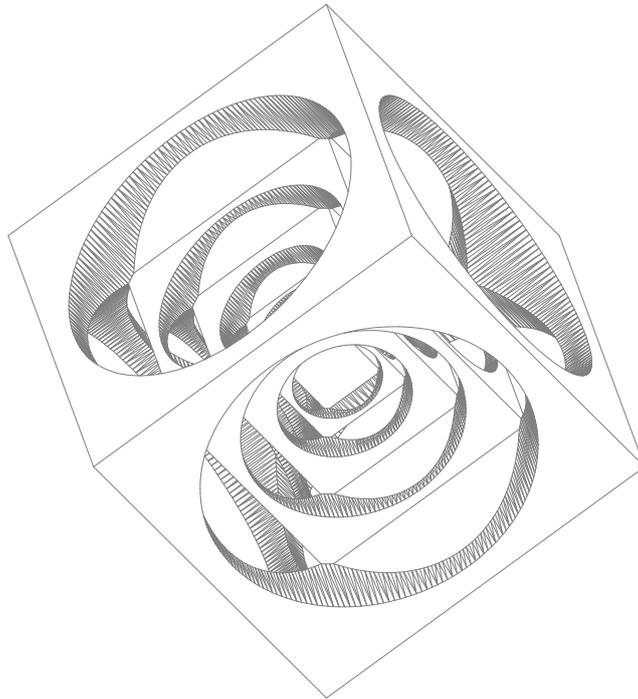


POV-Ray: Ein interessantes CSG-Objekt

Stefan Lagotzki

20. Januar 2002

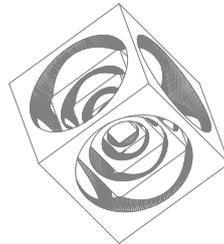


Im Rahmen eines kurzen Vortrags wird die Konstruktion des Modells mit POV-Ray beschrieben. In einer geplanten Fortsetzung wird auf die Verarbeitung der entstandenen Grafiken mit ImageMagick eingegangen.

POV-Ray ist ein frei verfügbares Raytracing-Programm, ImageMagick ist eine frei verfügbare Kollektion von Grafikwerkzeugen. Beide Programme liegen einigen Linux-Distributionen bei und sind auch im Quellcode verfügbar.

Inhaltsverzeichnis

1	Das Modell	3
1.1	Querschnitt der Zylinder	3
1.2	Konstruktion der Zylinder	4
1.3	Anordnung der Zylinder	5
1.4	Das fertige Objekt	6
2	Raytracing des Modells	7
2.1	Materialzuweisung	7
2.2	Ablegen von Optionen in einer INI-Datei	9
2.3	Die vollständige POV-Datei	10



Abbildungsverzeichnis

1	Querschnitt der Zylinder	3
2	Zylinder in y-Richtung	5
3	Zylinder in x- und y-Richtung	6
4	Drehung des Modells 45° um die y-Achse	8

1 Das Modell

Der Original-Würfel ist ein bekanntes Werkstück aus der CNC-Ausbildung, das aus Aluminium, Holz und anderen Materialien gefertigt werden kann. Der Würfel wird liegend konstruiert und kann dann noch gedreht werden. Sein Schwerpunkt liegt im Punkt (0,0,0) und er besitzt eine Kantenlänge von 70mm.

Der Würfel wird von allen sechs Seiten jeweils senkrecht mit vier unterschiedlichen Tiefen und Durchmessern gefräst. Dazu werden in POV-Ray konzentrische Zylinder verwendet, die man nur einmal auf der Oberseite des Würfels konstruieren muss. Die notwendigen Festkörper für die anderen fünf Flächen des Würfels kann man danach durch räumliches Drehen aus der ersten Gruppe von Zylindern erzeugen.

1.1 Querschnitt der Zylinder

Der Querschnitt durch die vier Zylinder dient uns als Vorlage für die Konstruktion. Die Zylinder liegen auf der y-Achse des linkshändigen Koordinatensystems. Sie sind also senkrecht ausgerichtet.

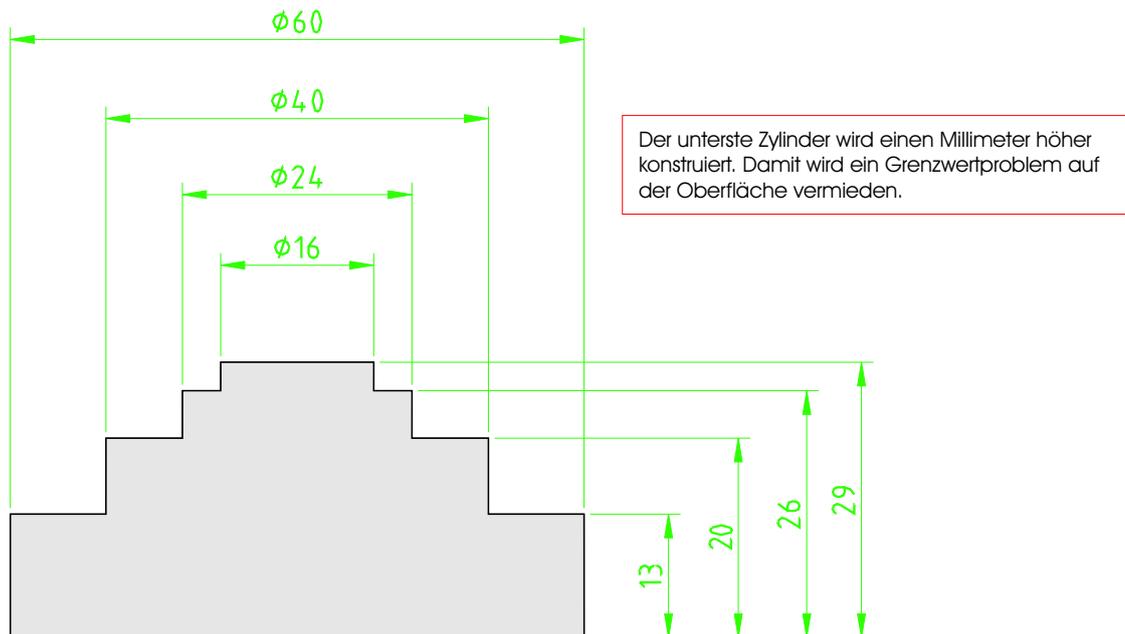


Abbildung 1: Querschnitt der Zylinder

Der Zylinder gehört zu den Grundkörpern in POV-Ray. Ein Zylinder wird durch die beiden Zentrumsunkte auf der Boden- und Deckfläche und durch seinen Radius beschrieben. Mit Hilfe von Zylindern kann man z.B. Bohrungen, Rohre, Gläser und in Kombination mit Kugeln auch *Blobs* herstellen.

1.2 Konstruktion der Zylinder

Im folgenden wird der Quelltext für die Konstruktion der ersten vier (konzentrischen) Zylinder erläutert. Diese vier Zylinder werden zum Objekt `fc1` vereinigt.

```
1  #declare fc1 = union {
2      cylinder {
3          <0, 36, 0>,
4          <0, 22, 0>,
5          30
6      }
7      cylinder {
8          <0, 36, 0>,
9          <0, 15, 0>,
10         20
11     }
12     cylinder {
13         <0, 36, 0>,
14         <0, 9, 0>,
15         12
16     }
17     cylinder {
18         <0, 36, 0>,
19         <0, 6, 0>,
20         8
21     }
22 }
```

1-22 Das Objekt `fc1` wird als `union` deklariert. Es soll später in der Konstruktion mehrfach verwendet werden.

2-7 innerhalb dieser Struktur wird hier der erste Zylinder konstruiert, er besitzt die Endpunkte $(0,36,0)$ und $(0,22,0)$ sowie den Radius 30mm.

8-21 hier werden die weiteren Zylinder deklariert.

1.3 Anordnung der Zylinder

Diese vier Zylinder werden nun durch 3D-Drehen so angeordnet, dass sie am Ende insgesamt sechs Mal vorhanden sind. Im ersten Schritt wird das Objekt fc2 deklariert:

```
1 #declare fc2 = union {  
2   object {fc1}  
3   object {fc1 rotate x*180}  
4 }
```

1-4 das neue Objekt fc2 ist wieder eine Vereinigung...

2 ... die aus dem Objekt fc1 ...

3 ... und dem 180° um die x-Achse gedrehten Objekt fc1 besteht.

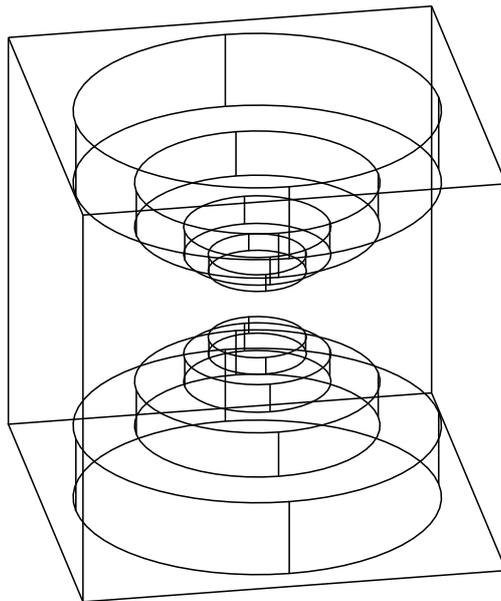


Abbildung 2: Zylinder in y-Richtung

Dieses Objekt benötigen wir nun noch zwei Mal. Einmal wird es 90° um die z-Achse und einmal 90° um die x-Achse gedreht. Das entstehende Objekt fca finden Sie im nächsten Quelltext.

```
1 #declare fca = union {  
2     object {fc2}  
3     object {fc2 rotate x*90}  
4     object {fc2 rotate z*90}  
5 }
```

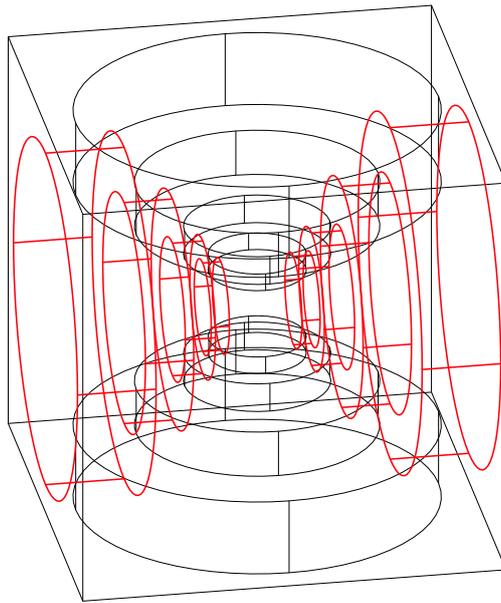


Abbildung 3: Zylinder in x- und y-Richtung

1.4 Das fertige Objekt

Nachdem wir nun alle Festkörper definiert haben, die von dem Würfel subtrahiert werden sollen, können wir das Objekt fertigstellen. Es wird die Differenz aus einem skalierten Würfel und dem Objekt `fca` gebildet. Die `UnitBox` ist eine Art „Einheitswürfel“.

```
1 difference{  
2     object {UnitBox scale 35}  
3     object {fca}  
4 }
```

2 Raytracing des Modells

2.1 Materialzuweisung

In diesem Abschnitt wird das Objekt nun zum Rendern vorbereitet. Zuerst erfolgt eine Materialzuweisung:

```
1  difference{
2      object {UnitBox scale 35}
3      object {fca}
4      texture { pigment { P_Chrome5 }
5                  finish { ambient 0.50 }
6      }
7      rotate <0 0 0>
8  }
```

Das Modell wird nun gespeichert, POV-Ray wird aufgerufen, und wir erhalten das erste Bild. Der Würfel ist hier nicht gedreht, so dass man genau auf die Frontseite (mit dem Zielpunkt 0,0,0) schaut. Das wird aber gleich noch geändert. Vorher wird noch der Kopf der Datei dokumentiert:

```
1  #include "colors.inc"
2  #include "shapes.inc"
3  #include "textures.inc"
4  #include "metals.inc"
5  #include "glass.inc"
6
7  // zum Drucken:
8  background {color White}
9
10 camera {
11     location <0, 0, -140.0>
12     direction <0, 0, 1>
13     up <0, 1, 0>
14     right <4/3, 0, 0>
15     angle 60
16     look_at <0, 0, 0>
17 }
18
19 light_source {< -50, 0, -50> color LightGray shadowless}
20 light_source {< 50, 0, -50> color LightGray shadowless}
```

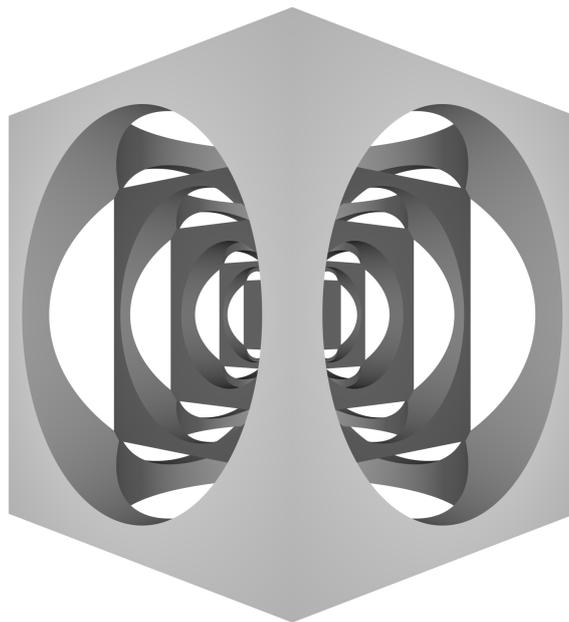
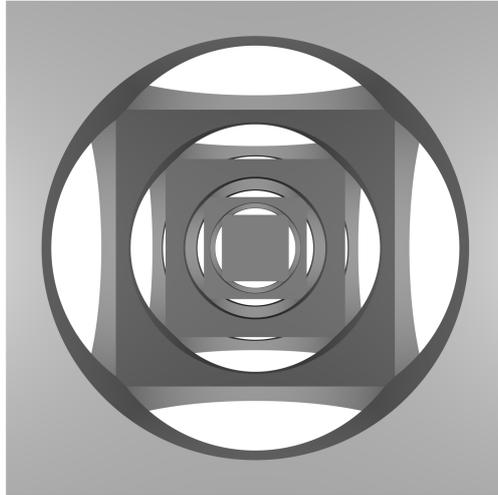


Abbildung 4: Drehung des Modells 45° um die y-Achse

2.2 Ablegen von Optionen in einer INI-Datei

Die Einstellungen zum Raytracing kann man natürlich auf der Kommandozeile übergeben. Aber es gibt einen (meiner Meinung nach) bequemeren Weg. Dazu schreibt man sich eine INI-Datei. Diese INI-Dateien haben den Vorteil, dass man damit den Prozess des Raytracings steuern kann (und außerdem kann man darin bestimmte Optionen dauerhaft ablegen).

```
1  Antialias=on
2  Antialias_Threshold=0.03
3  Antialias_Depth=9
4  Quality=9
5  Jitter=on
6  Jitter_amount=0.05
7
8  Width=2400
9  Height=1800
10
11 Output_File_Type=T
12
13 Pause_when_Done=off
14
15 Input_File_Name=csg_wuerfel_experimente.pov
16
17 Initial_Frame=0
18 Final_Frame=0
19 Initial_Clock=0
20 Final_Clock=0
```

In der INI-Datei werden also folgende Angaben gemacht:

- zur Qualität des Raytracing und des Antialiasing
- zur POV-Quelldatei
- zu den Maßen und zum Typ des späteren Bildes

Mit den letzten vier Zeilen kann man eine Animationssequenz programmieren. Dabei wird zuerst die Anzahl der Frames angegeben und dann der Wert einer Variablen *clock* initialisiert, deren Wert mit jedem neuen Frame geändert wird. Wenn man das Modell testet, dann sind diese Werte noch auf Null gesetzt und es wird nur ein Bild berechnet.

2.3 Die vollständige POV-Datei

```
1 #include "colors.inc"
2 #include "shapes.inc"
3 #include "textures.inc"
4 #include "metals.inc"
5 #include "glass.inc"
6 background {color White}
7
8 camera {
9     location <0, 0, -140.0>
10    direction <0, 0, 1>
11    up <0, 1, 0>
12    right <4/3, 0, 0>
13    angle 60
14    look_at <0, 0, 0>
15 }
16
17 light_source {< -50, 0, -50> color LightGray shadowless}
18 light_source {< 50, 0, -50> color LightGray shadowless}
19
20 #declare fc1 = union {
21
22     cylinder {
23         <0, 36, 0>,
24         <0, 22, 0>,
25         30
26     }
27
28     cylinder {
29         <0, 36, 0>,
30         <0, 15, 0>,
31         20
32     }
33
34     cylinder {
35         <0, 36, 0>,
36         <0, 9, 0>,
37         12
38     }
```

```
39
40     cylinder {
41         <0, 36, 0>,
42         <0, 6, 0>,
43         8
44     }
45 }
46
47 #declare fc2 = union {
48     object {fc1}
49     object {fc1 rotate x*180}
50 }
51
52 #declare fca = union {
53     object {fc2}
54     object {fc2 rotate x*90}
55     object {fc2 rotate z*90}
56 }
57
58 difference{
59     object {UnitBox scale 35}
60     object {fca}
61     texture { pigment { P_Chrome5 }
62                finish { ambient 0.250 }
63     }
64     rotate <0 clock 0>
65 }
```

Mit folgenden Einstellungen in der INI-Datei kann man eine Sequenz von elf Bildern erzeugen und das Modell um die y-Achse drehen. Dabei hat das fünfte Bild den Winkel 0° .

```
1 Initial_Frame=0
2 Final_Frame=11
3
4 Initial_Clock=-45
5 Final_Clock=45
```