

A. Filler, 2004

## Kurzanleitung:

# Transformationen, Boolesche Operationen, Prismen und Rotationskörper in POV-Ray<sup>1</sup>

## Transformationen

Werden die folgenden Befehle vor der letzten Klammer eines Objektes eingefügt, so wird das entsprechende Objekt verschoben, gedreht sowie gestreckt bzw. gestaucht:

```
translate <x,y,z> , rotate < $\varphi_x,\varphi_y,\varphi_z$ > , scale <x,y,z>.
```

### Beispiele:

`translate <1,2,3>` verschiebt ein Objekt um 1 Einheit in  $x$ -, 2 Einheiten in  $y$ - und 3 Einheiten in  $z$ -Richtung;

`rotate <90,15,30>` dreht ein Objekt  $90^\circ$  um die  $x$ -Achse,  $15^\circ$  um die  $y$ -Achse und  $30^\circ$  um die  $z$ -Achse;

`scale <2,3,0.5>` vergrößert ein Objekt auf die zweifache  $x$ - und die dreifache  $y$ -Ausdehnung und staucht es auf die halbe  $z$ -Ausdehnung zusammen.

Diese Transformationen können Sie in beliebiger Kombination und Reihenfolge in die Beschreibungen von Objekten einfügen. Beachten Sie aber, dass das Ergebnis bei der Nacheinanderausführung mehrerer Transformationen von der Reihenfolge abhängt.

## Boolesche Operationen: Vereinigungen, Schnitte, Differenzen

### Vereinigung: union

Durch Vereinigungen können komplexere Gebilde erzeugt werden. Transformationen und Texturen müssen dann nicht für jedes Teilobjekt beschrieben werden, sondern nur einmal für das gesamte Vereinigungsobjekt.

### Beispiel:<sup>2</sup>

```
union { cylinder {<-2,0,0>, <2,0,0>, 0.1}
  cylinder {<0,-2,0>, <0,2,0>, 0.1}
  torus {2,0.1}
  rotate <-40,-30,40>
  texture { silbergrau } }
```



### Erläuterungen:

`union { ... }` Die Vereinigung enthält beliebige Objekte, die dann als ein Ganzes behandelt werden. Die Eigenschaften des Gesamtobjektes werden am Ende – nach der Definition der Teilobjekte – angegeben.

`rotate <-40,-30,40>` Die Drehung bezieht sich auf alle Teilobjekte, die zur Vereinigungsmenge gehören.

`texture { ... }` Auch die Textur wird allen Teilobjekten zugewiesen.

<sup>1</sup>In dieses Anleitungsmaterial habe ich (in modifizierter Form) Beiträge von F. RIEPER einbezogen, der die Materialien für ein Unterrichtsprojekt anfertigte.

<sup>2</sup>Alle in dieser Anleitung gegebenen Beispiele können durch Einsetzen der Anweisungen in die Vorlagedatei `vorlage.pov` direkt gerendert werden.

### Schnittmenge: intersection

Wie in der Mengenlehre lassen sich mit POV-Ray Schnittmengen bilden. So entsteht aus zwei gleich großen Kugeln, die sich schneiden, eine Linse:

```
intersection { sphere {<-1,0,0>,1.5}
                sphere {<1,0,0>,1.5}
                rotate <0,-10,0>
                texture { silbergrau } }
```

**Erläuterung:** Durch den Befehl `intersection` entsteht der Durchschnitt der Teilobjekte, für zwei versetzte Kugeln also eine Linse.

**Aufgabe:** Bilden Sie den Durchschnitt aus einem Würfel und einer Kugel derart, dass ein „Würfel mit abgerundeten Ecken“ entsteht.

### Differenz (heraus schneiden): difference

```
difference { intersection { sphere {<-1,0,0>,1.5}
                          sphere {<1,0,0>,1.5} }
            cylinder {<-3,0,0>, <3,0,0>, 0.5}
            rotate <0,-78,30>
            translate <0.2,0.1,0>
            texture { blau_matt } }
```

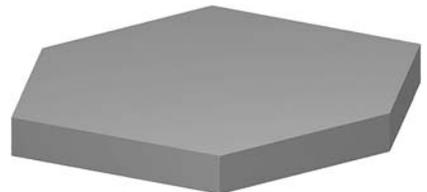
**Erläuterung:** Der Befehl `difference` schneidet aus der Linse (wie oben als Durchschnittsmenge zweier Kugeln erzeugt) einen Zylinder heraus. Die *Reihenfolge*, in der die Objekte genannt werden, ist hierbei wichtig.

**Aufgabe:** Schneiden Sie aus einem Würfel eine Kugel so heraus, dass ein Hohlwürfel entsteht.

## Prismen und Pyramiden

Durch Festlegung der Eckpunktkoordinaten der Grundfläche in der  $x$ - $z$ -Ebene lassen sich in POV-Ray Prismen und Pyramiden erzeugen.

```
prism { linear_sweep linear_spline
        0, // Höhe der Grundfläche
        1, // Höhe der Deckfläche
        7, // Zahl der Eckpunkte + 1
        <3,5>, <-3,5>, <-5,0>, <-3,-5>,
        <3, -5>, <5,0>, <3,5>
        texture { silbergrau } }
```



### Erläuterungen:

`linear_sweep` Festlegung eines geraden Prismas; `conic_sweep` erzeugt eine Pyramide.

`linear_spline` Legt fest, dass die Grundfläche des Prismas geradlinig begrenzt ist. Die Grundfläche kann auch von Kurven (Splines) begrenzt sein (siehe unten).

`<3,5> ... <3,5>` Die Koordinatenpaare legen die Grundfläche des Prismas in der  $x$ - $z$ -Ebene fest. Anfangs- und Endpunkt müssen übereinstimmen. Die Zahl der anzugebenden Punkte ist daher um 1 höher als die der tatsächlichen Eckpunkte.

**Hinweis:** Zur Änderung der Lage eines Prismas müssen die Befehle `rotate` und `translate` verwendet werden.

## Prismen mit Grundflächen, die von gekrümmten Kurven begrenzt sind

Die Erstellung eines Prismas mit einer Grundfläche, die durch eine „glatte“ Kurve (kubische Splinekurve) begrenzt wird, ist oft interessanter als die Erzeugung normaler Prismen.



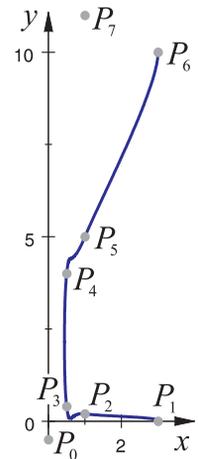
```
prism { cubic_spline, 0, 0.2           // Höhen der Grund- und Deckfläche
        17,                             // Gesamtzahl der Punkte
        <-2.5,0>                          // Kontrollpunkt
        <-2,0>, <-1.8,-0.5>, <-1,-1>, <0,-1>, // Punkte, durch welche die
        <1,-0.8>, <2,-0.5>, <2.5,-0.2>, <3,0>, // Kurve verläuft; der erste
        <2.5,0.2>, <2,0.5>, <1,0.8>, <0,1>, // und letzte Punkt müssen
        <-1,1>, <-1.8,0.5>, <-2,0>       // identisch sein
        <-2.5,0>                          // Kontrollpunkt
        texture{gruen_matt}                }
```

Die beiden Kontrollpunkte bestimmen die Richtungen, in welche die Begrenzungskurve der Grundfläche im ersten ihrer Punkte (der mit dem letzten Punkt übereinstimmt) weist.

## Rotationskörper

Durch Körper, die von Rotationsflächen (**sor** – Surface of Revolution) begrenzt werden, lassen sich viele reale Gegenstände modellieren. Zur Erzeugung eines Rotationskörpers müssen Koordinaten von Punkten eingegeben werden, durch die POV-Ray eine glatte Kurve (kubische Splinekurve) legt und diese um die  $y$ -Achse rotiert.

```
sor { 8, // Zahl der Stütz- + Kontrollpunkte
      <0,-0.5>, // Kontrollpunkt
      <3,0>, <1,0.2>, <0.5,0.4>, // Punkte der er-
      <0.5,4>, <1,5>, <3,10>, // zeugenden Kurve
      <1,11> // Kontrollpunkt
      scale <1,1.3,1>
      texture { silbergrau } }
```



### Erläuterungen und Hinweise:

- POV-Ray verwendet standardmäßig die  $y$ -Achse als Rotationsachse. Die 2D-Koordinaten der Punkte  $P_0$  bis  $P_n$  legen die erzeugende Kurve in der  $x$ - $y$ -Ebene fest.
- Der erste und der letzte Punkt (in der Skizze  $P_0$  und  $P_7$ ) sind Kontrollpunkte, die nicht auf der erzeugenden Kurve liegen, aber Einfluss auf den Kurvenverlauf (Richtung der Tangenten) in  $P_1$  bzw.  $P_6$  haben. Wird ein Rotationskörper mit `sor { 8, ...` erzeugt, so müssen 6 Kurvenpunkte und 2 Kontrollpunkte festgelegt werden.
- `scale <1,1.3,1>` streckt den Rotationskörper in  $y$ -Richtung um den Faktor 1,3.
- Um die Lage eines Rotationskörpers zu verändern, müssen die Befehle `rotate` und `translate` verwendet werden.
- Der `sor`-Befehl erzeugt „massive“ Rotationskörper. Um Rotationsflächen zu erstellen, können Sie den `lathe`-Befehl verwenden (siehe POV-Ray-Hilfe, Stichwort „Lathe“).