Octree-Generierung anhand rotierender Objekte

Seminararbeit

vorgelegt am

Lehrstuhl für Praktische Informatik IV
Prof. Dr. Wolfgang Effelsberg
Universität Mannheim

im
November 2002

von
Cornelius Siller

Inhaltsverzeichnis

Inhaltsverzeichnis *2*

1 Einleitung *3*

2 Voxels *3*

3 Prinzipielle Vorgehensweise *4*

4 Bildanalyse *5*

4.1 Aufteilung in Hintergrund / Vordergrund **5**

4.2 Projektion 3D nach 2D **6**

4.3 Projektion der Voxel **8**

5 Kontinuierliche Rotation *9*

5.1 Rotation **9**

5.2 Octree Generation **9**

5.3 Vorteil der Octree-Methode **11**

6 Kameraparameter und Bildanalyse *12*

6.1 Klassifikation in Bildelemente **12**

6.2 Ermittlung der Ausrichtung **12**

6.3 Kameraparameter **14**

7 Beschränkungen und Optimierungen des Verfahrens *15*

7.1 Distance Vector Maps **15**

7.2 Visuelle Hülle **17**

7.3 Erfassung der Polbereiche **18**

8 Zusammenfassung *21*

Literaturverzeichnis *22*

1 Einleitung

Gegenstand dieser Seminararbeit ist die Beschreibung des 1990 von Richard Szeliski in [1] beschriebenen Verfahrens zur Rekonstruktion von 3-dimensionalen Objektdaten aus 2-dimensionalen Kameraaufnahmen in Form einer Rundumsequenz[1].
Das zu analysierende Objekt wird auf einen Objektteller gestellt, welcher sich mit kleinen Winkelschritten dreht. Eine stationäre Kamera nimmt ein Bild von diesem Objekt pro Drehschritt auf und übermittelt diese Bilddaten an einen angeschlossenen Rechner. Im Rechner werden die Bilddaten in Echtzeit analysiert und verwendet, um ein internes Modell des Objektes anhand der neu gewonnenen Informationen zu aktualisieren. Mit jedem Winkelschritt steigt die Wirklichkeitstreue des Modells. Ist eine vollständige Umdrehung abgeschlossen, wird die Genauigkeit erhöht und die Bildanalyse für die nächste Umdrehung mit dieser erhöhten Genauigkeit erneut vorgenommen.

2 Voxels

3-dimensionale virtuelle Welten kann man zur Vereinfachung mit Hilfe der Voxeltechnik beschreiben. Dazu wird eine 3-dimensionale begrenzte Szene in viele Würfel von gleichen Ausmaßen unterteilt. Entsprechend der geforderten Genauigkeit der Szenenbeschreibung erfolgt die Wahl der Würfelanzahl. Eine hohe gewünschte Genauigkeit erfordert eine hohe Würfelzahl. Im Prinzip sind Voxel das 3-dimensionale Analogum zu Pixeln einer 2-dimensionalen Fläche.
Jeder Voxel besitzt einen bestimmten Zustand. Dieser wird beim Erstellen der Szene ermittelt und kann später durch Werkzeuge zur Szenenmanipulation verändert werden. In der Regel wird als Zustand gespeichert, ob dieser Voxel leer oder ausgefüllt ist und welche Farbe die Füllmasse besitzt. Eventuell kann man, wenn bekannt, noch weitere Lichtcharakteristika des im Voxel befindlichen Materials speichern, z.B. ob stark oder schwach reflektierend, matt oder glänzend, klar oder diffus bei lichtdurchlässigem Material.
Dies ermöglicht eine photorealistische Darstellung, wenn die Szene später gerendert werden soll, nachdem Veränderungen vorgenommen wurden, z.B. wenn man eine Punktlichtquelle hinzugefügt hat oder die Helligkeit des Umgebungslichtes verändert hat.

---

[1] Die in dieser Arbeit dargestellten Grafiken sind zum Teil aus [1] entnommen.

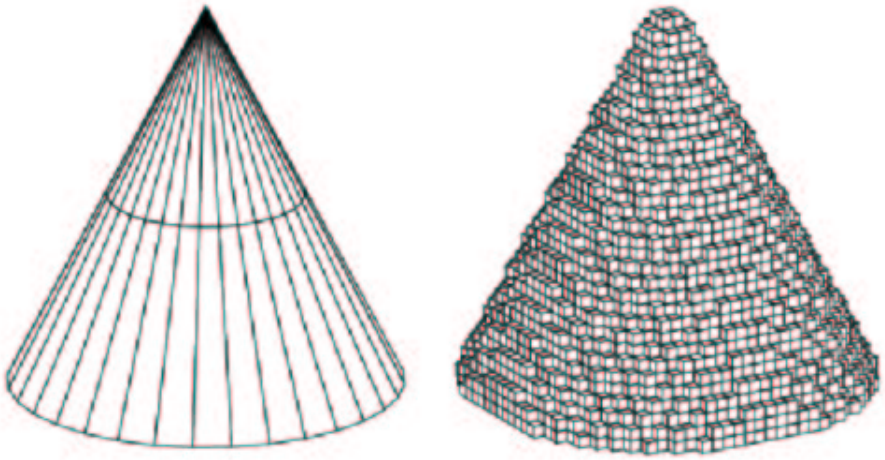

Abbildung 1: links ein gleichmäßiger Kegel, rechts seine Voxeldarstellung

Soll eine vorgegebene Szene mittels der Voxeltechnik beschrieben werden, müssen zunächst die zu erfassenden Ausmaße festgelegt werden. Es empfiehlt sich, wie im oben dargestellten Beispiel (Auflösung = 32x32x32 Voxel) kubische Ausmaße zu wählen. Durch Vorgabe der Voxelgenauigkeit kann man dann die Zahl der gesamten Voxel und des erforderlichen Speichers berechnen:

$$erforderl.\ Speicher = \frac{Kantenlänge\ der\ Szene^3\ Speicher\ pro\ Voxel}{Kantenlänge\ eines\ Voxels^3}$$

Der erforderliche Speicher nimmt also mit der dritten Potenz zu, wenn die Genauigkeit zunimmt. Halbiert man die Kantenlänge eines Voxels, steigt die erforderliche Speichermenge auf das achtfache an.
Um hier Speicherplatz zu sparen, kann man die Voxeldaten hierarchisch ordnen und in einem Baum organisiert abspeichern. Dieses Verfahren wird weiter unten beschrieben und angewandt werden.

3 Prinzipielle Vorgehensweise

Man konstruiert im Rechner ein leeres kubisches Voxelfeld, dessen Größe so gewählt wird, daß das zu analysierende Objekt mit Sicherheit hinein paßt.
Das erste Kamerabild wird analysiert und .die zum Objekt gehörende Bildfläche entsprechend markiert.

Das Voxelfeld projiziert man nun in das Kamerabild und legt es über das Objekt. Für jeden projizierten Voxel erhält man einen Projektionsrahmen. Gehören alle Pixel innerhalb dieses Projektionsrahmens zum Objekt, wird im Voxelmodell dieses Voxel als zum Objekt gehörend markiert.

4 Bildanalyse

4.1 Aufteilung in Hintergrund / Vordergrund

Das beschriebene Verfahren unterscheidet zwischen dem zu analysierenden Objekt und dem Hintergrund anhand der Farbunterschiede. Daher ist es wichtig, daß man beim Modellaufbau eine Hintergrundfarbe wählt, die nicht am Objekt selber auftaucht. Wie der Hintergrund, so muß auch der Objektteller eine andere Farbe als das Objekt aufweisen.
Wenn diese Maßgaben eingehalten werden, so kann der Algorithmus bei der Bildanalyse einfach anhand eines Farbvergleiches Hintergrund und Objektteller ausschließen. Der übrigbleibende Bildteil gehört zum Objekt.
Da bei Kameraaufnahmen die Farben in der Regel nicht exakt wiedergegeben werden, sondern durch Rauschen beeinflußt werden, empfiehlt sich, die vordefinierten Farben von Hintergrund und Objektteller nicht einfach mit den durch die Kamera aufgezeichneten Farben zu vergleichen, sondern auch ähnliche Farben zu akzeptieren. Dazu legt man für die Hintergrund – und Vordergrundfarbe ein Farbspektrum fest. Liegt die Farbe eines Pixels des Kamerabildes innerhalb eines der definierten Spektren, wird das Pixel dementsprechend als Hintergrund bzw. Vordergrund klassifiziert. Um eine Unterscheidung zwischen Hintergrund, Objekt und Objektteller zu gewährleisten, sind die Farben beim Modellaufbau daher mit hinreichenden Abständen zu wählen.

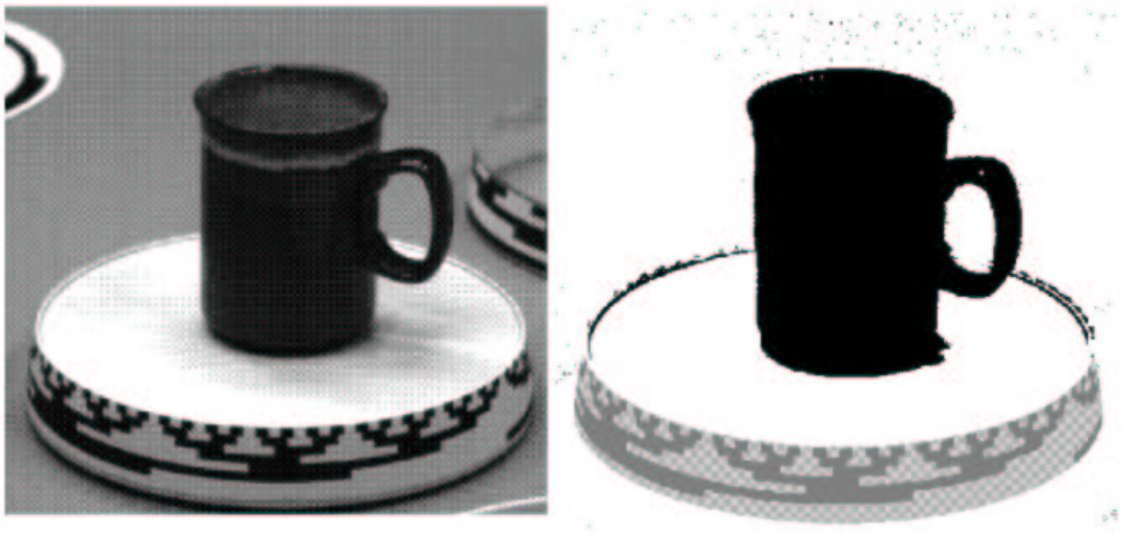

Abbildung 2: Input und Output der Bildanalysefunktion

Die zwei oberen Grafiken zeigen das Beispiel eines Modellaufbaus. Die erste zeigt das von der Kamera übermittelte Bild. Bei der zweiten wurde die Bildanalyse durchgeführt und die Einteilung in Objektfläche, Hintergrund und Markierungscode vorgenommen.

4.2 Projektion 3D nach 2D

Bei der Darstellung von 3-dimensionalen virtuellen Objekten ergibt sich das Problem, wie diese auf einer 2-dimensionalen Anzeigenfläche dargestellt werden sollen.
Ein einfaches, jedoch nicht exaktes Verfahren ist die orthographische Projektion. Hierbei erfolgt die Projektion eines 3-dimensionalen Punktes mit den Koordinaten (x,y,z) durch einfaches Weglassen der z-Koordinate auf den 2-D Punkt (x,y).

Das optisch korrekte Verfahren wird in der folgenden Grafik veranschaulicht:

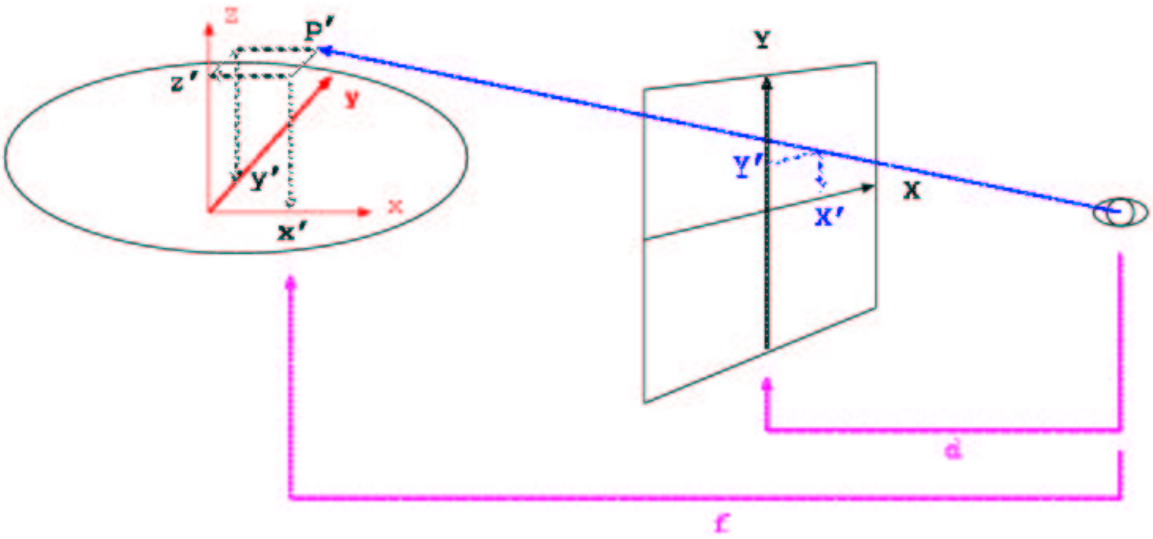

Abbildung 3: Projektion 3D nach 2D

Die Projektionsfläche befindet sich im Abstand *d* vom Betrachter. Der Abstand der zur Projektionsfläche parallelen durch den Punkt *p* gehenden Ebene vom Betrachter ist hier *f*. Der Punkt p im Raum wird nach der folgenden Formel auf den Punkt (X‘, Y‘) der Projektionsebene projiziert:

$$\frac{Y'}{y'} = \frac{d}{f}$$

Sinnvoll ist es, das Koordinatensystem so festzulegen, daß eine Achse senkrecht durch die Projektionsfläche verläuft. Konkret bedeutet das für das beschriebene Verfahren, daß eine Koordinatenachse (im folgenden wird die x-Achse gewählt) durch das Drehzentrum und den Ort der Kamera verlaufen sollte. Auf diese Weise kann man *f* einfach ermitteln als $|x\_kamera - x\_p|$.

4.3 Projektion der Voxel

Liegt das Kamerabild eines Objektes vor und möchte man feststellen, ob ein bestimmter Würfel vollständig von diesem Objekt umschlossen wird oder nicht, projiziert man ihn auf die oben beschriebene Weise in die Bildfläche. Es ergibt sich ein Sechseck. Nun überprüft man, ob die innerhalb dieses Sechsecks liegenden Pixel Vordergrund- oder Hintergrundpixel sind. Sind alle Pixel Hintergrundpixel, liegt das Sechseck außerhalb, sonst innerhalb oder teilweise innerhalb der Objektsilhouette.

Liegt es vollständig außerhalb der Objektfläche, steht definitiv fest, daß auch der betrachtete Voxel vollständig außerhalb des Objektes ist. Liegt das Sechseck nur teilweise außerhalb der Objektfläche, läßt sich keine nähere Aussage treffen. Auch wenn das Sechseck vollständig innerhalb der Objektfläche liegt, läßt sich nichts endgültiges sagen, da man die Ausdehnung des Objektes entlang der Blickrichtung der Kamera nicht kennt

Man kann also nicht sagen, ob der Würfel vor, in oder hinter dem Objekt liegt. Man kann lediglich die Voxel ausschließen, die außerhalb der Silhouettenpyramide liegen, die ihre Spitze im Ort des Betrachters (= Ort der Kamera) hat und deren Seiten so gelegt sind, daß sie genau an den Rändern der Objektsilhouette anliegen..

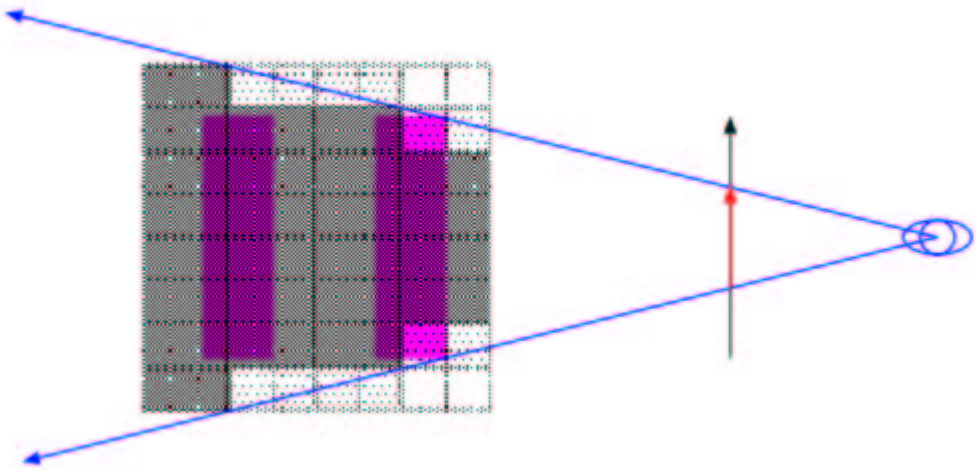

Abbildung 4: die Silhouettenpyramide

Die obige Abbildung zeigt eine Ansicht des Modellaufbaus von oben. Rechts ist der Betrachter, der in Richtung des links gelegenen lila markierten Objektes blickt. Das Objekt hinterläßt auf der in der Mitte gelegenen Projektionsebene die rot markierte Silhouette. Die blauen, an den Rändern der Silhouette anliegenden Strahlen repräsentieren die Silhouettenpyramide. Man sieht, daß theoretisch alle schwarz markierten Voxel zum Objekt gehören könnten.

5 Kontinuierliche Rotation

Nun wird eine kontinuierliche Sequenz von Kameraaufnahmen des sich drehenden Objektes betrachtet. Im Rechner wird ein Objektmodell erstellt, welches mit jedem Bild verglichen und anhand neuer Bildinformationen aktualisiert wird. Das in [1] beschriebene Verfahren sieht vor, daß die Rotation in kleinen festgelegten Schritten zu erfolgen hat. Die Festlegung der Winkelgröße pro Rotationsschritt ist nötig, damit das intern gespeicherte Modell des Objektes ebenfalls um den entsprechenden Winkel gedreht wird und damit korrekt in die Bildfläche projiziert werden kann.

5.1 Rotation

Um ein 3-dimensionales Objekt um ein bestimmtes Drehzentrum um einen bestimmten Winkel zu drehen, dreht man jeden einzelnen Oberflächenpunkt *(x, y, z)* des Objektes. Besteht das Objekt aus Voxeln, dreht man die Eckpunkte jedes einzelnen Voxels. Für die im Verfahren benötigte Drehung um die z-Achse geschieht dies mit der folgenden Formel:

$$x' = x\cos(\alpha) - y\sin(\alpha)$$
$$y' = x\sin(\alpha) + y\cos(\alpha)$$
$$z' = z$$

5.2 Octree Generation

Das Octree-Verfahren beginnt damit, daß ein virtueller Würfel generiert wird, von dem man sich sicher ist, daß das zu vermessende Objekt vollständig darin Platz findet. Diesen Hauptwürfel unterteilt man in acht gleich große Subwürfel.

Für das erste aufgenommene Bild zentriert man den Hauptwürfel über der Drehachse des Objekttellers. Man projiziert jeden Subwürfel in die Bildfläche, konstruiert seine quadratische Hülle und schaut in der distance-vector map nach, ob diese Hülle komplett außerhalb von der Objektfläche, teilweise oder vollständig in ihr liegt.

Jedem dieser drei Zustände wird zur Vereinfachung eine Farbe zugeordnet:

- vollständig außerhalb: weiß
- teilweise außerhalb: grau
- vollständig innerhalb: schwarz

In einem leeren Baum wird an den Wurzelknoten für jeden Subwürfel ein Sohnknoten hinzugefügt. Jeder Knoten wird nach dem Projektionstest mit der Farbe des Ergebnisses markiert.

Beim nächsten Bild dreht man zuerst den Hauptwürfel und seine Subwürfel um die angegebene Gradzahl. Danach führt man für jeden Subwürfel wieder den Projektionstest durch und gleicht die Ergebnisse mit den in den Knoten gespeicherten Farben ab. Ist ein Knoten als weiß markiert, bleibt er auf jeden Fall weiß, da der dazugehörige Würfel definitiv außerhalb des Objektes liegt. Man braucht ihn also bei weiteren Projektionstests nicht mehr zu berücksichtigen. War der Knoten schwarz oder grau, und wird in diesem Durchlauf festgestellt, daß er außerhalb des Objektes liegt, so wird er als weiß markiert. Ein grau markierter Knoten bedeutet, daß der zugehörige Würfel bei einem der vorigen Bilder teilweise außerhalb der Objektfläche gelegen hat. Daher kann er nicht mehr schwarz werden, auch wenn der Würfel bei diesem Durchgang innerhalb der Objektfläche gelegen hat. Ein schwarzer Knoten bleibt nur dann schwarz, wenn der aktuelle Zustand wieder schwarz ist.

alte Farbe Testergebnis	schwarz	grau	weiß
innerhalb	schwarz	grau	weiß
ungewiß	grau	grau	weiß
außerhalb	weiß	weiß	weiß

Tabelle 1: Aktualisierung der Voxelzustände

Ist ein vollständiger Durchgang abgeschlossen, so hat man ein grobes virtuelles Modell des Objektes vorliegen. Um die Genauigkeit zu erhöhen, führt man erneut eine volle Drehung durch. Dazu wird jeder Subwürfel erneut in acht Subwürfel unterteilt. Im Baum werden diese Subwürfel durch Hinzufügen von Knoten an den entsprechenden Vaterknoten gespeichert. Untersucht werden bei der erneuten Drehung jedoch nur noch die Würfel, deren Knoten grau markiert sind. Alle Würfel mit schwarzen Knoten liegen definitiv vollständig im Objekt und müssen ebenso wie alle Würfel mit weißen Knoten, die definitiv vollständig außerhalb des Objektes liegen, nicht mehr untersucht werden.

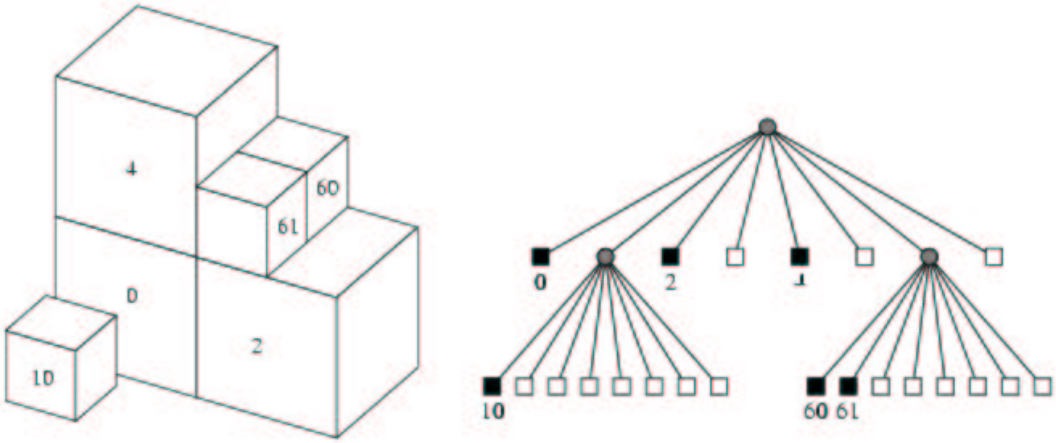

Abbildung 5: Beispiel eines Octree-Baums

Diese Grafik stellt den Baum und das dazugehörige virtuelle Objektmodell nach 2 Drehungen dar. Würfel 0, 2 und 4 liegen vollständig innerhalb des Objektes. Knoten 3, 5 und 7 liegen vollständig außerhalb des Objektes. Untersucht wurden bei der zweiten Drehung daher nur noch Knoten 1 und 6.

5.3 Vorteil der Octree-Methode

Statt mit jeder Umdrehung die Genauigkeit zu verdoppeln (= Halbierung der Kantenlänge der Voxels), könnte man auch von Anfang an mit der gewünschten Genauigkeit arbeiten und den Hauptwürfel nicht in acht, sondern in die sich aus der gewünschten Voxelauflösung ergebenden Anzahl unterteilen. Für jeden dieser Würfel müßte man dann den Projektionstest durchführen. Informationen über Räume, die die Zielwürfelgröße um ein vielfaches übersteigen und vollständig innerhalb oder außerhalb des Objektes liegen, würden bei dieser Vorgehensweise nicht gewonnen und berücksichtigt werden. Im obigen Beispiel würden z.B. bei einer Zielgenauigkeit von ¼ der Kantenlänge des Hauptwürfels ¼*¼*¼ = 1/64 → 64 Würfel betrachtet werden.

Bei Verwendung der Octree-Methode werden im obigen Beispiel jedoch nur 24 Würfel betrachtet. Auf diese Weise spart man sich also 64 – 24 = 40 Projektionstests. Allgemein ist die Knotenzahl und damit der benötigte Speicherplatz und Rechenaufwand beim Octree-Verfahren proportional zur Oberfläche des zu untersuchenden Objektes.

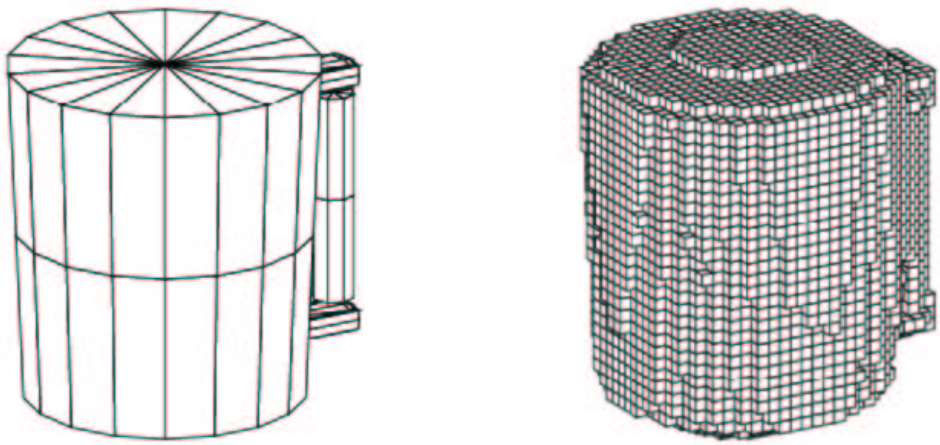

Abbildung 6: Vorgabe und Rekonstruktion

Ein Beispiele einer Tasse und ihrer Rekonstruktion mit dem Octree-Verfahren.

6 Kameraparameter und Bildanalyse

6.1 Klassifikation in Bildelemente

Zunächst werden einige Bilder ohne das Objekt aufgenommen.. Anhand der erhaltenen Werte wird für jedes Pixel der zulässige Wertebereich und die Varianz bestimmt. Fällt der Farbwert eines Pixels später außerhalb dieses Wertebereiches, gehört er mit großer Sicherheit zum Objekt.

6.2 Ermittlung der Ausrichtung

Um die aktuelle Ausrichtung des Objekttellers zu erhalten, wird an diesem ein Graycode angebracht. Die Graycodemarkierungen sind in den Farben schwarz und weiß vorgenommen. Der ungefähre Ort der Markierungen in der Bildfläche ist bekannt, außerdem kann der Algorithmus anhand den Farben ermitteln, ob eine Bildfläche zur Graycodemarkierung gehört.

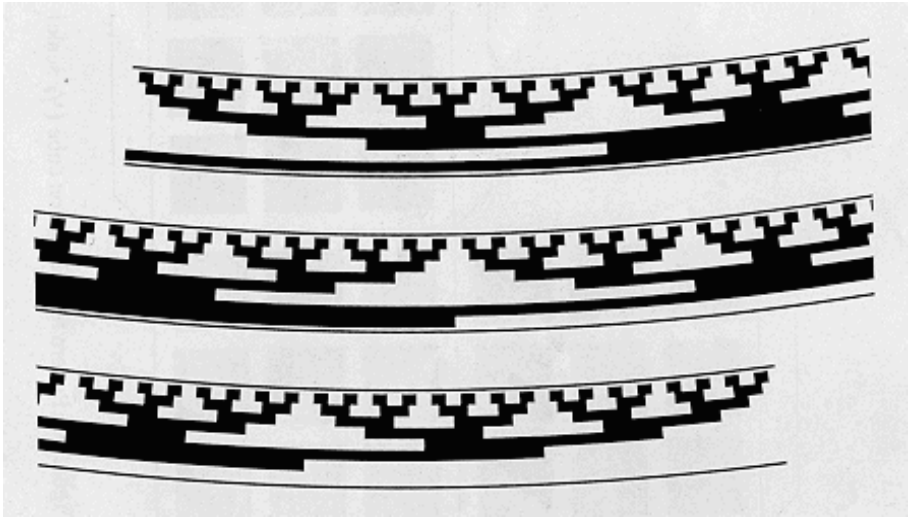

Abbildung 7: Beispiel für eine Graycodemarkierung

Um mit Hilfe des Graycodes die aktuelle Ausrichtung des Objekttellers zu ermitteln, werden im horizontal mittleren Bildbereich mehrere vertikale nebeneinander liegende Pixelstreifen untersucht. Die Anzahl der zu untersuchenden Pixelstreifen sollte so gewählt sein, daß sie zusammen breiter sind als eine Graycodemarkierung. Nun wird für jeden Pixelstreifen der Positionswert der darin enthaltenen Graycodemarkierung ermittelt. Da alle Pixelstreifen zusammen breiter sind als eine Graycodemarkierung, werden mit Sicherheit mindestens zwei Graycodemarkierungen entschlüsselt. Anhand des Übergangs zwischen diesen zwei Graycodemarkierungen kann man die aktuelle Ausrichtung des Tellers bestimmen. Bei Verwendung eines 8-bit Graycodes läßt sich damit eine ungefähre Genauigkeit von 0,1° erreichen.

Abbildung 8: Analyse der mittleren Pixelstreifen

Das obige Beispiel wird wie folgt entschlüsselt:
Für die rote Spalte erhält man den Graycode weiß-weiß-weiß-schwarz-weiß-schwarz-weiß-schwarz oder 11101010. Die gelbe Spalte rechts daneben hat einen anderen Graycode, nämlich 01101010. Also weiß man, daß sich der Übergang von Position 11101010 zu Position 01101010 zwischen dem roten und dem gelben Streifen befindet und kann daraus die aktuelle Position ermitteln.
Der Vorteil des Graycodes gegenüber dem einfachen Binärcode besteht darin, daß sich jeweils nur ein Bit ändert, wenn die nächste Position betrachtet wird. Dadurch werden unangenehme Sprünge vermieden, die man hinnehmen müßte, wenn man aufsteigende Binärzahlen als Code verwenden würde. Wäre nach einer 00000001 die nächste Position mit 00000010 markiert, also mit der nächsten Binärzahl, dann bestünde z.B. die Gefahr, daß sich die weißen Bereiche der Einsen leicht überlappen. Man würde also möglicherweise zwischen den Ausrichtungen mit den Binärcodes 00000001 und 00000010 die Ausrichtung 00000011 erhalten, was zu einem Fehler führen würde, da die Ausrichtung 00000011 in Wirklichkeit erst nach der Ausrichtung 00000010 kommt, nicht davor.
Der oben verwendete Graycode muß also noch explizit in aufsteigende Zahlen umgewandelt werden. Für einen 3-stelligen Binärcode könnte dies so aussehen:

000 = 1
001 = 2
101 = 3
111 = 4
011 = 5
010 = 6
110 = 7
100 = 8

6.3 Kameraparameter

Zur automatischen Ermittlung von Kameraposition und –ausrichtung plaziert man einen Referenzwürfel, dessen Abmessungen bekannt sind, auf dem Objektteller. Ein Kamerabild des Referenzwürfels wird aufgenommen und die Entfernung zum Objekt und die Ausrichtung der Kamera berechnet. Dies geschieht in den folgenden grafisch dargestellten Schritten:

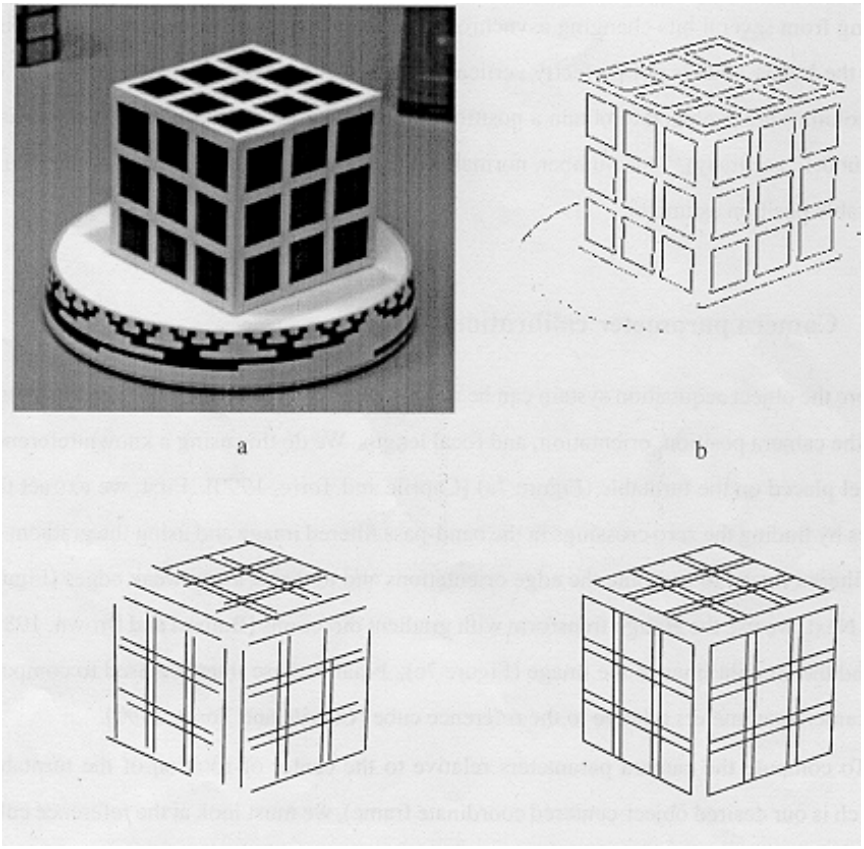

Abbildung 9: Analyse des Referenzwürfels

Links oben ist das Kamerabild dargestellt. Im Rechner werden dann zunächst die Farbunterschiede am Referenzwürfel ausgewertet und Punkte zwischen Bereichen mit großen Farbunterschieden markiert (rechts oben). Die so entstehenden Punktemengen an den Rändern der Oberflächenquadrate werden durch Linien approximiert (links unten). Ein internes Modell des Referenzwürfels wird so lange gedreht und verschoben, bis es am besten auf die Linienmenge paßt. Die Darstellung des Referenzwürfels, nachdem Ort und Ausrichtung an die Informationen aus dem Kamerabild angepaßt wurden, ist rechts unten.

7 Beschränkungen und Optimierungen des Verfahrens

7.1 Distance Vector Maps

Der Pixeltest, mit dem festgestellt wird, ob alle Pixel innerhalb der Projektionsform eines Voxels zur bzw. nicht zur Objektfläche gehören, ist sehr rechenaufwendig. Das Aufstellen einer „distance-vector map“ für jedes Bild bringt einen großen Geschwindigkeitsvorteil. Als Datenstruktur verwendet man ein Feld mit einem Eintrag für jeden Bildpunkt.

Für jeden Bildpunkt wird nun festgestellt, welche Seitenlänge das größte Quadrat hat, das gerade noch vollständig innerhalb der zum Objekt gehörenden Bildfläche liegt. Der betrachtete Bildpunkt ist dabei die linke untere Ecke dieser Quadrate. Die Seitenlänge des größten Quadrates, das diese Bedingungen erfüllt, wird in der distance-vector map gespeichert.

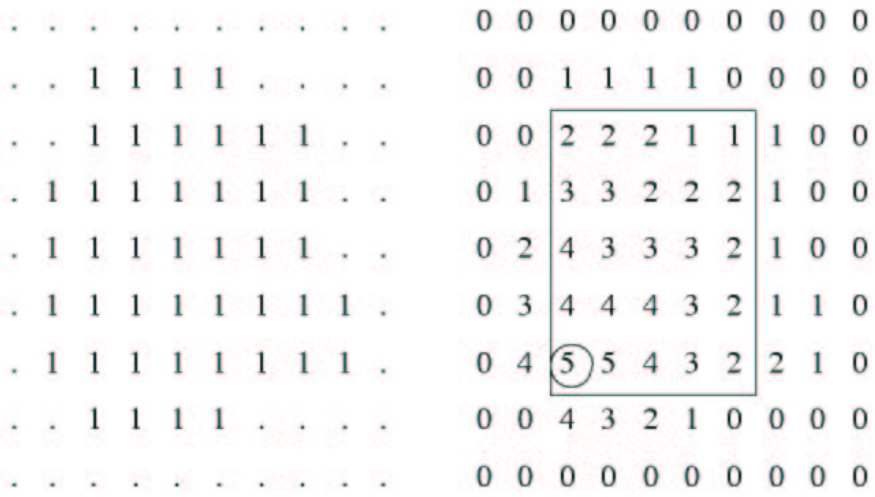

Abbildung 10: eine Objektform und ihre Distance-Vector Map

Hier ein Beispiel für eine distance-vector map. Im linken Teil stellen die Einsen die Fläche dar, die als zum Objekt gehörig erkannt wurde. Alle anderen Punkte gehören zum Hintergrund. Im rechten Teil wurde für dieses Bild die distance-vector map aufgestellt. Für den eingekreisten Bildpunkt z.B. kann man ein 5 Flächeneinheiten großes Quadrat in die Objektfläche legen, dessen linke untere Ecke auf diesem Bildpunkt liegt.
Nach der selben Methode kann man für alle Quadrate, die vollständig außerhalb der Objektfläche liegen, eine distance-vector map aufstellen
Möchte man feststellen, ob ein bestimmtes vorgegebenes Quadrat innerhalb oder außerhalb der Objektfläche liegt, schaut man zunächst nach, ob an der entsprechenden Stelle in der ersten distance-vector map ein Eintrag vorhanden ist. Ist dies der Fall, so ergeben sich zwei Möglichkeiten. Ist das Quadrat größer als dort angegeben, liegt es teilweise innerhalb der zum Objekt gehörenden Fläche. Ist es kleiner oder gleich der dort angegebenen Größe, dann liegt es vollständig innerhalb der Objektfläche. Analog kann man mit Hilfe der zweiten distance-vector map feststellen, ob ein Quadrat vollständig oder nur teilweise außerhalb der Objektfläche liegt.

7.2 Visuelle Hülle

Das beschriebene Verfahren verwendet die Silhouetten eines Objektes, um daraus ein 3D Modell zu erstellen. Für konvexe Probleme besteht das Problem, daß manche Oberflächen des zu rekonstruierenden Objektes so liegen, daß am Objektrand, also in der Silhouette des Objektes auftauchen.

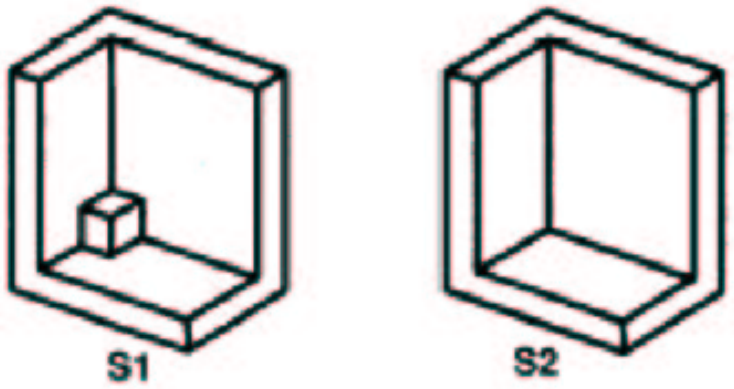

Abbildung 11: Objekte ohne rekonstruierbaren Unterschied

Hier sind 2 Objekte dargestellt. Objekt S1 und S2 haben in jeder Orientierung die gleiche Silhouette. Das oben beschriebene Verfahren kann also nicht zwischen diesen Objekten unterscheiden und damit keine korrektes virtuelles Modell erzeugen.
In [3] zeigt Richard Szeliski auf, daß das oben beschriebene Verfahren nur in der Lage ist, die Oberflächen eines Objektes korrekt wiederzugeben, die auf seiner visuellen Hülle liegen.

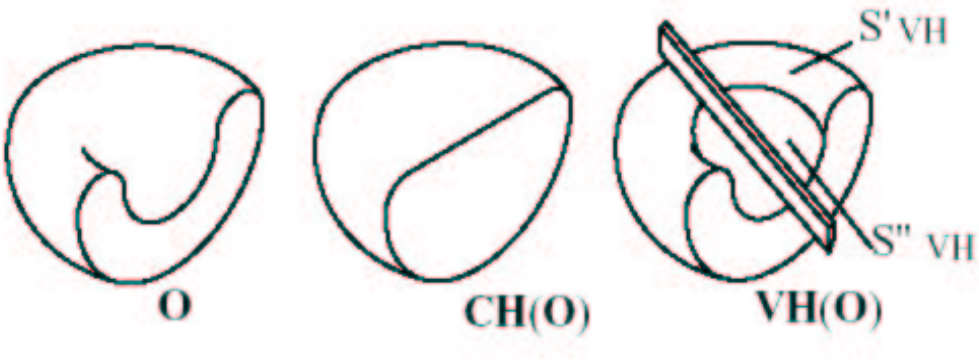

Abbildung 12: Objekt, komplexe Hülle, visuelle Hülle

Zur Veranschaulichung der visuellen Hülle dient die obige Grafik. Links ist mit O das reale Objekt dargestellt. Es handelt sich um ein konkaves Objekt, d.h., es gibt Punkte innerhalb des Objektes, deren Verbindungslinie nicht vollständig innerhalb des Objekts verläuft. In der Mitte ist mit CH(O) die konvexe Hülle dieses Objektes dargestellt. Das Objekt wird so

ergänzt, daß die Verbindungslinie zwischen zwei beliebigen Punkten in CH(O) stets vollständig in CH(O) verläuft. Daher der Begriff konvexe Hülle. Rechts ist mit VH(O) die visuelle Hülle des Objektes dargestellt. Wenn man beliebige Geraden an das Objekt O anlegt, d.h., Geraden, die das Objekt berühren aber nicht schneiden, so ist VH(O) der erweiterte Raum, der von keiner dieser Geraden geschnitten wird.
Die visuelle Hülle ist stets gleich oder kleiner als die konvexe Hülle eines Objektes. Daher gibt das beschriebene Verfahren konvexe Objekte korrekt wieder.

7.3 Erfassung der Polbereiche

Nach dem beschriebenen Modellaufbau wird das Objekt auf dem Objektteller plaziert, und rotiert. Aus der sich ergebenden Bildsequenz wird das virtuelle Objektmodell erstellt. Dies hat den Nachteil, daß die an den Polen der Rotationsachse gelegenen Objektbereiche weniger intensiv oder gar nicht betrachtet werden. Zur Veranschaulichung soll nochmals das Tassenbeispiel dienen.

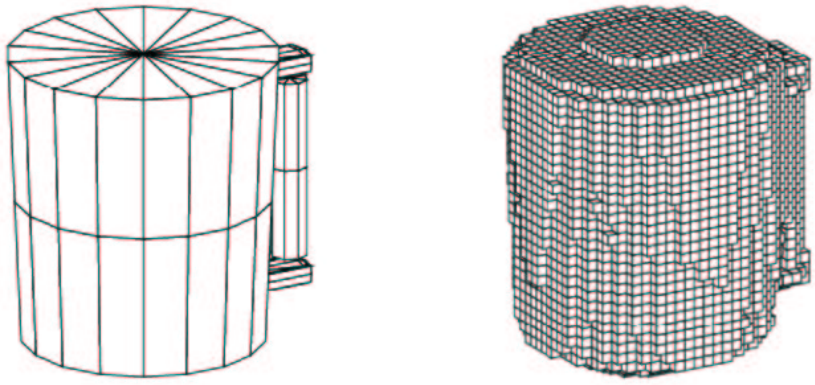

Abbildung 13: ein vorgegebenes Objekt und seine Rekonstruktion

Die Kamera befindet sich etwas über der Tasse und blickt daher von oben auf die runde Oberfläche. Diese wird als Oval in die Projektionsebene projiziert, welches sich auch bei der Drehung nicht verändert. Die Voxel, die sich im unmittelbar über der Tassenoberseite befindlichen kegelförmigen Raum befinden, werden stets in dieses Oval projiziert und deswegen für die gesamte Drehung als schwarz bzw. zum Objekt gehörend markiert.
Die folgende, aus [2] entnommene Grafik, macht dies für das Beispiel eines Elefanten deutlich:

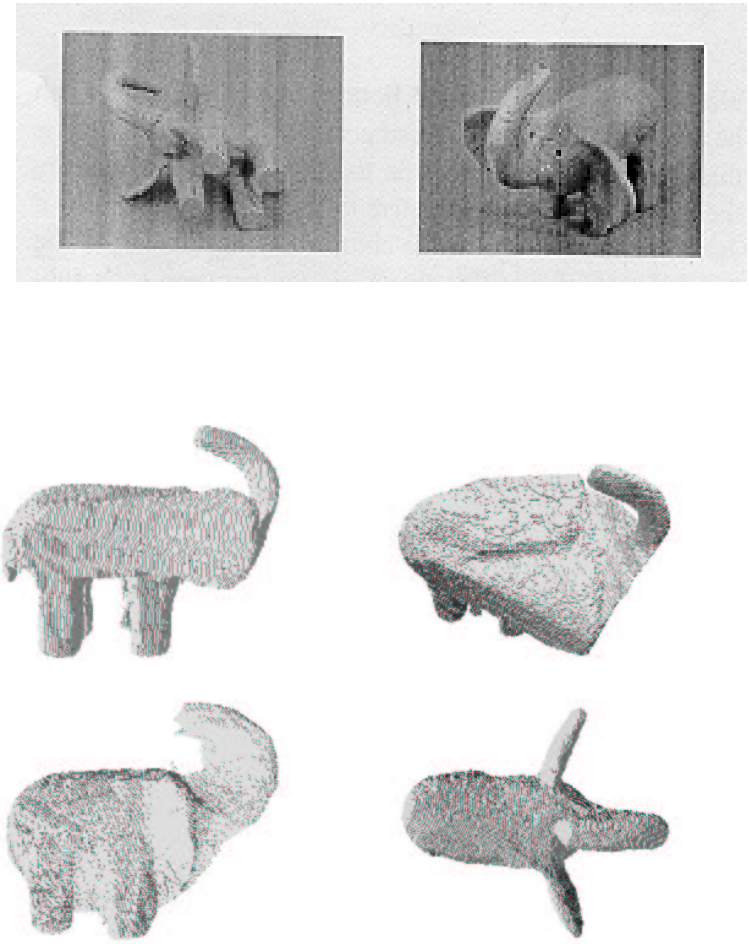

Abbildung 14: Elefant und seine Rekonstruktionen nach Drehung um verschiedene Achsen

Der Elefant wird beim ersten Durchlauf auf den Objektteller gestellt, beim zweiten Durchlauf um 90° gedreht auf den Objektteller gelegt. In der unteren der zwei oben dargestellten Grafiken ist oben das virtuelle Modell des gestellten Elefanten dargestellt, unten das Modell des gelegten Elefanten. Man erkennt, daß die Seitenansicht des gestellten Elefanten durch das Modell relativ genau wiedergegeben wird, die Ansicht von oben jedoch relativ ungenau ist. Der gelegte Elefant wird gut von oben und von unten wiedergegeben, von der Seite jedoch nicht.

In [2] wird ein Verfahren beschrieben, wie die zwei ermittelten Objekte miteinander kombiniert werden können, um ein möglichst akkurates Modell zu gewinnen.

Zuerst muß eine Transformationsmatrix H gefunden werden, die das eine virtuelle Modell so verschiebt und dreht, daß die beiden Modelle möglichst gut übereinstimmen. Als Startwerte für diese Matrix verwendet man solche, die den Masseschwerpunkt beider Objekte auf einen

gemeinsamen Punkt transformieren, und die Richtungen der Grundachsen beider Objekte übereinstimmend machen. Nun dreht man ein Objekt in kleinen Schritten um jede seiner drei Achsen und schaut, bei welcher Ausrichtung das mit *E* bezeichnete Fehlerkriterium minimal wird:

$$D_{12} = \sum_{i=1}^{N} \min\{|\mathbf{H}^{-1}\vec{p}_{1,i} - \vec{p}_{2,1}|^2, \ldots, |\mathbf{H}^{-1}\vec{p}_{1,i} - \vec{p}_{2,J}|^2\}$$

$$D_{21} = \sum_{j=1}^{N} \min\{|\mathbf{H}^{-1}\vec{p}_{2,j} - \vec{p}_{1,1}|^2, \ldots, |\mathbf{H}^{-1}\vec{p}_{2,j} - \vec{p}_{1,I}|^2\}$$

$$E = \frac{D_{12}+D_{21}}{2N} + w \cdot \min\{\frac{V_1-V_s}{V_1}, \frac{V_2-V_s}{V_2}\}$$

Jeder Punkt *p_1i* aus Modell 1 wird durch Multiplikation mit der Transformationsmatrix H gedreht und der quadratische Abstand zum nächstliegenden Punkt in Modell 2 ermittelt. Es werden dabei N Punkte auf der Oberfläche von Modell 1 betrachtet, die zufällig ausgewählt werden. In [2] wird für N bei einer Modellaufteilung in 256x256x256 Voxels der Wert 10.000 vorgeschlagen.

D_12 ist also die Summe der minimalen quadratischen Abstände der aus Modell 1 ausgewählten Punkte zu ihrem jeweils nächstliegenden Punkt in Modell 2, nachdem jeder Punkt p_1i durch Multiplikation mit H gedreht wurde.

Entsprechend ist D_21 die Summe der minimalen quadratischen Abstände der aus Modell 2 ausgewählten Punkte zu ihrem jeweils nächstliegenden Punkt in Modell 1, nachdem jeder Punkt p_2i durch Multiplikation mit der inversen Matrix von H gedreht wurde.

Das Ergebnismodell erhält man, indem man Modell 1 mit der im obigen Schritt ermittelten Matrix H multipliziert und das so transformierte Modell 1 mit Modell 2 schneidet. Voxel, die in Modell 1 und in Modell 2 liegen, werden in das Ergebnismodell übernommen.

Bei Verwendung dieses Verfahrens ergibt sich für das obige Beispiel dann das folgende, genauere Ergebnismodell:

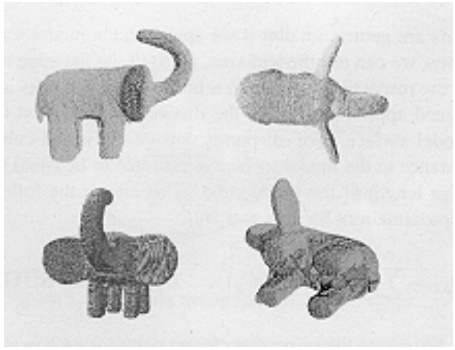

Abbildung 15: verschiedene Ansichten des Schnitts von zwei Rekonstruktionsformen

8 Zusammenfassung

Das vorliegende Verfahren ermöglicht es, eine ungefähres virtuelles Modell eines einfachen Objektes zu ermitteln. Vorteile sind die einfache Implementierbarkeit in Software, die große Geschwindigkeit des Algorithmus, die es möglich macht, das Verfahren auch in Echtzeit anzuwenden, und die automatische Erkennung der Kameraparameter, die einen einfachen und unkomplizierten Aufbau der Kameraszene garantiert.

Nachteile sind die etwas grobe Wiedergabe der realen Objektform durch die Voxeldarstellung und die Beschränkung auf die Wiedergabe der visuellen Hülle, wodurch komplexe Objekte mit geringer Genauigkeit wiedergegeben werden.

Literaturverzeichnis

[1] Richard Szeliski, Real-Time Octree Generation from Rotating Objects, 1990

[2] Jochen Wingbermühle, Robust Registration of Coarse Binary Volume Models

[3] Richard Szeliski, From Images to Models: A Personal Retrospective

[4] Wolfgang Niem, Automatische Rokonstruktion starrer dreidimensionaler Objekte aus Kamerabildern, 1999