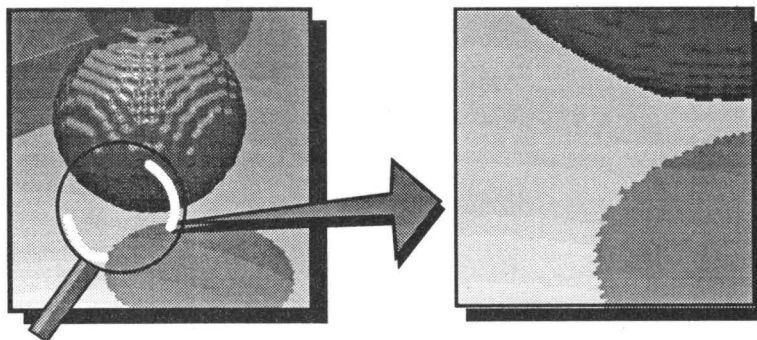


Abbildung 3.6: Artefakte aufgrund unterschiedlicher Abtastraten bei der Berechnung der Shadow Map.

Eine Lösung des in Abb. 3.6 gezeigten Problems bietet das von Reeves, Salesin, Cook [16] entwickelte, als *Percentage Closer Filtering* (PCF) bezeichnete Verfahren. Die Entscheidung, ob ein Punkt im Schatten liegt, wird hier nicht nur von einem Element der Shadow Map abhängig gemacht, sondern zusätzlich von den Einträgen in der unmittelbaren Nachbarschaft. Die Autoren konnten zeigen, daß eine einfache Mittelwertbildung, wie sie beispielsweise von Glättungsfiltren verwendet wird, keine Verbesserung bringt. Beim PCF wird deshalb für jeden Eintrag der Nachbarschaft explizit geprüft, ob der gespeicherte Wert größer oder kleiner als der Tiefenwert des gerade untersuchten Punktes ist. Das Verhältnis der Einträge mit einem kleineren Tiefenwert zur Gesamtheit der betrachteten Einträge ergibt dann den prozentualen Anteil des Schattens, der auf den aktuellen Punkt fällt (s. Abb. 3.7).

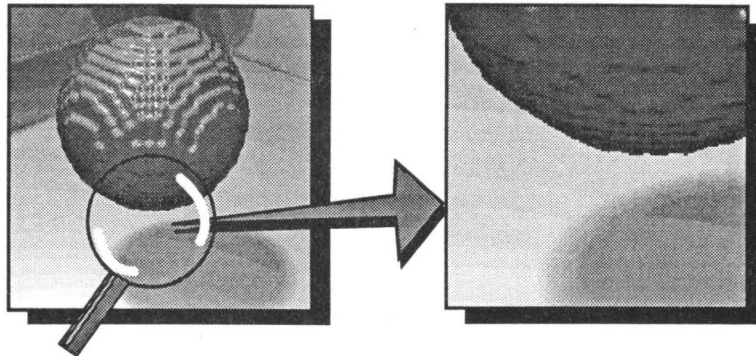


Abbildung 3.7: Mit PCF-Filter geglätteter Schatten einer Kugel.
Die Größe des Filters beträgt 11 x 11 Einträge.

Die Anzahl der zur Schattenberechnung berücksichtigten Nachbarn kann frei vom Anwender definiert werden. Die Ausdehnung des Filters bestimmt die Weichheit der Schattenübergänge.

3.2.2.2 Selbstbeschattung Die fehlerhafte Berechnung eines Schattens, den ein Objekt auf sich selbst wirft, wird als Selbstbeschattung bezeichnet. Sie tritt immer dann auf, wenn verhältnismäßig große Objekte unter einem flachen Winkel in die Shadow Map projiziert werden. In Abhängigkeit vom Winkel werden dann Objektbereiche mit erheblicher Tiefenänderung nur durch ungenügend viele Einträge in der Shadow Map repräsentiert. Dies hat zur Folge, daß für Teile des Objekts ein unerwünschter Schattenwurf berechnet wird, obwohl es unmittelbar vor der Lichtquelle liegt (s. Abb. 3.8). Unter Umständen kann dieser Fehler sogar zu störenden Moiré-Mustern im Bild führen.

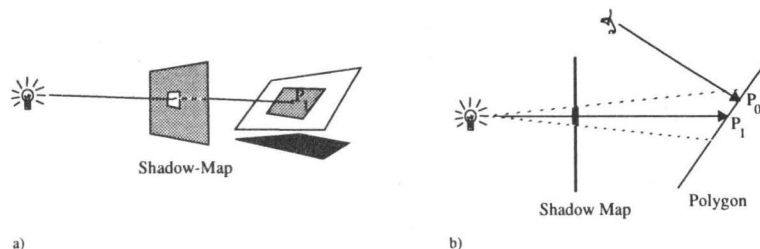


Abbildung 3.8: (a) Der Mittelpunktstrahl definiert den Tiefenwert des Shadow-Map-Eintrags. (b) Punkt P_0 liegt im Schatten, obwohl er unmittelbar vor der Lichtquelle liegt.

Beim Ray-Tracing wird die Selbstbeschattung in der Regel dadurch vermieden, indem zusätzlich zum Tiefenwert eine Dreiecks-*ID-Nummer* gespeichert wird. Die ID-Nummer ordnet dem Tiefenwert der Shadow Map eindeutig ein Dreieck der Szene zu, das diesem Wert entspricht. Beim Schattentest wird dann geprüft, ob die ID des getroffenen Dreiecks mit der ID des Eintrags in der Shadow Map übereinstimmt. In diesem Fall fällt kein Schatten auf das Dreieck. Diese Vorgehensweise bietet sich in unserem System nicht an, denn während der Visualisierung der Volumendaten werden in der Regel keine Dreiecke erzeugt, auf die später beim Schattentest Bezug genommen werden kann.

Eine andere Möglichkeit ist die Definition eines konstanten *Bias*, um den der Tiefenwert des Testpunktes vor dem Vergleich mit dem Eintrag der Shadow Map angehoben wird. Allerdings

ist die Wahl eines geeigneten Bias sehr schwierig oder gar unmöglich, und so ergeben sich zwangsläufig sichtbare Fehler im Bild (s. Abb. 3.10b).

Eine sehr effiziente Methode, die angesprochenen Effekte zu unterdrücken, ist das unter anderem beim Ray-Tracing eingesetzte *Subsampling*. Zur Bestimmung eines Tiefenwertes der Shadow Map wird durch jede Ecke des Eintrags ein Strahl verfolgt und mit der Ebene, in der das aktuelle Polygon liegt, geschnitten. Als Tiefenwert wird jeweils das Maximum der vier Werte gespeichert (s. Abb. 3.9).

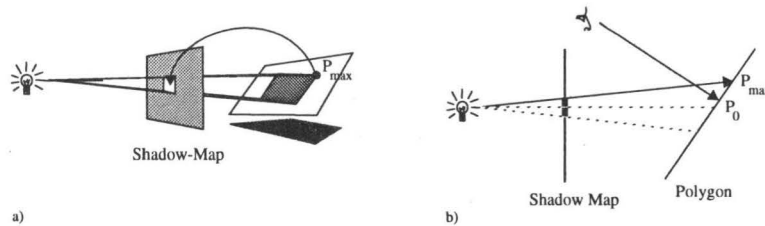


Abbildung 3.9: (a) Der Eckpunktstrahl mit maximalem Tiefenwert P_{max} definiert den Eintrag der Shadow Map. (b) Punkt P_0 wird korrekt beleuchtet.

Zur Beschleunigung des Verfahrens kann ausgenutzt werden, daß das Maximum der Tiefenwerte eines Polygons immer in der gleichen Ecke des Eintrags angenommen wird, so daß für alle weiteren Einträge, die das Polygon in der Shadow Map überdeckt, jeweils nur ein Strahl erzeugt werden muß. Mit diesem verhältnismäßig einfachen Verfahren kann eine Selbstbeschattung, wie in Abbildung 3.10c am Beispiel der Visualisierung eines Jochbeins, sicher ausgeschlossen werden.

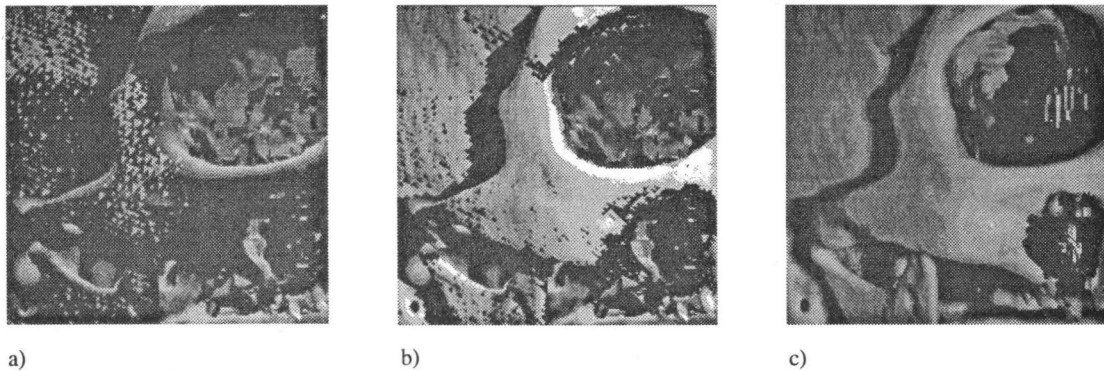


Abbildung 3.10: Bildartefakte durch Selbstbeschattung. (a) kein Antialiasing (b) Bias = 1.00 mm (c) Subsampling

4 Zusammenfassung

Wir haben im Rahmen dieses Berichts ein einfaches Verfahren zur approximativen Berechnung von Schatteninformation und dessen Integration in ein System zur Visualisierung hybrider Datensätze vorgestellt. Es arbeitet auf beliebig komplexen Szenen und bietet einen schnellen Schattentest. Bei der Generierung von Animationssequenzen entfällt eine Neuberechnung der Shadow Maps, solange die Position der Lichtquellen bzw. der Objekte nicht verändert wird.

Die Berechnung der Shadow Maps kann analog der Visualisierung in einem verteilten System erfolgen, indem jeder Server die Shadow Maps für n Blöcke der Szene berechnet und sie an den Client schickt. Der Client setzt die Shadow Maps zusammen und verschickt sie vor der eigentlichen Visualisierung an alle Server [13].

Von Nachteil ist der benötigte Speicherplatz für die einzelnen Shadow Maps, der in Abhängigkeit von der Anzahl der Lichtquellen und deren Position im Raum erheblich ansteigen kann. Zur Zeit ist die Berechnung des Schattenwurfs eines transparenten Objekts nicht möglich. Die Verarbeitung transparenter Objekte ist ein komplexes Problem, denn die Absorption und Dispersion von Licht müßten simuliert werden. Die Dispersion, d.h. die Lichtbrechung an der Trennschicht zweier Medien ist in unserem System durch die blockweise Verarbeitung der Daten nicht möglich. Die Absorption von Lichtenergie soll in Zukunft durch die Berechnung eines Absorptionsprofils für jeden Eintrag der Shadow Map simuliert werden. Durch Kompressionsmethoden, wie beispielsweise der Vektorquantisierung, soll der Speicherplatzbedarf relativ niedrig gehalten werden.

An dieser Stelle möchten die Autoren Herrn Priv.-Doz. Dr. E.-L. Schmidt vom Klinikum Kaiserslautern danken, der uns die in Abschnitt 3.2.1 visualisierten Computertomographie-daten freundlicherweise zur Verfügung gestellt hat. Ein besonderer Dank gebührt auch Herrn Dipl. Ing. Peter Decker, der die mühevollen Segmentierung der Rohdaten vorgenommen hat.

Literatur

- [1] Alan Watt and Mark Watt. *Advanced Animation and Rendering Techniques*. Addison-Wesley, New York, July 1992.
- [2] P. R. Atherton, K. Weiler, and D. Greenberg. Polygon Shading Generation. In *Computer Graphics*, volume 3 of 12, pages 275–281. SIGGRAPH, August 1978.
- [3] K. Weiler and P. G. Atherton. Hidden Surface Removal Using Polygon Area Sorting. In *SIGGRAPH '77*, pages 214–222, 1977.
- [4] F. C. Crow. Shadow Algorithms for Computer Graphics. In *Computer Graphics*, volume 3 of 11, pages 242–248. SIGGRAPH, July 1977.
- [5] P. Bergeron. A General Version of Crow's Shadow Volumes. *IEEE Computer Graphics & Applications*, 6(9):17–28, September 1986.
- [6] A. Appel. Some techniques for shading machine renderings of solids. In *AFIPS 1968 Spring Joint Computer Conference*, volume 32, pages 37–45, 1968.
- [7] A. S. Glassner. *An Introduction to Ray Tracing*. Academic Press, London, 1989.
- [8] M. Levoy. Display of Surfaces from Volume Data. *IEEE Computer Graphics & Applications*, pages 29–37, May 1988.

- [9] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume Rendering. *Computer Graphics*, 22(4):65–74, August 1988.
- [10] P. Sabella. A Rendering Algorithm for Visualizing 3D Scalar Fields. *Computer Graphics*, 22(4):51–58, August 1988.
- [11] R. H. van Lengen. The Volume Priority Z-Buffer. In H. Hagen, H. Müller, and G.M. Nielson, editors, *Focus on Scientific Visualization*, pages 293–304. Springer, New York, August 1992.
- [12] R. H. van Lengen and J. Meyer. Efficient 3-D Visualization of Hybrid Medical Data Sets. Technical Report 257/94, Department of Computer Science, University of Kaiserslautern, Oktober 1994.
- [13] Dietrich Bongartz. Verteilte Visualisierung hybrider medizinischer Datensätze. Diplomarbeit, Universität Kaiserslautern, FB Informatik, Postfach 3049, D-67653 Kaiserslautern, Dezember 1994.
- [14] William E. Lorensen and Harvey E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, 21(4):163–169, July 1987.
- [15] Roger Röhrig. Theoretische und empirische Ansätze zur Visualisierung medizinischer Datensätze. Diplomarbeit, Universität Kaiserslautern, FB Informatik, Postfach 3049, D-67653 Kaiserslautern, Juni 1994.
- [16] William T. Reeves, David H. Salesin, and Robert L. Cook. Rendering Antialiased Shadows with Depth Maps. *Computer Graphics*, 21(4):283–291, July 1987.
- [17] E. A. Haines and D. P. Greenberg. The Light Buffer: A Shadow-Testing Accelerator. *Computer Graphics & Applications*, 6(9):6–16, September 1986.
- [18] Mathias Matzat. Marching Cube Algorithmus zur Oberflächenkonstruktion medizinischer Daten. Projektarbeit, März 1994.
- [19] L. Williams. Casting Curved Shadows on Curved Surfaces. *Computer Graphics*, 12(3):270–274, August 1978.