

# Kapitel 16

## 3D-Repräsentation

Für 3-dimensionale Objekte gibt es mehrere Möglichkeiten der Repräsentation (d.h. Definition des Objekts) und der Darstellung (d.h. Projektion des Objekts auf den Bildschirm auch *rendering* genannt):

### Repräsentation

- Elementarobjekt mit Definitionspunkten,
- Drahtmodell,
- Flächenmodell mit Punkt- und Flächenliste und Normalen,
- CSG (constructive solid geometry) mit mengentheoretischer Verknüpfung von Elementarobjekten.

### Darstellung

- Punktmodell
- Drahtmodell mit sämtlichen Kanten,
- Drahtmodell mit Entfernung verdeckter Kanten,
- Flächenmodell mit Einfärbung, ohne abgewandte Flächen,
- Flächenmodell mit Einfärbung, ohne verdeckte Teile von Flächen,
- Flächenmodell mit Einfärbung, ohne verdeckte Teile von Flächen, mit Beleuchtungsmodell
- Körpermodell mit Berechnung von Schattenbildung, Spiegelungen und Brechungen.

Zur Berechnung der Darstellung wird die Viewing-Pipeline benötigt, welche eine Transformation der Definitionspunkte vornimmt (Kapitel 15), die nicht sichtbaren Flächen entfernt (Kapitel 17) sowie das Rastern der projizierten Flächen durchführt (Kapitel 18).

In diesem Kapitel werden die verschiedenen Repräsentationsarten behandelt und einige dazu gehörende Beispiele.

## 16.1 Elementarobjekte

Für den Benutzer sollte die Beschreibung einer Szene durch *Elementarobjekte* erfolgen. Diese können unterschiedlich kompliziert sein, sollten aber durch wenige Parameter beschrieben werden können (Kugel, Quader, Pyramide, Kegel, Zylinder,...). Eine Kugel z.B. ist bereits durch Mittelpunkt und Radius eindeutig im Raum plaziert. Es ist sinnvoll, jedes Objekt in seinem eigenen, lokalen Modellkoordinatensystem zu definieren. Dessen Ursprung wird im Inneren des Objekts gewählt und das gesamte Objekt in ein Einheitsvolumen (z.B.  $-1 \leq x, y, z \leq +1$ ) eingeschlossen. Für Orts- und Größenveränderungen sind Transformationen zuständig.

## 16.2 Drahtmodell

In der nächsten Repräsentationsklasse wird das Elementarobjekt als *Drahtmodell* durch eine Liste von Kanten repräsentiert. Jede Kante besteht aus zwei Punkten im kartesischen Koordinatensystem. Beim Würfel verbinden die Kanten die Eckpunkte, bei einer Kugel werden die Längen- und Breitenkreise durch  $n$ -Ecke angenähert, wobei mit  $n$  die Güte der Approximation steigt. Das Drahtmodell skizziert nur die Umrisse eines Objekts und enthält keine zusätzlichen Flächen- oder Volumeninformationen.

## 16.3 Flächenmodell

Es werden Objekte durch approximierte oder analytische Flächen, z.B. durch eine Liste von konvexen Polygonen, repräsentiert. Ein solches Polygon wird durch seine Eckpunkte beschrieben, die durch Kanten verbunden sind und eine Fläche umschließen.

Beim Würfel verbinden die Kanten die Eckpunkte, bei einer Kugel werden die Längen- und Breitenkreise durch  $n$ -Ecke angenähert, wobei mit  $n$  die Güte der Approximation steigt.

| Punktliste              | Flächenliste          |
|-------------------------|-----------------------|
| $P_1 : (x_1, y_1, z_1)$ | $F_1 : p_1, p_2, p_4$ |
| $P_2 : (x_2, y_2, z_2)$ | $F_2 : p_1, p_4, p_3$ |
| $P_3 : (x_3, y_3, z_3)$ | $F_3 : p_1, p_3, p_3$ |
| $P_4 : (x_4, y_4, z_4)$ | $F_4 : p_4, p_2, p_3$ |

Zur vollständigen Beschreibung einer Fläche gehört noch die Angabe, welche Seite “innen” und welche Seite “außen” liegt. Dies geschieht durch Angabe des Normalenvektors: Er steht senkrecht auf der Fläche und zeigt von innen nach außen. Für die Approximation gekrümmter Flächen wird häufig pro Eckpunkt eine Normale verwendet.

## 16.4 Flächenmodell mit Halbkantendarstellung

Jedes Objekt enthält eine Liste von Flächen, die von Halbkanten begrenzt werden. Die Halbkanten sind von außen betrachtet im Uhrzeigersinn orientiert und zeigen auf ihre jeweils linke Nachbarfläche sowie ihre Anfangs- und Endpunkte.

Die Halbkantendarstellung eignet sich zur effizienten Entfernung von verdeckten Kanten und Flächen. Eine Kante zwischen Punkt  $P_1$  und Punkt  $P_2$ , welche die Flächen  $F_1$  und  $F_2$  trennt, taucht einmal als Halbkante  $(P_1, P_2)$  in der Kantenliste zu  $F_2$  auf mit einem Verweis auf die Nachbarfläche  $F_1$  und ein weiteres Mal als  $(P_2, P_1)$  in der Kantenliste zu  $F_1$  mit einem Verweis auf die Nachbarfläche  $F_2$ . Werden nun alle Halbkanten einer Fläche  $F_1$  bearbeitet, so regelt die Sichtbarkeit von  $F_1$  und die Sichtbarkeit der jeweils anstoßenden Fläche die Sichtbarkeit der jeweiligen Halbkante. Das doppelte Zeichnen einer Kante läßt sich vermeiden, indem bei jeder Fläche vermerkt wird, ob ihre Halbkanten bereits bearbeitet wurden.

Auch die Flächennormalen können in der Datenstruktur gespeichert werden.

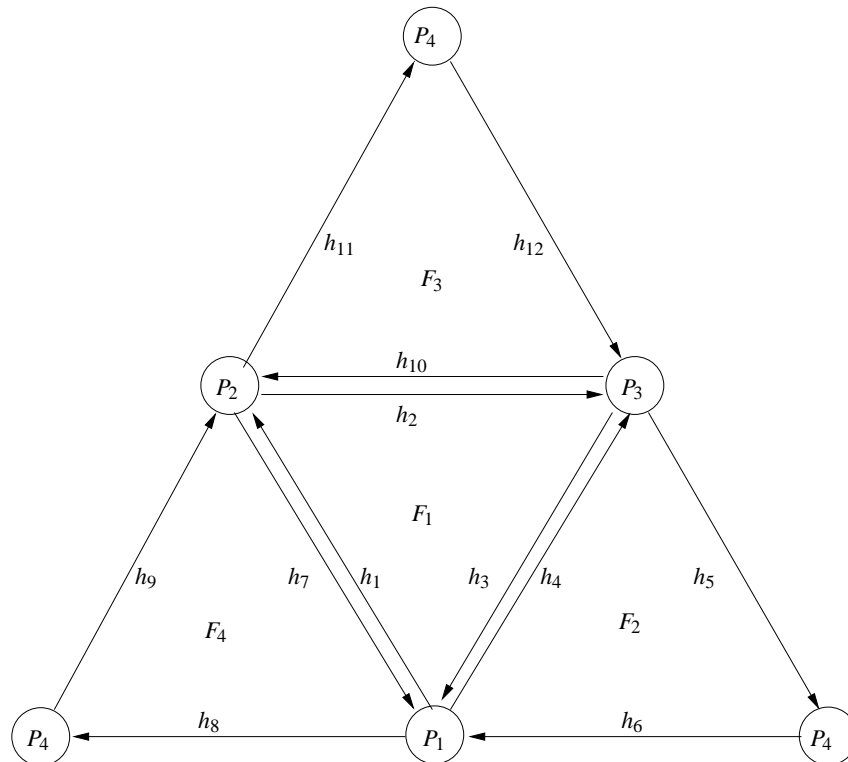


Abbildung 16.1: Tetraeder mit 4 Knoten, 12 Halbkanten, 4 Flächen

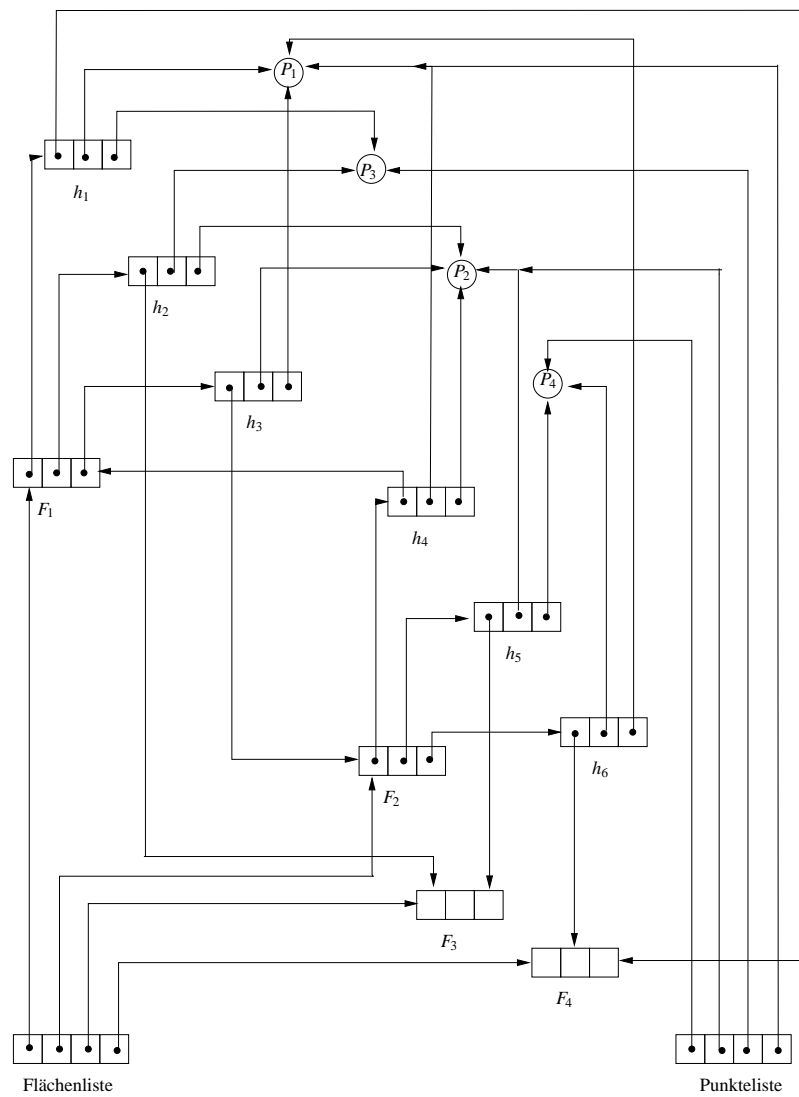


Abbildung 16.2: Objektstruktur für Tetraeder. Nicht gezeichnet sind Halbkanten für  $F_3$  und  $F_4$

## 16.5 Polyeder

Abbildung 16.3 zeigt eine vom Projektionsalgorithmus erzeugte Szene mit Würfel und Tetraeder in der Drahtmodell-Darstellung mit gestrichelten Rückkanten.

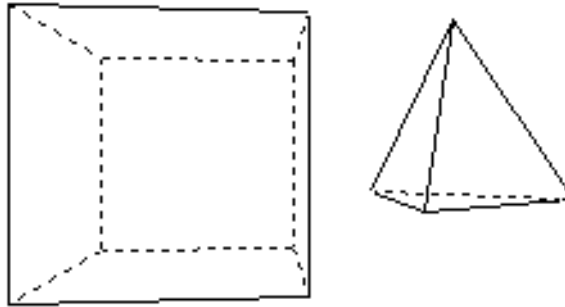


Abbildung 16.3: Vom Projektionsalgorithmus berechnete Drahtgitter-Darstellung

Ein *Würfel* mit der Kantenlänge 1 hat im MC die homogenen Eckpunktkoordinaten

$$\begin{array}{ll}
 (+0.5, +0.5, +0.5, 1), & (+0.5, +0.5, -0.5, 1), \\
 (+0.5, -0.5, +0.5, 1), & (+0.5, -0.5, -0.5, 1), \\
 (-0.5, +0.5, +0.5, 1), & (-0.5, +0.5, -0.5, 1), \\
 (-0.5, -0.5, +0.5, 1), & (-0.5, -0.5, -0.5, 1).
 \end{array}$$

Da jede der sechs Flächen des Würfels eben ist, gibt es pro Fläche nur einen Normalenvektor, der durch das Kreuzprodukt zweier benachbarter Kanten berechnet werden kann. Die Reihenfolge der Punkte ist dabei signifikant, da jeweils aufeinanderfolgende Punkte eine Kante bilden. Ein *Quader* entsteht aus dem Würfel durch ungleichmäßige Skalierung beim Modeling.

Auch alle anderen Polyeder werden im MC in der Flächendarstellung definiert, ein Tetraeder beispielsweise mit Kantenlänge 1 und Schwerpunkt im Ursprung.

## 16.6 Gekrümmte Flächen

Eine gekrümmte Oberfläche, die analytisch beschrieben werden kann, lässt sich beliebig gut durch Polygone approximieren. Die Gruppe der Objekte mit analytisch gekrümmter Oberfläche besteht aus zwei Typen: Während z.B. die Kugel nur aus einer solchen Fläche besteht, sind Zylinder und Kegel aus mehreren Flächen zusammengesetzt.

Die Repräsentation der gekrümmten Körper erfolgt im Flächenmodell mit einer Flächen- und Eckpunktliste sowie einer Normalenliste, die pro Eckpunkt die zugehörige Normale enthält.

## 16.7 Zylinder

Der zur  $z$ -Achse symmetrische *Zylinder* mit Höhe 2 und Radius 1 besteht aus einer Mantelfläche und zwei Deckflächen. Die beiden Kreisscheiben bei  $z = 1$  und  $z = -1$  lassen sich durch regelmäßige  $n$ -Ecke darstellen. Die Mantelfläche wird durch  $n$  Rechtecke approximiert. Dabei wachsen mit  $n$  sowohl die Genauigkeit der Approximation als auch der Rechenaufwand. Sei  $\alpha = (2\pi)/n$ , so lauten für ein solches Rechteck die vier Eckpunkte und die zugehörigen Normalenvektoren:

| Eckpunkt  | Normalenvektor                                     |
|---|--|
| $(\cos(\phi), \sin(\phi), +1, 1)$                   | $(\cos(\phi), \sin(\phi), 0, 0)$                   |
| $(\cos(\phi + \alpha), \sin(\phi + \alpha), +1, 1)$ | $(\cos(\phi + \alpha), \sin(\phi + \alpha), 0, 0)$ |
| $(\cos(\phi + \alpha), \sin(\phi + \alpha), -1, 1)$ | $(\cos(\phi + \alpha), \sin(\phi + \alpha), 0, 0)$ |
| $(\cos(\phi), \sin(\phi), -1, 1)$                   | $(\cos(\phi), \sin(\phi), 0, 0)$                   |

mit  $\phi = k \cdot \alpha$ ,  $k \in \{0, \dots, n-1\}$ .

Als Normalenvektor in einem Eckpunkt dieser Fläche wird also der tatsächliche Normalenvektor der Mantelfläche genommen. Dadurch erhalten aneinandergrenzende Flächen in ihren gemeinsamen Eckpunkten auch denselben Normalenvektor. Auf diese Weise erzeugt das Programm bei der späteren Beleuchtung den Eindruck eines stetigen Übergangs zwischen den Flächen. Der Betrachter nimmt statt einer  $n$ -eckigen Säule den Zylinder wahr.

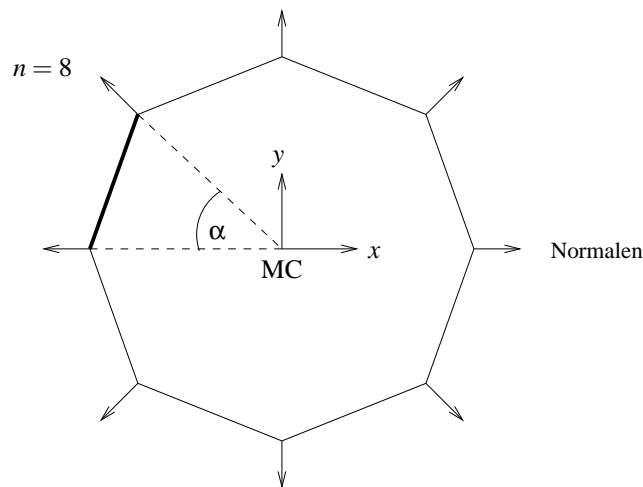


Abbildung 16.4: Zylinder im Grundriß mit Normalenvektoren der Mantelfläche

## 16.8 Kugel

Die Oberfläche einer *Kugel* mit Radius 1 kann beschrieben werden durch

$$(\sin(\theta) \cos(\phi), \sin(\theta) \sin(\phi), \cos(\theta)), 0 \leq \phi < 2\pi, 0 \leq \theta < \pi.$$

Zur Approximation durch Flächen wird der Vollwinkel in  $n$  Teile zerlegt:

$$\alpha = (2\pi)/n, \quad n \in \mathbb{N} \text{ gerade}.$$

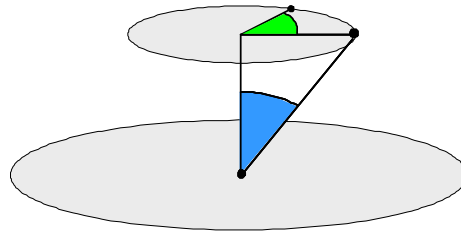


Abbildung 16.5: Konstruktion einer Kugel mit Radius 1 im Ursprung

Dadurch entstehen auf der Kugel  $n/2$  gleichgroße Längengrade und  $(n/2 - 1)$  verschieden große Breitenkreise. Diese schneiden  $n$  Dreiecke an jedem Pol und  $n(n/2 - 2)$  Vierecke aus der Kugeloberfläche. Die Ortsvektoren der Eckpunkte eines Dreiecks am Nordpol ( $\theta = 0$ ) lauten

$$(0, 0, +1, 1), (\sin(\alpha) \cos(\phi), \sin(\alpha) \sin(\phi), \cos(\alpha), 1), (\sin(\alpha) \cos(\phi + \alpha), \sin(\alpha) \sin(\phi + \alpha), \cos(\alpha), 1),$$

mit  $\phi = k \cdot \alpha, k \in \{0, \dots, n - 1\}$ .

Die Eckpunkte eines der Vierecke haben die Ortsvektoren

$$\begin{aligned} &(\sin(\theta) \cos(\phi), \sin(\theta) \sin(\phi), \cos(\theta), 1) \\ &(\sin(\theta) \cos(\phi + \alpha), \sin(\theta) \sin(\phi + \alpha), \cos(\theta), 1) \\ &(\sin(\theta + \alpha) \cos(\phi + \alpha), \sin(\theta + \alpha) \sin(\phi + \alpha), \cos(\theta + \alpha), 1) \\ &(\sin(\theta + \alpha) \cos(\phi), \sin(\theta + \alpha) \sin(\phi), \cos(\theta + \alpha), 1) \end{aligned}$$

mit  $\phi = k \cdot \alpha, k \in \mathbb{N}, k < n$  und  $\theta = l \cdot \alpha, l \in \mathbb{N}, 0 < l < (n/2 - 1)$ .

Als Normalenvektor wird in jedem Eckpunkt der Ortsvektor als Richtungsvektor ( $w = 0$ ) eingetragen, denn der Radiusvektor steht senkrecht auf der Kugeloberfläche. Einen *Ellipsoid* erzeugt der Rendering-Algorithmus aus der Kugel durch ungleichmäßige Skalierung beim Modeling.

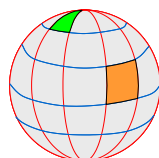


Abbildung 16.6: Breiten- und Längengrade einer Kugel-Approximation

## 16.9 Bezier-Flächen

Die in Kapitel 7.3 eingeführten Bezier-Kurven können auch zur Definition einer gekrümmten Fläche im Raum verwendet werden. Da eine Fläche zweidimensional ist, muß eine weitere Parameterdimension hinzugefügt werden, d.h. statt  $0 \leq t \leq 1$  gilt nun  $0 \leq u, v \leq 1$ . Im folgenden Beispiel werden 16 Kontrollpunkte mit kubischen Bernsteinpolynomen gewichtet:

$$P(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 \cdot B_{i,3}(u) \cdot B_{j,3}(v) \cdot P_{i,j}$$

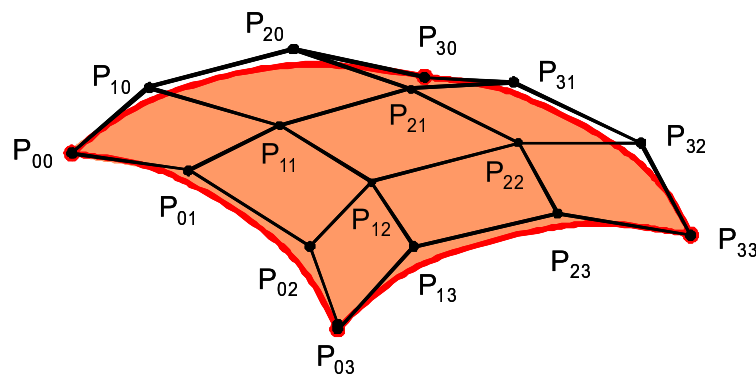


Abbildung 16.7: Gitternetz mit 16 Kontrollpunkten für Bezier-Fläche im Raum

Beim Verlängern einer Bezier-Fläche ist darauf zu achten, dass die neue Reihe von Kontrollpunkten tangential die bisherige Fläche verlängert, d.h.  $P_{i,2}, P_{i,3} = Q_{i,0}, Q_{i,1}$  sind collinear und das Verhältnis der Abstände  $|P_{i,3} - P_{i,2}| / |Q_{i,1} - Q_{i,0}|$  ist konstant.

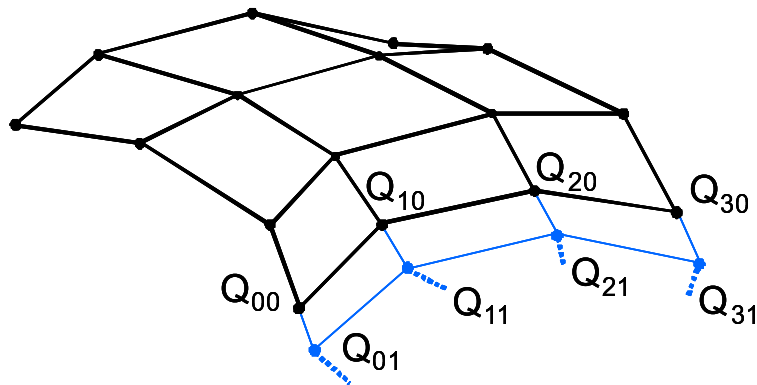


Abbildung 16.8: Verlängerung einer Bezierfläche



## 16.10 NURBS-Flächen

Eine NURBS-Fläche sieht in homogenen Koordinaten folgendermaßen aus:

$$\vec{q}^h(u, w) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j}^h N_{i,k}(u) M_{j,l}(w)$$

Die  $P_{i,j}^h$  sind die vierdimensionalen homogenen Stützpunkte. Die  $N_{i,k}$  und die  $M_{j,l}$  sind die nicht-rationalen B-Spline-Basisfunktionen aus Kapitel 7. Um affine Koordinaten zu erreichen, wird wieder durch die homogenen Koordinaten dividiert:

$$\vec{q}(u, w) = \frac{\sum_{i=0}^n \sum_{j=0}^m h_{i,j} P_{i,j} N_{i,k}(u) M_{j,l}(w)}{\sum_{i=0}^n \sum_{j=0}^m h_{i,j} N_{i,k}(u) M_{j,l}(w)} = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} S_{i,k}(u, w)$$

wobei  $P_{i,j}$  die dreidimensionalen Stützpunkte sind. Sie ergeben das *Kontrollnetz* für die Oberfläche. Bei den  $S_{i,j}(u, w)$  handelt es sich um die biparametrisierten Flächenbasisfunktionen

$$S_{i,j}(u, w) = \frac{h_{i,j} N_{i,k}(u) M_{j,l}(w)}{\sum_{i=0}^n \sum_{j=0}^m h_{i,j} N_{i,k}(u) M_{j,l}(w)}$$

Die  $S_{i,j}(u, w)$  sind nicht das Produkt der  $R_{i,k}(u)$  und der  $R_{j,l}(w)$  aus Kapitel 7. Sie haben aber ähnliche analytische Eigenschaften, so daß die NURBS-Fläche ähnliche analytische und geometrische Eigenschaften, wie die früher erwähnten NURBS-Kurven:

- Die  $S_{i,j}(u, w)$  addieren für ein festes Paar von  $u$  und  $w$  zu 1.
- $S_{i,j}(u, w) \geq 0 \forall u, w$
- Der maximale Grad der Basispolygone ist gleich der Zahl der Kontrollpunkte -1 in der entsprechenden Dimension.
- Eine NURBS-Kurve der Ordnung  $k, l$  (Grad  $k-1, l-1$ ) ist überall  $C^{k-2}, C^{l-2}$  stetig.
- NURBS-Flächen sind invariant bzgl. perspektivischer Projektion. Es genügt, die Stützpunkte mit der Viewing Pipeline zu transformieren und die eigentliche Interpolation im DC vorzunehmen.
- Wenn gilt  $h_{i,j} \geq 0 \forall i, j$ , liegt die Fläche innerhalb der konvexen Hülle des Kontrollnetzes.
- Der Einfluß eines Stützpunktes ist begrenzt auf  $\pm k/2, \pm l/2$  Interpolationsabschnitte in jeder Parameterdimension.
- Um die Normale der Fläche in einem beliebigen Punkt zu berechnen, müssen die Richtungsableitungen bzgl.  $u$  und  $w$  in diesem Punkt gebildet werden.

## 16.11 CSG (constructive solid geometry)

Sollen die beschriebenen Objekte auch im physikalischen Sinne realisierbar sein (zum Beispiel über wohldefinierte Volumen verfügen), so müssen die gespeicherten Oberflächen zusätzliche Eigenschaften erfüllen. Die zugrundeliegende Theorie heißt CSG (constructive solid geometry). Um solche Objekte zu erzeugen, beginnt man mit wohldefinierten Objekten und erzeugt durch regularisierte Mengenoperationen  $\cup^*$  (Vereinigung),  $\cap^*$  (Durchschnitt),  $\setminus^*$  (Differenz) neue, wiederum physikalisch sinnvolle Objekte. Hierbei bedeutet für die mengentheoretische Operation  $op \in \{\cup, \cap, \setminus\}$  die regularisierte Operation  $op^*$  den Abschluss des Inneren der Verknüpfung von A mit B:

$$A \text{ op}^* B = \text{closure}(\text{interior}(A \text{ op} B))$$

Die hierarchische Struktur wird beschrieben durch einen binären Baum, dessen Blätter beschriftet sind mit den Elementarobjekten und dessen innere Knoten beschriftet sind mit den regularisierten Mengenoperationen  $\cup^*$  (Vereinigung),  $\cap^*$  (Durchschnitt),  $\setminus^*$  (Differenz). Die Wurzel repräsentiert das resultierende Objekt, welches aus den Elementarobjekten an den Blättern unter Anwendung der Operationen an den inneren Knoten konstruiert werden kann.

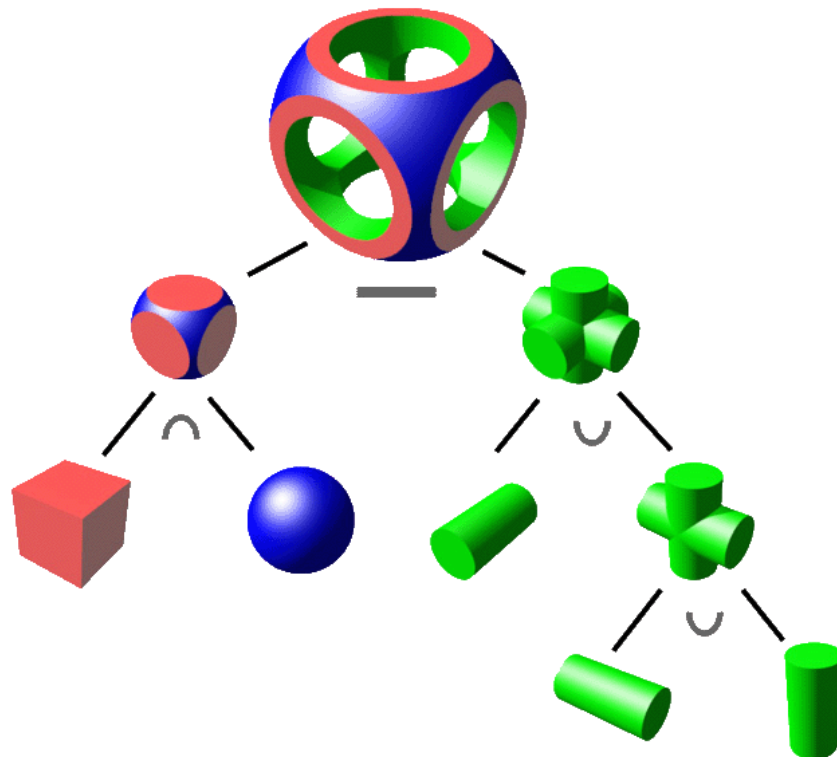


Abbildung 16.9: Darstellung der Verknüpfungshierarchie (Beispiel aus Wikipedia)

## 16.12 Octree

Zur Verwaltung der räumlichen Anordnung von Objekten im dreidimensionalen Raum eignet sich der *Octree*. Es handelt sich um die Erweiterung um die dritte Dimension des 2-dimensionalen *Quadtree*.

Abbildung 16.10 zeigt die Platzierung von 4 schwarzen Rechtecken A,B,C,D in der Ebene. Der zugehörige Quadtree teilt rekursiv die repräsentierte Fläche in 4 Quadranten ein und notiert an den Knoten, ob er nichts enthält (weiss), teilweise Objekte enthält (grau) oder ein Rechteck repräsentiert (schwarz).

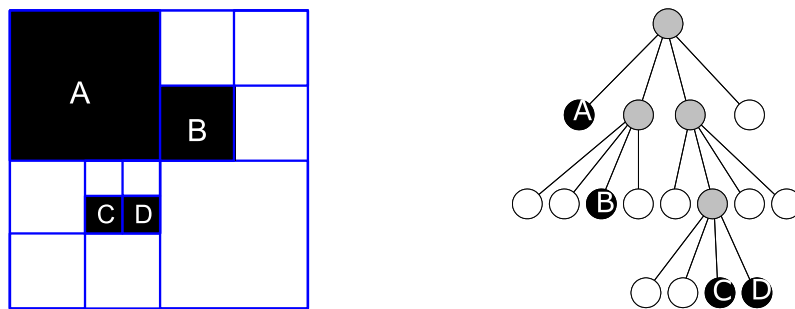


Abbildung 16.10: 4 Rechtecke in der Ebene mit zugehörigem Quadtree

Abbildung 16.11 zeigt die Platzierung von drei Würfeln A,B,C im Raum. Der zugehörige Octree teilt rekursiv den repräsentierten Raum in 8 Oktanten ein und notiert an einem Knoten, ob er nichts enthält (weiss), teilweise Objekte enthält (grau) oder einen Quader repräsentiert (farbig).

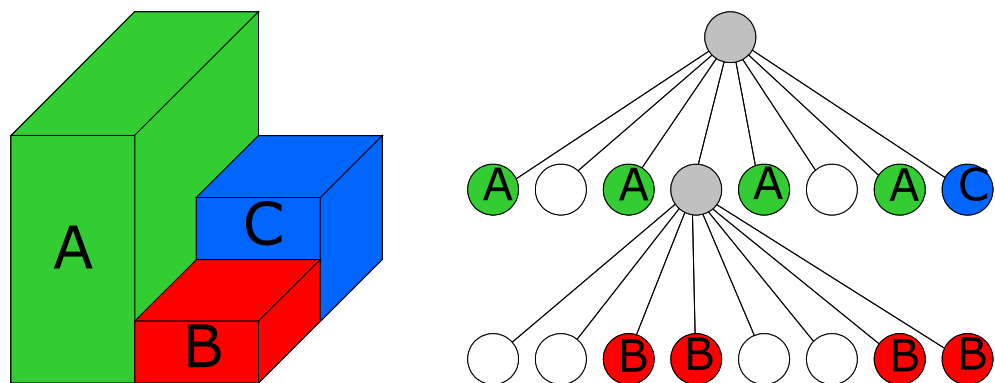


Abbildung 16.11: 3 Quader im Raum mit zugehörigem Octree

### 16.13 Java-Applet zur Wire-Frame-Projektion

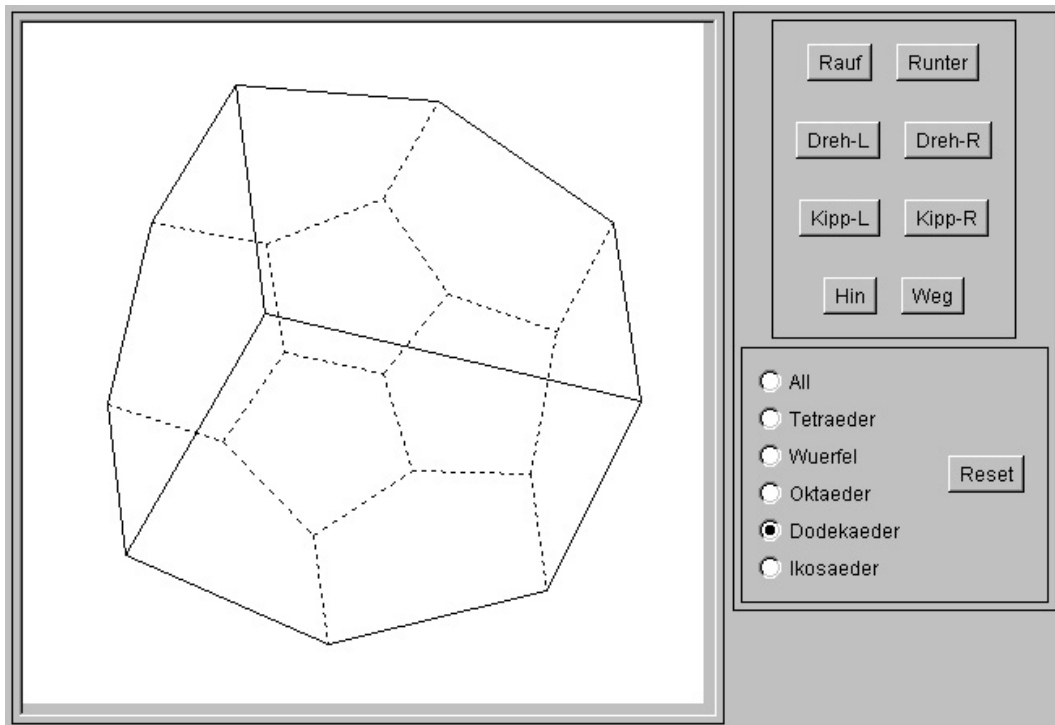


Abbildung 16.12: Screenshot vom 3D-wire-Applet