

コンピュータグラフィックス (2023 年度版)

本実験テーマでは、コンピュータグラフィックス (CG) による画像生成を扱う。画像生成には、最も標準的な手法であるレイトレーシング法を用いる。ただし実験の時間が限られているため、レイトレーシング法を搭載したフリーソフトウェアとして有名な POV-Ray (Persistence of Vision Ray Tracing) バージョン 3.7.0 を用いて、仮想の物体や光源、カメラを設定し、シーンを表現するスクリプトを書く。このようなソフトウェアを利用すれば、OpenGL 等の CG 専用ライブラリを参照してプログラムを組まなくても、手軽に CG 画像をつくることができる。

本稿ではまず、POV-Ray の使い方を説明する前に、必要となる CG の基本的な考え方について解説する。その後で、例題に基づいて POV-Ray の使い方を説明する。POV-Ray の完全な解説は、ソフトウェアのヘルプメニューから得られる (ただし英文)。なお、以前のバージョン 3.5 向けに大分大学による優れた和訳もあるので、必要に応じて閲覧してほしい：

<http://www.arch.oita-u.ac.jp/a-kse/povip/povip/pov35ref.html>.

POV-Ray の最新版や関連情報を入手できる公式サイトは

<http://www.povray.org/>

であるが、インターネットのサーチエンジンを用いて検索すれば、他にも有用なサイトを多数知ることができる。例えば、Mac 版の非公式コードは以下からダウンロード可能である：

http://megapov.inetart.net/povrayunofficial_mac/

最後に、POV-Ray を用いた CG 入門書を 3 冊紹介するので、適宜参考にしてほしい。

- 1) 鈴木 広隆, 倉田 和夫, 佐藤 尚 : POV-Ray による 3 次元 CG 制作 -モデリングからアニメーションまで-, 画像情報教育振興協会, 2008.3
- 2) 齊藤 剛, 年森 敦子, 田代 裕子 : 3DCG をはじめよう POV-Ray 入門, オーム社, 2009.7
- 3) 松下 孝太郎, 山本 光, 柳川 和徳, 鈴木 一史, 星 和磨, 羽入 敏樹 : POV-Ray で学ぶはじめての 3DCG 制作, 講談社, 2017.2

1. コンピュータグラフィックス概説

コンピュータグラフィックス (以降 CG) は、コンピュータの内部に仮想の物体を設定し、それに仮想の光源から出る光を当てて、仮想のカメラで撮影したときに得られる写真を、計算によってデジタル画像として作成する技術である。実際に画像をつくる計算方法にはさまざまなものがあるが、そのなかでも、レイトレーシング法 (光線追跡法、略してレイトレともいう) は、アルゴリズムの簡潔さ、得られる画像の多彩さによって、現在最も広く使われている画像生成アルゴリズムの一つである。

1.1 色の表現

CG ではさまざまな場面で、色を数値として指定することが必要になる。色の指定は通常、赤(R)・緑(G)・青(B)の 3 色、すなわち光の三原色に分解し、それぞれの原色の強さを 0 から 1 までの実数値で指定することによって行なう。ただしコンピュータの内部では 0 から 255 までの整数値に変換して保持するので、1 つの色の指定に 3 バイトが必要である。図 1 に、光の三原色の基本的な組合せを示す。

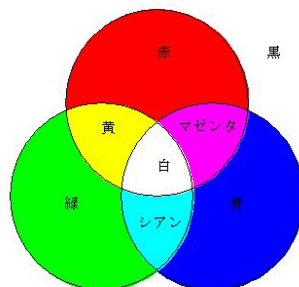


図 1 光の三原色とその組合せ

1.2 モデリングとレンダリング

CG による画像作成は次の 2 ステップで行なう。

- モデリング
- レンダリング

モデリングは、架空の 3 次元世界を造る作業であり、レンダリングはその架空の世界の見え方を調べ、実際に画像を生成する作業である。本実験では、レンダリングを POV-Ray に任せ、モデリングだけを行なえばよい。

モデリングでは、架空の 3 次元世界を構成する物体、光源、カメラについて、次のような性質を指定する。

- 物体：形、位置、大きさ、色、表面の状態、反射率、屈折率
- 光源：位置、色、大きさ、形、配光特性
- カメラ：位置、ピントを合わせる位置

なお、レンダリングするときには、生成する画像のサイズ（画素数）とアンチエイリアシングの有無を設定する。アンチエイリアシングは画像に現れるギザギザのパターンを軽減する作業で、これを行なうと画質は向上するが、計算時間が長くなる。

1.3 レイトレーシング法

レイトレーシング法では、図 2 に示すように、カメラとその焦点位置とを結ぶ直線に直角な平面を仮定し、その上に画像を作る。

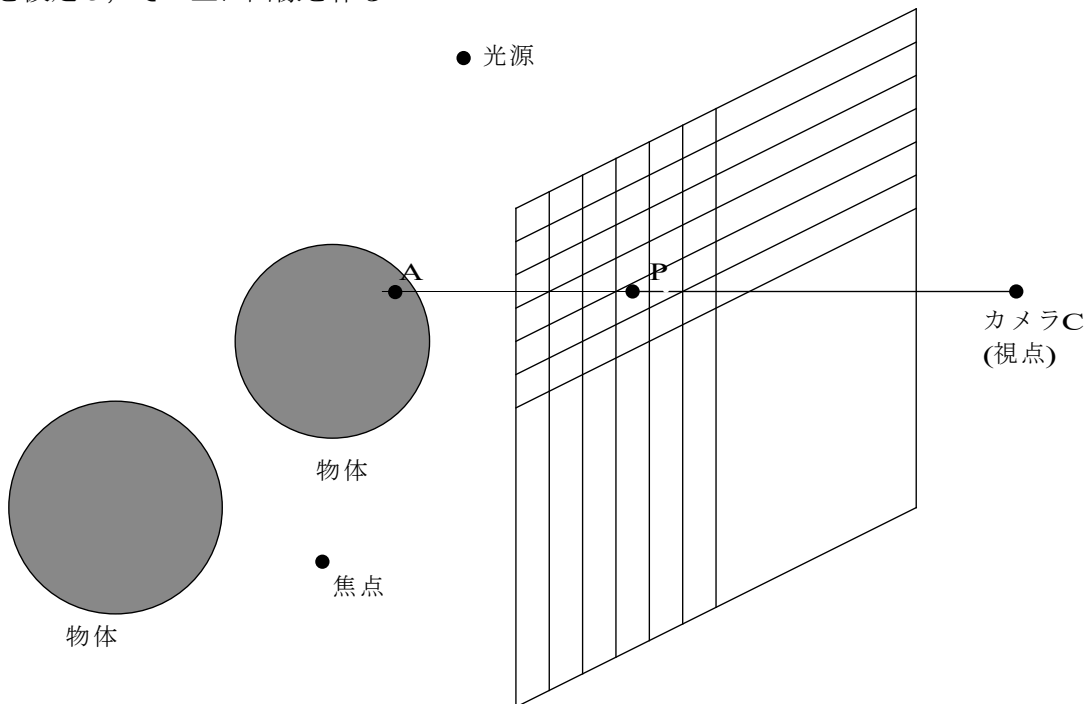


図 2 レイトレーシング法の原理

各画素に塗る色は、次のようにして決める。ある画素 P がカメラ C からどのような色に見えるかは、画素 P からカメラ C に到達する色によって決まる。そこで C から P を通る直線を延長し、最初に交わる物体 A の色を画素 P に塗る。どの物体とも交わらないときは、あらかじめ定めた背景色を塗る。

このため、カメラと画像上の各画素とを結ぶ直線を定め、各直線とすべての物体との交点計算を行う必要がある。これが、レイトレーシング法が遅いといわれる理由であるが、最近、コンピュータの高速化や、ソフトウェア上の工夫（空間の分割による交点計算対象物体の限定など）、特殊なハードウェアの使用（並列コンピュータなど）によって、レイトレーシング法の利用範囲が大きく広がっている。

1.4 シェーディング

前項に示した原理のままでは、全く立体感のない絵しか得られない。光源の位置を考慮に入れず、赤い物体と最初に交わったときには、すべて同じ赤を塗ってしまうからである。しかし実際には、全体が同じ赤色をした物体であっても、光源の位置によって、明るく見えるところや暗く見えるところが生じる。これを計算によって再現する作業をシェーディングとよぶ。

シェーディングでは、普通、カメラに入る光を次の3種類の光の和であると考ええる。

- 環境光 (ambient)
- 拡散反射光 (diffuse)
- 鏡面反射光 (specular. POV-Ray では phong の語も用いる)

通常環境では、光源からの直射光があたらない場所であっても、暗黒となることはない。壁や天井、床などからの反射光、空中のチリによる散乱光が到達するからである。このような現象を近似するため、物体上の（直射光が当たる場所も当たらない場所も）あらゆる場所に一定の明るさを与えてしまう。このような光を環境光とよぶ。

一方、直射光があたった場所について、その光があらゆる方向に均一に散乱反射をするものと仮定する光を拡散反射光とよぶ。拡散反射光の明るさは、ランバートの法則により、入射角 θ の余弦に比例する（図 3a）。

拡散反射光はツヤのない表面を表わす光であるのに対して、鏡面反射光は物体の表面にツヤを与え、ハイライトをつける。鏡面反射光の明るさは、正反射方向とカメラ方向との角度差 α に基づいて、 $\cos^n \alpha$ に比例させるのが普通である（図 3b）。

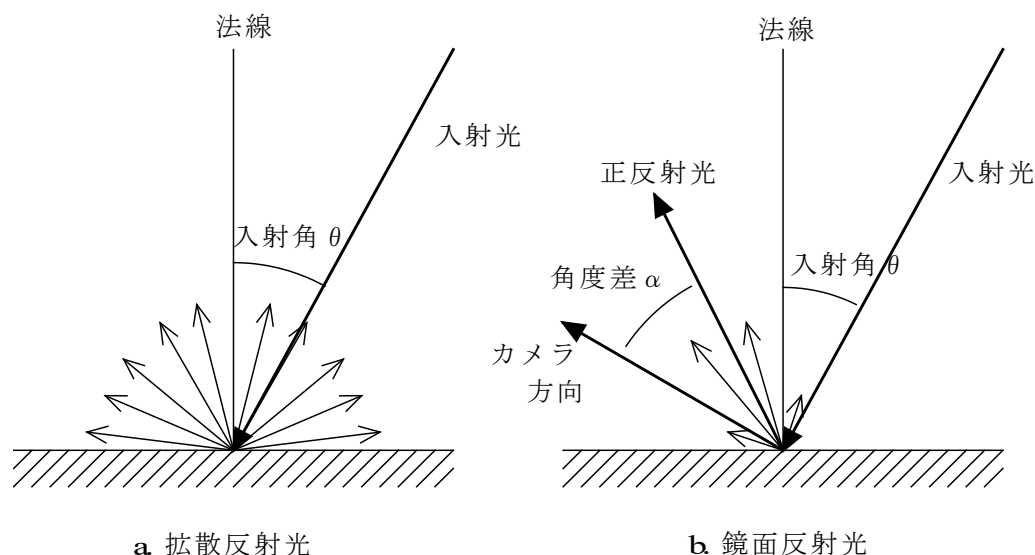


図3 拡散反射光と鏡面反射光

環境光に相当する部分をより厳密に計算する手法としてはラジオシティ法があり、照明設計などに用いられている（POV-Rayでも実行可）。また、ここに鏡面反射光の式はフォン(Bui-Tuong Phong)が提案したもので、簡単な計算ながら良い結果を与えるので広く使われている。ただし、特に金属に大きな入射角で入射する光では実測結果がこの式から大きく外れてしまうため、より厳密な計算方法も提案されている。

2. POV-Ray の使い方

POV-Ray はレイトレーシング法によって画像を作成するためのフリーソフトウェアで、Windows や UNIX マシン、Macintosh などさまざまなコンピュータ上で動作する版が配布されている。本節では Windows 上で実験用にセットアップされた POV-Ray ver. 3.7.0 を使う環境を前提として説明する。POV-Ray に与えるシーンファイルは、POV-Ray 組込みのテキストエディタにより作成する（他のテキストエディタも利用可）。シーンファイルには拡張子.pov をつける。

2.1 POV-Ray の起動とシーンデータの作成

プログラムメニューの中から [POV-Ray for Windows](#) を選択する。デスクトップ上に [POV-Ray](#) のアイコンがある場合には、それをダブルクリックしてもよい。

新しいシーンファイルを作るときには、[File](#) メニューの中の [New File](#) を選択するかツールバーの [New](#) ボタンをクリックすれば、画面上に新しいウインドウが開いて、[図 4](#) に示すようなシーンデータを入力できる状態になる。

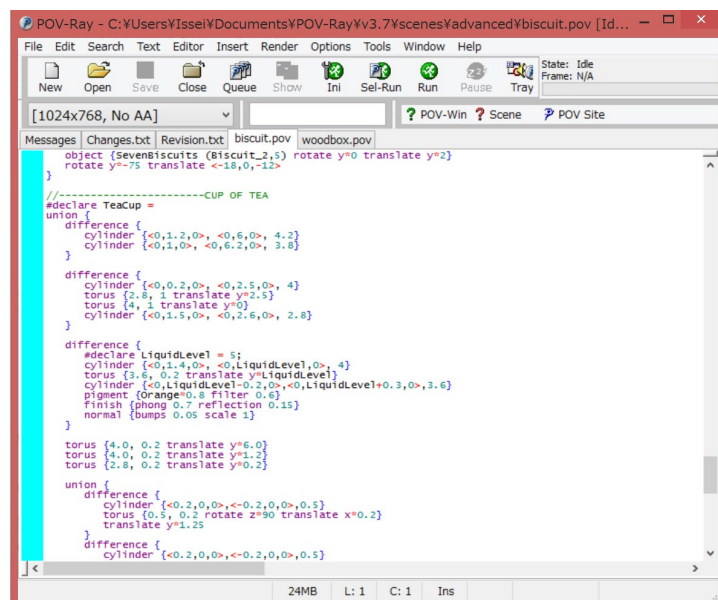


図 4 POV-Ray のエディタ基本画面

2.2 シーンデータの保存

作成したシーンを保存するときには、[File](#) メニュー中の [Save](#) やツールバーの [Save](#) ボタンを用いる。ここでファイルの名前は、例えば「学籍番号.pov」とし、実験の専用フォルダに保存する。名前を変えてシーンを保存したいときには、[File](#) メニュー中の [Save As ...](#) でファイル名を変えて保存すればよい。

2.3 画像の作成

シーンデータを入力後、ツールバーの [Run](#) アイコンをクリックするか、[Render](#) メニューの中の [Start Rendering](#) を選択すると、画像ウインドウが開いて、その中に画像が生成されると同時に、png 形式の画像ファイルも生成される。

シーンデータ中に何らかの誤りがあった場合には、画像が表示されず、ウインドウの下部にエラーメッセージ(英語)が表示される。このときには、シーンデータを修正してから、再度この手順を実行する。エラーメッセージは [Messages](#) タグをクリックしても読むことができる。

2.4 画像サイズの設定

シーンを作成しているときには、画像の大きさをなるべく小さくしておかないと、描画に時間がかかって効率が悪い。一方、シーン完成後に画像を印刷する場合には、大きな画像にしないと、貧弱で細部がよく見えない。

作成する画像の大きさや画質を切り替える手順を以下に示す。図 4 に示した基本画面の左上に **800×600, No AA** のような表示がある。これは「横 800 画素、縦 600 画素で、アンチエイリアシングを行わずに画像を作る」という指定である。この枠をクリックすると、指定可能な解像度が現れるので、適当なものをクリックすればよい。AA 0.3 がついている項目を選択すれば、アンチエイリアシングを行う設定に切り替わる。

2.5 画像の印刷

最終画像のレンダリングを **800×600, AA 0.3** などを指定して実行すると、シーンファイルを保存したフォルダに、同名の生成画像が png 形式で生成される（たとえば、test.pov から test.png が作られる）。そのファイルをペイントツールで開き、印刷の向きを横にすれば、A4 判にうまく収まるように印刷できる。なお、シーンファイル自身の印刷には File メニューの **Print ...** を用いる。

2.7 サンプルやマニュアルの利用

標準的なインストールにより、膨大な量の POV-Ray サンプルシーンファイルが利用可能になるので、適宜実行してみるとよい。

また、Help メニューの中の **HELP on POV-Ray** を選ぶと、マニュアルの目次部分が表示される。ハイライト部をクリックすれば、より詳細な情報が表示される。

3. POV-Ray によるシーン記述の例

POV-Ray でシーンを記述する際は、図 5 のような直交座標系を使用して座標を指定する。

- x 軸：画面中央を左から右に向かう
- y 軸：画面中央を下から上に向かう
- z 軸：画面の中心を手前から奥に向かう

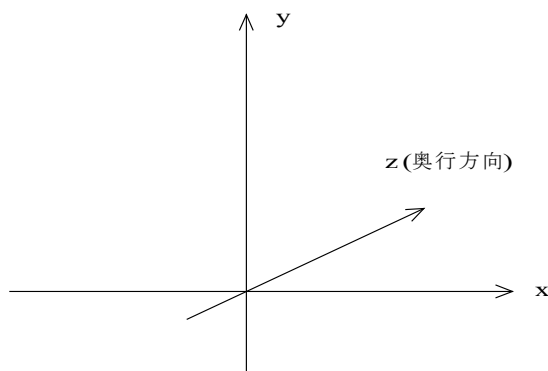


図 5 POV-Ray の座標系（左手座標系）

すなわち、数学で通常使う座標系とは違い、左手系を採用している。この方が、奥行き (z) 座標の大小関係とカメラからの距離の大小関係が一致して、シーンを考えやすいためである。

ここで、シーン記述の簡単な例を示す。左端の行番号は説明用であり、実際には入力しない。

例 1. 単純な物体

```
1: #include "colors.inc"
2: #include "textures.inc"
3:
4: camera { location < -4.0, 2, -3.0 >
5:         look_at < 0.0, 0.0, 0.0 >
6:         angle 60
7: }
8:
9: light_source { < 20, 30, -25 > color White }
10:
11: sphere { < 0, 0, 0 >, 1
12:         pigment { Green }
13:         finish { Shiny }
14: }
15:
16: box { < -2, -1.2, -2 >, < 2, -1, 2 >
17:     texture { Red_Marble }
18:     finish { ambient 0.2
19:             diffuse 0.8
20:             phong 1
21:     }
22: }
23:
24: background { color Gray50 }
```

各行の意味を説明する。1 行目では、さまざまな色の名前を定義したファイル `colors.inc` をインクルードしている。これによって、以後のシーン記述で `White`, `Green`, `Gray50` などの色名を使えるようになる。逆に、このファイルをインクルードしない場合には、赤、緑、青の各成分の強さを直接数値で指定しなければならない。

2 行目では、さまざまなテクスチャや表面仕上げの名前を定義したファイル `textures.inc` をインクルードしている。これによって、以後のシーン記述で `Red_Marble` などのテクスチャ名、`Shiny` などの表面仕上げ名を使えるようになる。

4 行目では、カメラの位置を設定している。5 行目は、そのカメラが向いている位置の指定である。また 6 行目はカメラの視野角を 60° に設定している。

9 行目では、光源の位置と色を指定している。

11 行目では、中心が原点(0,0,0)にある半径 1 の球を設定している。12 行目ではそれが緑色であることを、13 行目ではその表面がツヤツヤしていることを指定している。

16 行目では、対角線の両端の座標が(-2, -1.2, -2), (2, -1, 2)であるような、座標軸に平行な直方体を指定している。17 行目はその表面に赤大理石の模様があることの指定である。また、18 行目から 20 行目は、その表面のシェーディングの指定で、環境光を 2 割、拡散反射光を 8 割とし、鏡面反射光のパラメータの値を 1 としている。

24 行目は背景色の指定で、`Gray50` は中間的な明るさの灰色である。

このシーン定義から得られた画像を図 6 に示す。

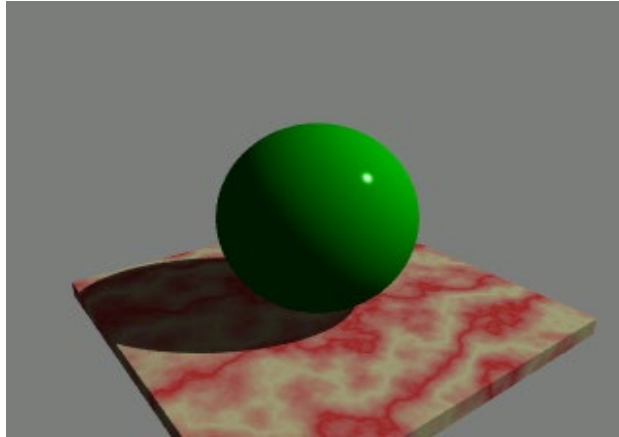


図 6 例 1 の結果画像

例 2. box.pov : CSG (物体同士の和, 積, 差演算)

```

1: // POV-Ray sample data file for CSG operation
2:
3: camera {
4:     location <0, 0, -8>
5:     direction <0, 0, 1.2071>
6:     look_at <0, 0, 0>
7: }
8:
9: intersection{
10: sphere { <0.0, 0.0, 0.0>, 2
11:     finish {
12:         ambient 0.2
13:         diffuse 0.8
14:         phong 1
15:     }
16:     pigment { color red 1 green 1 blue 0 }
17: }
18:
19: box { <-2.0, -0.7, -2.0>, <2.0, 0.7, 2.0>
20:     finish {
21:         ambient 0.2
22:         diffuse 0.8
23:     }
24:     pigment { color red 1 green 0 blue 1 }
25:
26:     rotate <-20, 30, 0>
27: }}
28:
29: light_source { <-10, 3, -20> color red 1 green 1 blue 1 }
30: background { color red 0.5 green 0.5 blue 0.5 }

```

1 行目は注釈である。注釈には `/* */` の形式の書き方もあり, それを使えば, 数行にまたがる注釈を書くことができる。

この例の 9 行目から 27 行目では, 球と直方体という基本的な物体 2 つを積集合という演算によって組み合わせることにより, やや複雑な物体を構成している. このような操作を CSG (Constructive Solid Geometry) とよび, 次の 3 種類が用意されている (図 7) .

- 和集合 : いくつかの物体の合成によって新しい物体を定義する.
 $\text{union} \{ \text{物体 1} \ \text{物体 2} \ \cdots \ \text{物体 } n \}$
- 積集合 : いくつかの物体の共通部分として新しい物体を定義する.
 $\text{intersection} \{ \text{物体 1} \ \text{物体 2} \ \cdots \ \text{物体 } n \}$
- 差集合 : ある物体から他の物体との共通部分を除去して新しい物体を定義する.
 $\text{difference} \{ \text{物体 1} \ \text{物体 2} \ \cdots \ \text{物体 } n \}$ によって, 物体 1 から他のすべての物体を除去した物体が作られる. 厚みがゼロの部分を残さないように注意すること.

これらの集合演算で作った物体をさらに集合演算で組み合わせることもできる. 組合せの過程は CSG 木というデータ構造として記述することができる.

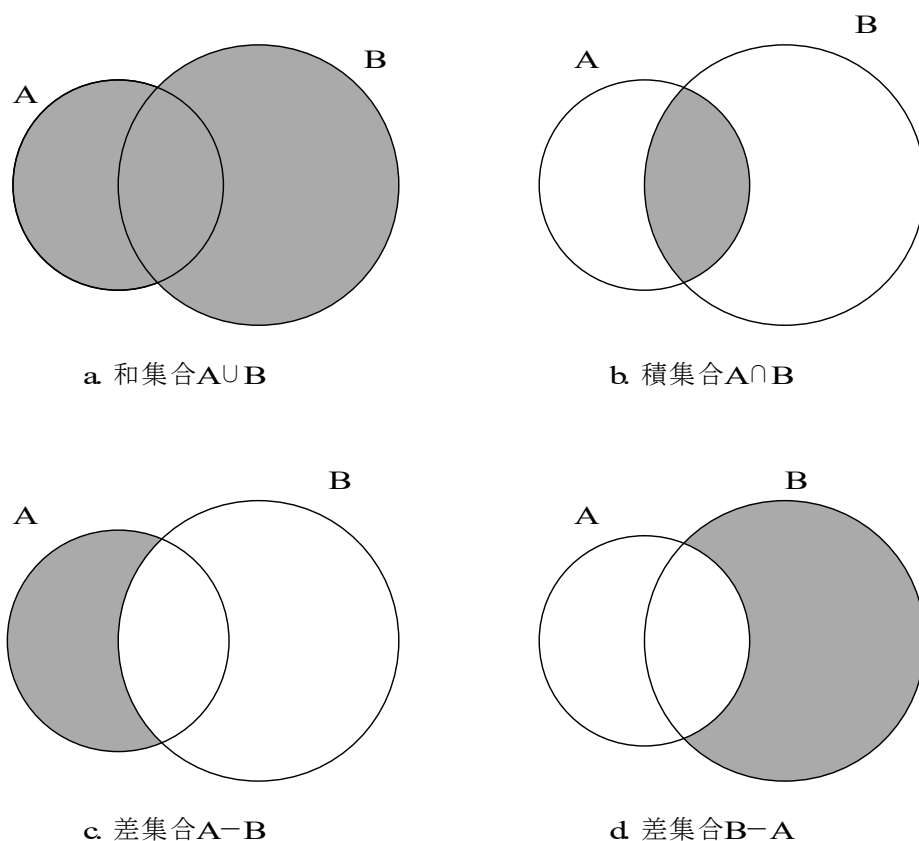


図 7 CSG

26 行目は, 物体の回転の指定である. x 軸回りに -20 度回転してから y 軸回りに 30 度回転する. 単純な回転であれば, $\text{rotate } 30 * y$ のような書き方をしてもよい.

このような座標変換の書き方を以下に示す.

- 平行移動 : $\text{translate} \langle 10, -20, 30 \rangle$ は, x 軸方向に 10, y 軸方向に -20 , z 軸方向に 30 移動する.
- 回転移動 : $\text{rotate} \langle 10, -20, 30 \rangle$ は, x 軸回りに 10 度回転してから, y 軸回りに -20 度回転し, さらに z 軸回りに 30 度回転する.
- スケーリング : $\text{scale } 5$ は全方向に 5 倍拡大する.

この例で得られた画像を図 8 に示す.

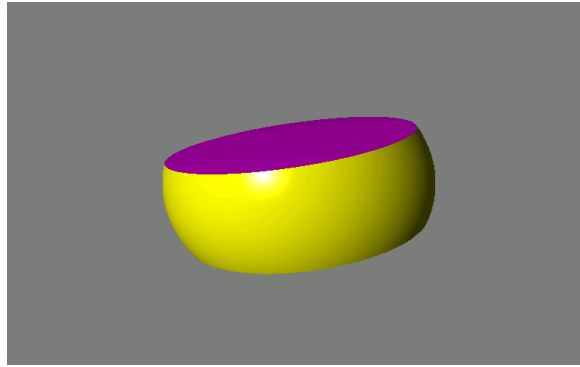


図 8 例 2 の結果画像

4. シーン記述の概要

本節では、POV-Ray におけるシーン記述の主要な形式を示す。詳細については Help メニュー等を参照すること。

4.1 位置の記述

[形式] `< x 座標, y 座標, z 座標 >`

[例] `< 3.14, -1.414, 365 >`

4.2 座標変換

以下の座標変換は、物体だけではなくテクスチャにも適用することができる。

● 移動

[形式] `translate < x 方向の移動量, y 方向の移動量, z 方向の移動量 >`

[例] `translate < 0.0, 2.0, 0.0 >`

● 回転

[形式] `rotate < x 軸回りの回転角, y 軸回りの回転角, z 軸回りの回転角 >`

`x 軸回り, y 軸回り, z 軸回りの順に回転する。その他の順序, 例えば, y 軸, z 軸, x 軸の順に回転したい場合には例 b のような書き方を用いる。`

[例 a] `rotate < 10, 20, -5 >`

[例 b] `rotate < 0, 20, -5 >`

`rotate < 10, 0, 0 >`

● スケーリング

[形式 1] `scale 倍率`

[例] `scale 0.5`

[形式 2] `scale < x 方向の倍率, y 方向の倍率, z 方向の倍率 >`

[例] `scale < 0.5, 1.0, 1.0 >`

4.3 色の記述

[形式] `color red 赤の強さ green 緑の強さ blue 青の強さ`
 ただし、各色の強さは 0 から 1 までの数値。5.1 項参照。

[例] `pigment { color red 1.0 green 0.5 blue 0.5 }`

4.4 カメラの記述

[形式] camera {
 location < 位置 >
 look_at < 焦点を合わせる位置 >
 angle 視野角
}

[例] camera {
 location < 3, 5, -10 >
 look_at < 0, 2, 1 >
 angle 25
}

4.5 光源の記述

[形式] light_source { < x 座標, y 座標, z 座標 > 色 }

[例] light_source { < 3, 5, -6 > color Gray50 }

点光源の他に、スポットライト、面光源も指定できる。詳しくはマニュアルを参照。

4.6 基本物体の記述

● 球

[形式] sphere { < 中心の x 座標, 中心の y 座標, 中心の z 座標 >, 半径 }

[例] sphere { < 0, 25, 0 >, 10
 pigment { Blue }
}

● 直方体

[形式] box { < 左下の頂点の座標 >, < 右上の頂点の座標 > }

[例] box { < 0, 0, 0 >, < 1, 1, 1 >
 texture { Red_Marble }
}

● 円柱

[形式] cylinder { < 底の円板の中心の座標 >, < ふたの円板の中心の座標 >, 半径 }
半径の直後に open と指定すると、両端の円板のないチューブになる。

[例 a] cylinder { < 0, 0, 0 >, < 0, 3, 0 >, 2
 pigment { Green }
}

[例 b] cylinder { < 0, 0, 0 >, < 0, 3, 0 >, 2
 open
 pigment { DMFWood4 scale 4 }
}

● 円錐

[形式] cone { < 底の円板の中心の座標 >, 底の半径,
 < ふたの円板の中心の座標 >, ふたの半径 }

[例] cone { < 0, 0, 0 >, 2, < 0, 3, 0 >, 0
 pigment { Orange }
}

- トーラス

[形式] `torus { 主半径, 副半径 }`

トーラスの中心は原点で, xz 平面上にあり, y 軸が穴を貫通している (図 9) .

[例] `torus { 5, 0.5 }`

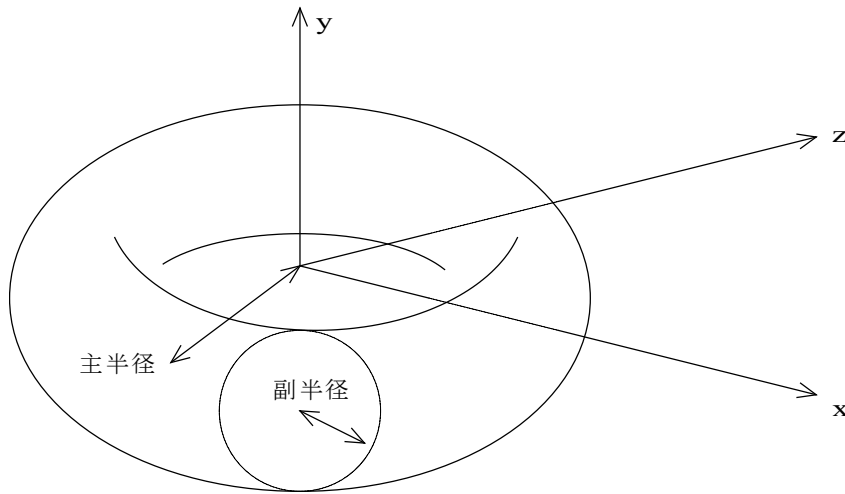


図 9 トーラス

- メタボール

等方的に噴射するガス球 (構成要素) を複数配置し, その密度の和が一定の値 (閾値) になる位置に面を描く方法

[形式] `blob {`

`threshold` しきい値

`component` 強さ, 半径, \langle 中心の座標 \rangle

`component` 強さ, 半径, \langle 中心の座標 \rangle // 必要な構成要素の数だけ

`component` 強さ, 半径, \langle 中心の座標 \rangle // 繰り返す

`}`

中心から減衰する強さの合計がちょうど閾値と一致する場所が面になる.

[例] `blob {`

`threshold` 0.6

`component` 1.0, 1.0, \langle 0.75, 0, 0 \rangle

`component` 1.0, 1.0, \langle -0.375, 0.64952, 0 \rangle

`component` 1.0, 1.0, \langle -0.375, -0.64952, 0 \rangle

`scale` 2

`pigment { Yellow }`

`}`

- 三角形

[形式] `triangle { \langle 頂点 1 の座標 \rangle , \langle 頂点 2 の座標 \rangle , \langle 頂点 3 の座標 \rangle }`

[例] `triangle { \langle 0, 0, 0 \rangle , \langle 1, 0, 0 \rangle , \langle 0, 1, 0 \rangle }`

`pigment { Green }`

`}`

- 平面

[形式] `plane { \langle 法線ベクトル \rangle , 原点からの距離 }`

`plane { \langle A, B, C \rangle , D }` は式 $Ax + By + Cz = D$ が表わす無限平面を定義する.

[例] `plane { \langle 0, 1, 0 \rangle , 2 }`

- 2 次曲面

[形式] quadric { < A, B, C >, < D, E, F >, < G, H, I >, J }

次の式が表わす曲面を定義する.

$$Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + J = 0$$

[例 a] 中心が <0,0,0>, 半径が 1 の球

```
quadric { < 1, 1, 1 >, < 0, 0, 0 >, < 0, 0, 0 >, -1
        pigment { Gold }
    }
```

[例 b] z 軸にそった無限に長い半径 1 の円柱

```
quadric { < 1, 1, 0 >, < 0, 0, 0 >, < 0, 0, 0 >, -1
        pigment { Aquamarine }
        finish { Shiny }
    }
```

[例 c] 原点を頂点とし, z 軸にそった無限に長い円錐

```
quadric { < 1, 1, -1 >, < 0, 0, 0 >, < 0, 0, 0 >, 0
        texture { Red_Marble }
    }
```

- 柱体 y 軸にそった n 角柱

[形式] prism { linear_sweep linear_spline

底面の y 座標, 上面の y 座標, n,

<x1, z1>, <x2, z2>, ... , <xn, zn> }

[例] 正三角柱

```
prism {
    linear_sweep linear_spline
    0, 1, 3,
    <-1/2, 0>, <0, sqrt(3)/2>, <1/2, 0>
    pigment { Gray50 }
}
```

- 文字列 xy 平面上に, 原点から +x 軸方向に描いた文字列

[形式] object { text { ttf "フォントファイル名", "文字列", 文字の厚さ, 0 } }

[例] object {

text { ttf "crystal.ttf", "Keio Univ.", 1, 0 }

texture { Red_Marble }

}

これら以外にも, いくつかの物体を定義することができる. また, ファイル shapes.inc, shapes2.inc, shapesq.inc に定義されているさまざまな物体も

object { 物体名 }

の形式で利用できる. 詳細は 5.2, 5.3, 5.4 の各項を参照のこと.

4.7 表面仕上げの記述

[形式 a] finish { 表面仕上げの名前 }

「表面仕上げの名前」は 5.6 節参照.

[例] finish { Shiny }

[形式 b] finish {

diffuse 拡散反射光の割合

```

    ambient 環境光の割合
    reflection 反射率
    phong    フォーンハイライトの強さ
    refraction 屈折の有無
    ior      屈折率
}

```

4.8 テクスチャの記述

- 物体色

[形式] pigment { 色 }

[例 a] pigment { Green }

[例 b] pigment { color red 0 green 0.5 blue 0 }

- 市松模様

[形式] pigment { checker color 色 1 color 色 2 }

- 六角形パターン

[形式] pigment { hexagon color 色 1 color 色 2 color 色 3 }

- 画像ファイルの貼付け

[形式] texture { pigment { image_map { gif "画像ファイル名" map_type 0 } } }

ここで、map_type 0 は平面へのマッピングを表し、画像が (0,0) から (1,1) までを単位として繰り返し貼り付けられる (z 方向へは金太郎飴のように無限に伸びる)。より大きな領域に 1 回だけ貼り付けたい場合は

```
texture { pigment { image_map { gif "sample.gif" map_type 0 once } } scale < 2, 3, 0 > }
```

のような書き方をする。map_type 1 は球面マッピング、map_type 2 は円柱マッピングである。

その他のテクスチャに関しては 5.5 項やマニュアルを参照のこと。

4.9 CSG の記述

- 和集合 (合体)

[形式] union { 物体 1 物体 2 ... 物体 n }

物体 1 \cup 物体 2 \cup ... \cup 物体 n を定義する。

[例] union {

```
sphere { < 0, 0.5, 0 > 1 pigment { Red } }
```

```
sphere { < 0, 0.0, 0 > 1 }
```

```
sphere { < 0, -0.5, 0 > 1 }
```

```
pigment { Blue }
```

```
finish { Shiny }
```

```
}
```

- 積集合 (共通部分)

[形式] intersection { 物体 1 物体 2 ... 物体 n }

物体 1 \cap 物体 2 \cap ... \cap 物体 n を定義する。

[例] intersection {

```
sphere { < -0.75, 0, 0 >, 1 }
```

```
sphere { < 0.75, 0, 0 >, 1 }
```

```
pigment { Yellow }
```

- 差集合

[形式] difference { 物体 1 物体 2 ... 物体 n }

物体 1 − 物体 2 − ... − 物体 n を定義する．厚さがゼロの部分を残さないように注意すること．

[例] difference {
 sphere { < -0.75, 0, 0 >, 1 }
 sphere { < 0.75, 0, -0.25 >, 1 }
 pigment { Yellow }
 }

5. Include ファイル

5.1 ファイル colors.inc で定義された色

| | red | green | blue |
|------------|----------|----------|----------|
| Gray05 | 0.05 | 0.05 | 0.05 |
| Gray10 | 0.10 | 0.10 | 0.10 |
| Gray15 | 0.15 | 0.15 | 0.15 |
| Gray20 | 0.20 | 0.20 | 0.20 |
| Gray25 | 0.25 | 0.25 | 0.25 |
| Gray30 | 0.30 | 0.30 | 0.30 |
| Gray35 | 0.35 | 0.35 | 0.35 |
| Gray40 | 0.40 | 0.40 | 0.40 |
| Gray45 | 0.45 | 0.45 | 0.45 |
| Gray50 | 0.50 | 0.50 | 0.50 |
| Gray55 | 0.55 | 0.55 | 0.55 |
| Gray60 | 0.60 | 0.60 | 0.60 |
| Gray65 | 0.65 | 0.65 | 0.65 |
| Gray70 | 0.70 | 0.70 | 0.70 |
| Gray75 | 0.75 | 0.75 | 0.75 |
| Gray80 | 0.80 | 0.80 | 0.80 |
| Gray85 | 0.85 | 0.85 | 0.85 |
| Gray90 | 0.90 | 0.90 | 0.90 |
| Gray95 | 0.95 | 0.95 | 0.95 |
| | | | |
| DimGray | 0.329412 | 0.329412 | 0.329412 |
| Gray | 0.752941 | 0.752941 | 0.752941 |
| LightGray | 0.658824 | 0.658824 | 0.658824 |
| VLightGrey | 0.80 | 0.80 | 0.80 |
| | | | |
| White | 1.0 | 1.0 | 1.0 |
| Red | 1.0 | 0.0 | 0.0 |
| Green | 0.0 | 1.0 | 0.0 |
| Blue | 0.0 | 0.0 | 1.0 |
| Yellow | 1.0 | 1.0 | 0.0 |
| Cyan | 0.0 | 1.0 | 1.0 |
| Magenta | 1.0 | 0.0 | 1.0 |
| Black | 0.0 | 0.0 | 0.0 |
| Aquamarine | 0.439216 | 0.858824 | 0.576471 |
| BlueViolet | 0.62352 | 0.372549 | 0.623529 |

| | | | |
|-------------------|----------|----------|----------|
| Brown | 0.647059 | 0.164706 | 0.164706 |
| CadetBlue | 0.372549 | 0.623529 | 0.623529 |
| Coral | 1.0 | 0.498039 | 0.0 |
| CornflowerBlue | 0.258824 | 0.258824 | 0.435294 |
| DarkGreen | 0.184314 | 0.309804 | 0.184314 |
| DarkOliveGreen | 0.309804 | 0.309804 | 0.184314 |
| DarkOrchid | 0.6 | 0.196078 | 0.8 |
| DarkSlateBlue | 0.419608 | 0.137255 | 0.556863 |
| DarkSlateGray | 0.184314 | 0.309804 | 0.309804 |
| DarkSlateGrey | 0.184314 | 0.309804 | 0.309804 |
| DarkTurquoise | 0.439216 | 0.576471 | 0.858824 |
| Firebrick | 0.556863 | 0.137255 | 0.137255 |
| ForestGreen | 0.137255 | 0.556863 | 0.137255 |
| Gold | 0.8 | 0.498039 | 0.196078 |
| Goldenrod | 0.858824 | 0.858824 | 0.439216 |
| GreenYellow | 0.576471 | 0.858824 | 0.439216 |
| IndianRed | 0.309804 | 0.184314 | 0.184314 |
| Khaki | 0.623529 | 0.623529 | 0.372549 |
| LightBlue | 0.74902 | 0.847059 | 0.847059 |
| LightSteelBlue | 0.560784 | 0.560784 | 0.737255 |
| LimeGreen | 0.196078 | 0.8 | 0.196078 |
| Maroon | 0.556863 | 0.137255 | 0.419608 |
| MediumAquaMarine | 0.196078 | 0.8 | 0.6 |
| MediumBlue | 0.196078 | 0.196078 | 0.8 |
| MediumForestGreen | 0.419608 | 0.556863 | 0.137255 |
| MediumGoldenrod | 0.917647 | 0.917647 | 0.678431 |
| MediumOrchid | 0.576471 | 0.439216 | 0.858824 |
| MediumSeaGreen | 0.258824 | 0.435294 | 0.258824 |
| MediumSlateBlue | 0.498039 | 0.0 | 1.0 |
| MediumSpringGreen | 0.498039 | 1.0 | 0.0 |
| MediumTurquoise | 0.439216 | 0.858824 | 0.858824 |
| MediumVioletRed | 0.858824 | 0.439216 | 0.576471 |
| MidnightBlue | 0.184314 | 0.184314 | 0.309804 |
| Navy | 0.137255 | 0.137255 | 0.556863 |
| NavyBlue | 0.137255 | 0.137255 | 0.556863 |
| Orange | 1.0 | 0.5 | 0.0 |
| OrangeRed | 1.0 | 0.0 | 0.498039 |
| Orchid | 0.858824 | 0.439216 | 0.858824 |
| PaleGreen | 0.560784 | 0.737255 | 0.560784 |
| Pink | 0.737255 | 0.560784 | 0.560784 |
| Plum | 0.917647 | 0.678431 | 0.917647 |
| Salmon | 0.435294 | 0.258824 | 0.258824 |
| SeaGreen | 0.137255 | 0.556863 | 0.419608 |
| Sienna | 0.556863 | 0.419608 | 0.137255 |
| SkyBlue | 0.196078 | 0.6 | 0.8 |
| SlateBlue | 0.0 | 0.498039 | 1.0 |
| SpringGreen | 0.0 | 1.0 | 0.498039 |
| SteelBlue | 0.137255 | 0.419608 | 0.556863 |
| Tan | 0.858824 | 0.576471 | 0.439216 |
| Thistle | 0.847059 | 0.74902 | 0.847059 |

| | | | |
|-----------------|----------|----------|----------|
| Turquoise | 0.678431 | 0.917647 | 0.917647 |
| Violet | 0.309804 | 0.184314 | 0.309804 |
| VioletRed | 0.8 | 0.196078 | 0.6 |
| Wheat | 0.847059 | 0.847059 | 0.74902 |
| YellowGreen | 0.6 | 0.8 | 0.196078 |
| SummerSky | 0.22 | 0.69 | 0.87 |
| RichBlue | 0.35 | 0.3 | 0.67 |
| Brass | 0.71 | 0.65 | 0.26 |
| Copper | 0.72 | 0.45 | 0.20 |
| Bronze | 0.55 | 0.47 | 0.14 |
| Bronze2 | 0.65 | 0.49 | 0.24 |
| Silver | 0.90 | 0.91 | 0.98 |
| BrightGold | 0.85 | 0.85 | 0.10 |
| OldGold | 0.81 | 0.71 | 0.23 |
| Feldspar | 0.82 | 0.57 | 0.46 |
| Quartz | 0.85 | 0.85 | 0.95 |
| NeonPink | 1.00 | 0.43 | 0.78 |
| DarkPurple | 0.53 | 0.12 | 0.47 |
| NeonBlue | 0.30 | 0.30 | 1.00 |
| CoolCopper | 0.85 | 0.53 | 0.10 |
| MandarinOrange | 0.89 | 0.47 | 0.20 |
| LightWood | 0.91 | 0.76 | 0.65 |
| MediumWood | 0.65 | 0.50 | 0.39 |
| DarkWood | 0.52 | 0.37 | 0.26 |
| SpicyPink | 1.00 | 0.11 | 0.68 |
| SemiSweetChoc | 0.42 | 0.26 | 0.15 |
| BakersChoc | 0.36 | 0.20 | 0.09 |
| Flesh | 0.96 | 0.80 | 0.69 |
| NewTan | 0.92 | 0.78 | 0.62 |
| NewMidnightBlue | 0.00 | 0.00 | 0.61 |
| VeryDarkBrown | 0.35 | 0.16 | 0.14 |
| DarkBrown | 0.36 | 0.25 | 0.20 |
| DarkTan | 0.59 | 0.41 | 0.31 |
| GreenCopper | 0.32 | 0.49 | 0.46 |
| DkGreenCopper | 0.29 | 0.46 | 0.43 |
| DustyRose | 0.52 | 0.39 | 0.39 |
| HuntersGreen | 0.13 | 0.37 | 0.31 |
| Scarlet | 0.55 | 0.09 | 0.09 |

5.2 ファイル **shapes.inc** で定義された形状(主なものに限定)

| | |
|--------------------------|--------------------------|
| Paraboloid_X (x 軸方向の放物面) | Paraboloid_Y (y 軸方向の放物面) |
| Paraboloid_Z (z 軸方向の放物面) | Hyperboloid (双曲面) |
| Hyperboloid_Y (双曲面) | |

5.3 ファイル **shapes2.inc** で定義された形状(主なものに限定)

| | | |
|---------------------|-------------------|----------------------|
| Tetrahedron (正四面体) | Octahedron (正八面体) | Dodecahedron (正十二面体) |
| Icosahedron (正二十面体) | Hexagon (正六角柱) | Pyramid (ピラミッド) |
| Pyramid2 (ピラミッド) | | |

5.4 ファイル **shapesq.inc** で定義された形状

| | | | |
|--------------------|------------------|------------------------|----------------|
| Bicorn | Crossed_Trough | Cubic_Cylinder | Cubic_Saddle_1 |
| Devils_Curve | Folium | Glob_5 | Twin_Glob |
| Helix | Helix_1 | Hyperbolic_Torus_40_12 | Lemniscate |
| Quartic_Loop_1 | Monkey_Saddle | Parabolic_Torus_40_12 | Piriform |
| Quartic_Paraboloid | Quartic_Cylinder | Steiner_Surface | Torus_40_12 |
| Sinsurf | | | Witch_Hat |

5.5 ファイル **textures.inc** で定義されたテクスチャ

| | | | | |
|----------------|-----------------|---------------|-----------------|----------------|
| Jade | Red_Marble | White_Marble | Blood_Marble | Blue_Agate |
| Sapphire_Agate | Brown_Agate | Pink_Granite | PinkAlabaster | |
| Blue_Sky | Bright_Blue_Sky | Blue_Sky2 | Blue_Sky3 | Blood_Sky |
| Apocalypse | Clouds | | | |
| Cherry_Wood | Pine_Wood | Dark_Wood | Tan_Wood | White_Wood |
| Tom_Wood | DMFWood1 | DMFWood2 | DMFWood3 | DMFWood4 |
| DMFWood5 | DMFLightOak | DMFDarkOak | EMBWood1 | Yellow_Pine |
| Rosewood | Sandalwood | | | |
| Glass | Glass2 | Glass3 | Green_Glass | |
| Metal | Chrome_Texture | Brass_Texture | Gold_Texture | Bronze_Texture |
| Copper_Texture | Silver_Texture | Chrome_Metal | Brass_Metal | Bronze_Metal |
| Gold_Metal | Bronze_Metal | Copper_Metal | Polished_Chrome | Polished_Brass |
| Spun_Brass | Silver1 | Silver2 | Silver3 | Brass_Valley |
| Peel | Water | Cork | | |

5.6 ファイル **textures.inc** で定義された表面仕上げ

| | |
|--------------|--|
| Dull | finish {specular 0.5 roughness 0.15} |
| Shiny | finish {specular 1 roughness 0.001} |
| Phong_Dull | finish {phong 0.5 phong_size 1} |
| Phong_Shiny | finish {phong 1 phong_size 200} |
| Glossy | finish {specular 1 roughness 0.0001 reflection 0.13} |
| Phong_Glossy | finish {phong 1 phong_size 300 reflection 0.13} |
| Luminous | finish {ambient 1 diffuse 0} |
| Mirror | finish {ambient 0 diffuse 0 reflection 1} |

5.7 ファイル **stones.inc** で定義されたテクスチャ

| | |
|----------|--|
| T_Stone1 | Deep Rose & Green Marble with large White Swirls |
| T_Stone2 | Light Greenish Tan Marble with Agate style veining |
| T_Stone3 | Rose & Yellow Marble with fog white veining |
| T_Stone4 | Tan Marble with Rose patches |
| T_Stone5 | White Cream Marble with Pink veining |
| T_Stone6 | Rose & Yellow Cream Marble |
| T_Stone7 | Light Coffee Marble with darker patches |

| | |
|-----------|---|
| T_Stone8 | Gray Granite with white patches |
| T_Stone9 | White & Light Blue Marble with light violets |
| T_Stone10 | Dark Brown & Tan swirl Granite with gray undertones |
| T_Stone11 | Rose & White Marble with dark tan swirl |
| T_Stone12 | White & Pinkish Tan Marble |
| T_Stone13 | Medium Gray Blue Marble |
| T_Stone14 | Tan & Olive Marble with gray white veins |
| T_Stone15 | Deep Gray Marble with white veining |
| T_Stone16 | Peach & Yellow Marble with white veining |
| T_Stone17 | White Marble with gray veining |
| T_Stone18 | Green Jade with white veining |
| T_Stone19 | Peach Granite with white patches & green trim |
| T_Stone20 | Brown & Olive Marble with white veining |
| T_Stone21 | Red Marble with gray & white veining |
| T_Stone22 | Dark Tan Marble with gray & white veining |
| T_Stone23 | Peach & Cream Marble with orange veining |
| T_Stone24 | Green & Tan Moss Marble |

付録. FAQ (Frequently Asked Questions)

◆Q1. 光源が暗い

光源の色を指定するとき、例えば 2*White とすると 2 倍の明るさになる。

◆Q2. 透明物体の和集合

ガラスのような透明物体同士の和集合を作ったとき、union では元の物体の面が透けて見える。完全に融合して境界面が残らないようにするには union の代わりに merge と指定すればよい。

◆Q3. 差集合の結果がおかしい

物体 A から物体 B を引くとき、両物体の表面が一致していると、そこに厚さ 0 の膜が残り、計算誤差によって、そこに物体がある/ないように描かれたりして、おかしい絵になる場合がある。物体 B が物体 A からはずれはみ出すように指定すれば、そのようなことは起こらない。また、difference を一時的に union に変えてみると、両物体の位置を確認できる。

◆Q4. 回転の結果がおかしい

回転は常に座標軸を中心にして行う。原点の位置で水平な姿勢で定義した物体を、傾いた姿勢で原点からはずれた位置に設定したいときには、先に回転してから移動すればよい。すなわち、

rotate < x, y, z >

translate < x, y, z >

の順序で書く。これを逆にすると物体の位置を制御するのが難しくなる。

◆Q5. 座標軸の設定

左右対称形の場合は、原点が真ん中にくるような位置で定義してから、必要な角度に回転し、さらに適切な場所に移動するのがよい(Q4 も参照のこと)。

◆Q6. 効率的な作業法

複雑な形状を一気にモデリングするのではなく「一つモデリング要素を加えたらレンダリング」するスタイルを繰り返した方が効率が良い。最終画像以外は 320×240 No AA を指定するとよい。

◆Q7. 光源の位置

物体から離れた方がよい結果の得られることが多い。

◆Q8. 類似の物体をたくさん並べたい

類似した物体を一定の規則に従って配置したい場合には、プログラムのループのような記述を利用することもできる。一例を示す。

```
#include "colors.inc"
#include "textures.inc"

camera { location <-4, 2, 3> look_at <0, 0, 0> }

light_source { <-20, 30, -25> color White }

#declare Count = 0;
#while ( Count < 5 )
    sphere { <0.7 * Count - 1.4, 0, 0> 0.2
        pigment { Gray30 } finish { Shiny }
    }
#declare Count = Count + 1;
#end

box { <-2, -0.4, -2>, <2, -0.2, 2>
    pigment { Gray80 } finish { Dull }
}

background { White }
```

この例では、次の点を中心としてもつ球を5個設定した。

$\langle -1.4, 0, 0 \rangle, \langle -0.7, 0, 0 \rangle, \langle 0, 0, 0 \rangle, \langle 0.7, 0, 0 \rangle, \langle 1.4, 0, 0 \rangle$

変数や式は、座標だけでなく、半径や色など数値を書けるところならどこでも使える。

下の例に示すような、物体のマクロ定義も可能である。

```
#declare Samp = merge{
    difference { box { <-2, 4, -2>, <2, 6, 2> }
        sphere { <0, 5.5, 0>, 1.5 } }
    cylinder { <0, 0, 0>, <0, 5, 0>, 0.5 }
    box { <-2, 0, -2>, <2, 0.5, 2> }
}

object { Samp texture { Cork } }
object { Samp translate <4, 0, 0> pigment { Red } }
```

◆Q9. 背景に模様を描きたい

背景色の指定を行う `background { }` では、色の指定しかできない。テクスチャを描きたい場合には、物体の背後に平面や大きな直方体を置き、その表面にテクスチャを指定すればよい。

◆Q10. 表示する文字が小さい

プログラムの文字が小さすぎて読みにくい場合には、ウインドウ内でマウスを右クリックし、現れるメニューから [Properties\(Colors, Keystrokes, etc.\)](#)→[Color/Font](#)→[Change](#) の順に選択した画面で設定する。