

3D Objektrepräsentationen

Dieses Kapitel beschäftigt sich mit der Frage, wie dreidimensionale Objekte im Computer repräsentiert werden, also mit den Konzepten und Datenstrukturen. Bereits im Kapitel „Graphikprimitive“ haben wir die polygonale Repräsentation von Objekten (Boundary Representation = BRep) als Konzept die Polygonlisten und als zugehörige Datenstruktur kennengelernt. Hier werden nun weitere Repräsentationsformen beschrieben.

■ Constructive Solid Geometry (CSG)

Bei CSG werden Objekte aus dreidimensionalen Primitiven mit Hilfe von Mengenoperationen konstruiert. Sie werden so angeordnet, dass sie in einer hierarchischen Datenstruktur angeordnet werden können, die – obwohl es sich eigentlich nur um einen kreisfreien Graphen handelt – normalerweise CSG-Baum genannt wird. Als Primitive dienen einfach geometrische Formen wie Kugel, Tetraeder, Würfel, Zylinder, die mit den Operatoren Vereinigung, Durchschnitt und Differenz verknüpft werden. Da alle Primitive trivialerweise konsistent sind und die Operatoren aus konsistenten Teilen nur konsistente Objekte erzeugen, sind bei CSG alle Objekte immer konsistent (keine Löcher in der Oberfläche, wohldefiniertes Inneres).

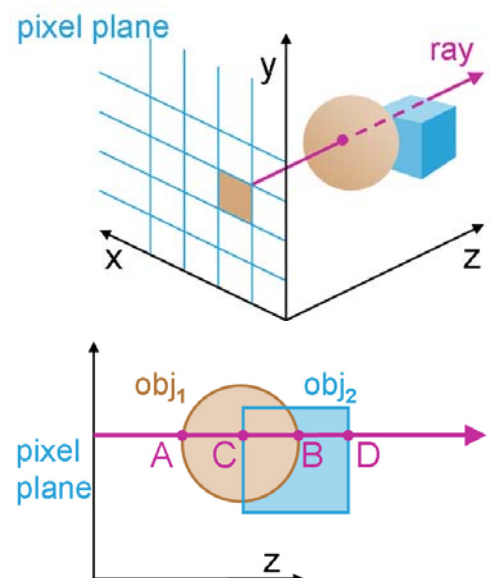
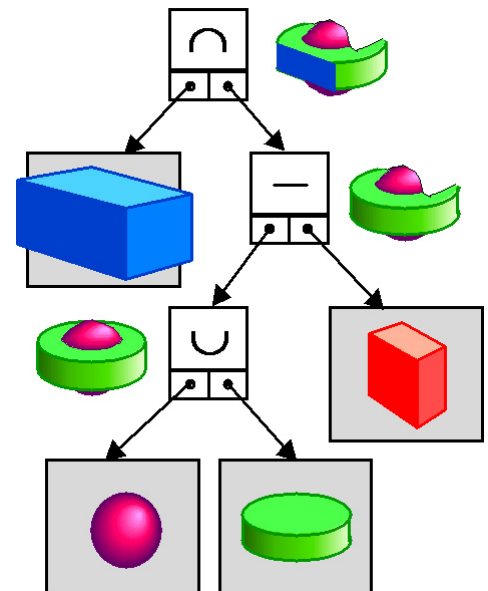
Zusätzlich enthält jeder Knoten eines CSG-Baums noch Transformationen (in Form von Matrizen), die angeben, welche Transformationen auf den darunter befindlichen Teilbaum angewendet werden. Primitive (z.B. achsenparalleler Einheitswürfel) erhalten dadurch ein weites Spektrum an Formen (z.B. beliebig im Raum positionierter Quader), und auch jedes komplexere Objekt kann noch verschoben, skaliert, gedreht usw. werden.

Vorteile von CSG-Objekten ist die exakte Repräsentation (eine Kugel ist wirklich eine Kugel!), der geringe Speicherbedarf und die Einfachheit von Transformationen. Der Hauptnachteil ist die wesentlich aufwändigere Berechnung von Bildern, also das kompliziertere Rendering. Dazu muss man entweder die Datenstruktur in eine BRep-Repräsentation umwandeln und auf herkömmliche Art rendern, oder man verwendet Ray-Casting oder Ray-Tracing zur direkten Bilderstellung.

Ray-Casting von CSG-Objekten

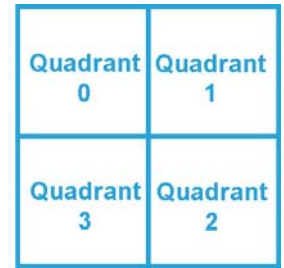
Die gebräuchlichste Methode um CSG-Objekte abzubilden ist das Ray-Casting, bei dem das Bild pixelweise berechnet wird. Für jedes Pixel wird in Blickrichtung ein Strahl (Ray) gelegt („auswerfen“ = to cast) und mit allen Objekten der Szene geschnitten. Der vorderste dieser Schnittpunkte gibt an, welches Objekt in diesem Pixel zu sehen ist, und das Pixel erhält dessen Farbe. Bei einem CSG-Baum erfolgt diese Berechnung rekursiv:

- bei **Endknoten** ist die Berechnung aller Schnittpunkte einfach,
- bei **Zwischenknoten** werden die Schnittpunktlisten der beiden Nachfolger entsprechend dem Operator verknüpft:
 aus den Listen (A,B) und (C,D) im Beispiel rechts entsteht
 bei Vereinigung die Liste (A,D),
 bei Durchschnitt die Liste (C,B),
 bei Differenz die Liste (A,C).
- bei der **Baumwurzel** wird der erste Punkt der verknüpften Schnittpunktliste ausgewählt.

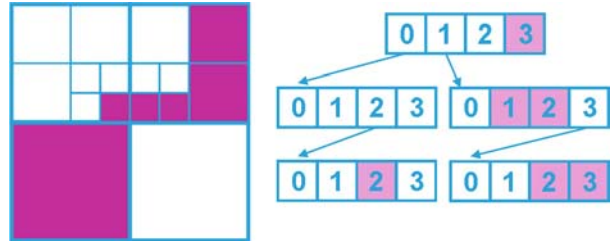


■ Quadrees und Octrees

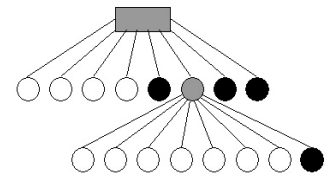
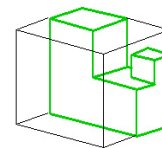
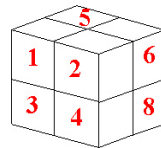
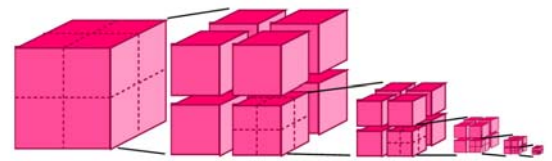
Ein *Quadtree* ist eine Datenstruktur, die zur Repräsentation beliebiger zweidimensionaler Strukturen geeignet ist. Der relevante Bereich wird überall dort in vier Viertel geteilt, wo die Information noch zu kompliziert ist um einfach abgelegt zu werden, andernfalls wird die einfache Information abgelegt. Jedem Bildbereich entspricht ein Knoten eines Baumes, in dem jeder Knoten (maximal) vier Nachfolger hat („Quadtree“).



Das nebenstehende Beispiel zeigt einen einfachen Quadtree, der eine zweifarbige einfache Graphik repräsentiert. Der Wurzelknoten entspricht dem ganzen Bild, die Knoten in der zweiten Reihe entsprechen den oberen 2 Vierteln des Bildes und die letzten 2 Knoten entsprechen den 2 Bereichen, die am feinsten aufgelöst sind.



Ein *Octree* ist die Erweiterung dieses Konzeptes auf drei Dimensionen. Ein beliebig geformtes Objekt (oder auch eine ganze Szene) innerhalb eines Würfels wird dadurch repräsentiert, dass „einfache“ Teilwürfel (leer oder ganz innerhalb eines Objektes) durch Endknoten beschrieben werden, und kompliziertere Teilwürfel (alle anderen!) in acht kleinere Teilwürfel (Oktanten) unterteilt werden, auf die wieder dieselben Regeln angewendet werden (rekursiv). Dadurch entsteht ein Baum, in dem jeder Knoten 8 Nachfolger hat („Octree“). Man hört mit der Unterteilung auch dann auf, wenn die Teilwürfel eine bestimmte Mindestgröße unterschreiten (z.B. ein Tausendstel der Gesamtgröße); in diesem Fall erhält der Knoten des Baumes die bestmögliche einfache Information. Das passiert zumindest bei allen schrägen Oberflächen irgendwann, und dann müssen diese Randwürfel entweder als innerhalb oder als außerhalb deklariert werden. Das nebenstehende Beispiel mit nur zwei Ebenen zeigt auch das Problem der beschränkten räumlichen Auflösung.



Octrees haben den Vorteil, dass man beliebige Formen repräsentieren kann und dass man schnell untersuchen kann, was sich an einer bestimmten räumlichen Position befindet. Dem stehen als Nachteile gegenüber: ungenaue Repräsentation, hoher Speicherbedarf, komplizierte Transformationen. Octrees werden genauso wie Quadrees rekursiv bearbeitet. Mengenoperationen sind so ganz einfach, dafür sind geometrische Transformationen (von Ausnahmen abgesehen) sehr aufwändig, weil der Octree komplett neu generiert werden muss. Das Rendering von Octrees ist dagegen bei Verwendung eines überschreibbaren Speichers einfach:

```

if Knoten ist einfach
then zeichne Knoten (d.h. tue nichts wenn Knoten leer ist)
else rekursiver Aufruf der 8 Oktanten von hinten nach vorne
    
```

■ Andere Objektrepräsentationen

Es gibt noch eine Fülle von anderen Objektrepräsentationen und zugehörigen Datenstrukturen, die zum Teil für sehr spezielle Objekte und Anwendungen entwickelt wurden. Dazu zählen BSP-Bäume, Fraktale, Graphische Grammatiken und prozedurale Modelle, Partikelsysteme, physikalisch basierte Modelle, dreidimensionale Volumendaten, und einige weitere. Wir wollen im Folgenden noch gekrümmte Oberflächen und Freiformflächen näher betrachten. Die anderen Datenstrukturen werden in fortführenden Lehrveranstaltungen behandelt.