POV-Ray をつかった CG 画像の作成

田中 宏樹

1.研究概要

3 次元コンピューターグラフィックス（3DCG）について学びリアルな画像を作成する。

2.具体的内容

今回は、POV-Ray（ポブ・レイ）と言うソフトを使って 3DCG 画像を作成する。

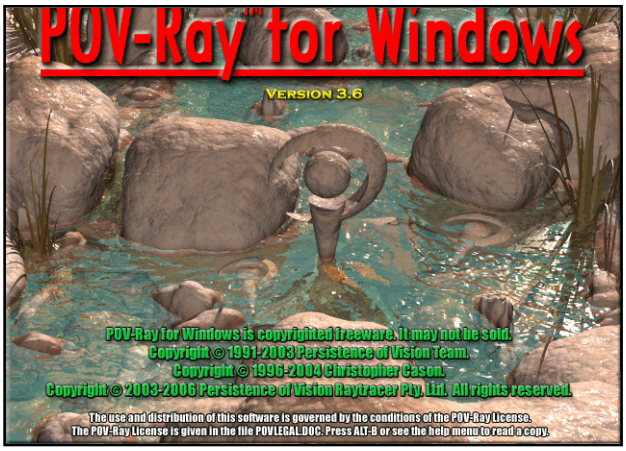

図 1　POV-Ray

(1) POV-Ray とは

CompuServe(コンピュサーブ)と言う、米国の大規模な通信ネットワークを中心とした、「POV-Ray Team」と言う団体によって開発されたフリーソフトの名称である。

「POV-Ray」のシーンファイルは拡張子が「.pov」のテキストファイルで、3D CG を作り上げるのに必要な要素をすべてプログラムによって記述される。

また、メモ帳などで作成した「.txt」の拡張子を「.pov」に変更することで POV-Ray で読み込み可能になる。

今までに 3D CG を制作した事のある人や 3D CG の初心者には難しい部分も多いがレンダラーとして使うにはお勧めである。

ちなみに POV-Ray では、「左手系」という方法が使われている。

(2) POV-Ray の問題点など

今回使用したソフト POV-Ray はフリーソフトでありながらとてもリアルな画像を作成することが出来るが、他の 3DCG 作成ソフトと違いプログラム言語の記述によって画像を作成するので操作に癖があり慣れるまでに時間がかかる。

また、POV-Ray のみで 3DCG を作成しようとするとあらゆるオブジェクトを作成する為のプログラムを覚えなくてはならないので大変である。

その他に、画像をリアルタイムに見ることが出来ないことも短所の一つと言えるだろう。

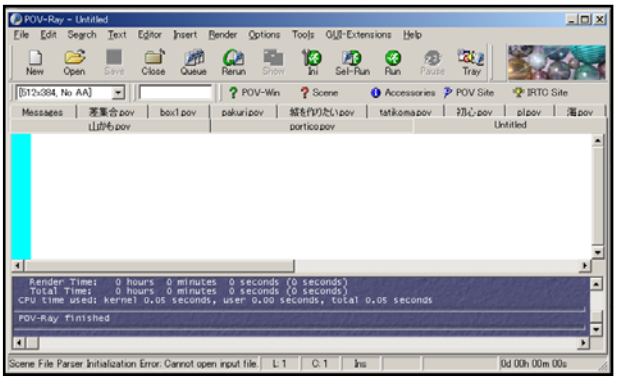

図 2　操作画面

(3) 作業手順

3DCG を作成する手順は物体の形状を作成しその物体に色や材質の設定をする「モデリング」作業と作成した画像に当たる照明・反射などの計算や視点・視線の設定を行い、画像を作成する「レンダリング」作業の 2 つに分けることが出来る。

(4) 練習

まず、使い方になれるために練習として基本的な図形を使った画像を作ることにした。

POV-Ray はまだまだ参考文献も少なく本を探すよりもインターネット上で利用者の解説などを探すほうが量も多そうだったのでネット上で日本語の利用参考マニュアルが公開されていたのでそれを元に作成した。

プログラム例

ここでは、POV-Ray で使う簡単なプログラム等について説明をしていく。

まずは、先ほど書いた「左手系」について説明する。

「左手系」とは「フレミングの左手の法則」の形に左手を開いて左手の親指を y 軸、人差し指を z 軸とした時、中指が x 軸の方向になる。

次に、POV-Ray のカメラや照明の設定から説明して行く。POV-Ray ではこのプログラムを打ち込まないと画像の作成は出来ない。

```
Camera { location <0,10,0>   ①
     look at <0,3,0>         ②
     angle 80 }              ③
light_source { <-10,20,-30>  ④
 color rgb<1,1,1> }          ⑤
```

①でカメラの位置、②ではカメラで見る方向③では物体からの距離の指定をする。

④は、照明の指定、⑤では照明の色を指定していく。

カッコ内の値は順々に x、y、z の値を指定する。しかし、⑤は別で Red,Green,Blue（赤、緑、青）の頭文字で指定され、それぞれ 0～1 の値で指定される。ちなみに、白だと<1,1,1> 黒だと<0,0,0>となる。

また、White、Black、Red などのつづりも扱うことが可能である。

これでひとまず、カメラ・照明の指定は終わる。しかし、映るものがないのでこのままでは真っ暗になってしまう。

そこで、床面の設定をしてレンダリングする。

床面の指定は、

```
plane{  y,0･･･  ①
        pigment { color rgb<0,0.5,1>}
        }
```

① の plane が無限に続く平面を作成するプログラムで y,0 は x、y、z の内床を作りたい面 y とその座標 0 の指定を行う。

また「pigment」以降で床の柄や色の指定を行う。

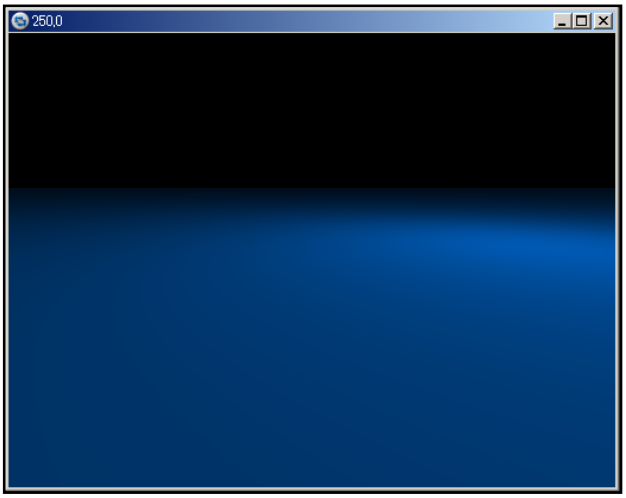

図 3　床面

これでは地平線がわからないので「空」を指定する。

プログラムはこんな感じ

```
background { rgb<0.3,0.4,1.2>}
```

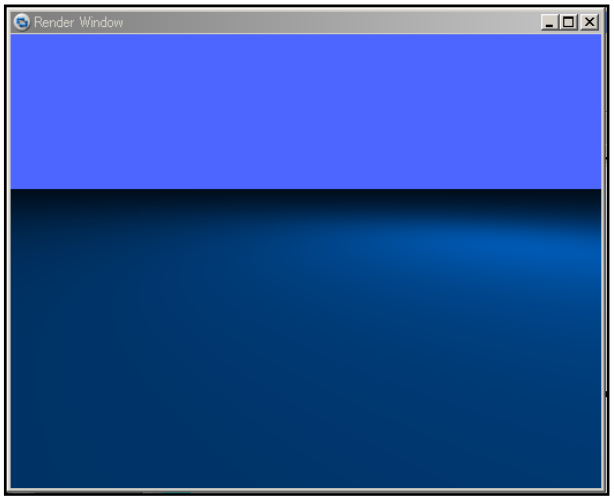

図 4　空

なにか地面と言うより海面みたいになっていることに今頃気づいてしまったので床に柄を指定すしより床らしくする。

```
pigment { checker color rgb <1,1,1>
          color rgb<0,0.5,1>}}
```

に変更すると、

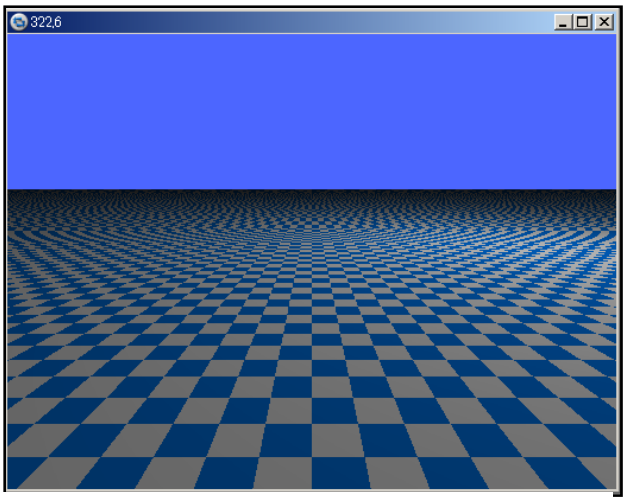

図５　床（模様）

このような感じになり床と空と言うありえない組み合わせにはなってしまったがさっきより大分まともになった。

次に、簡単なオブジェクトのプログラムを説明する。

直方体を作るプログラムは、

```
box {<3, 3, 3>, <1, 1, 1>          ・・・①
    pigment{color rgb<1, 1, 1>} }
```

である。

①　は、直方体の対角の頂点座標 x，y，z を指定しその間にオブジェクトを作成する。

構文は「box」である。

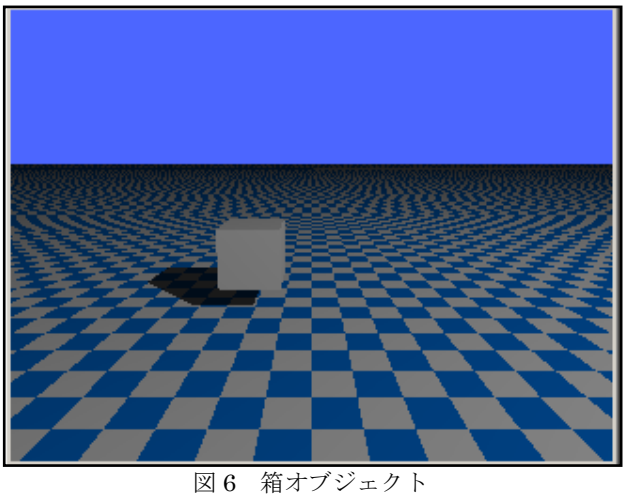

図６　箱オブジェクト

次に、球体を作るプログラムを説明する。

```
sphere {  <1.5, 3, -2>                ・・・①
          1                           ・・・②
          pigment{color rgb<1, 1, 1>} }
```

球体の場合は①で中心点 x，y，z を指定しそこからの半径を②で指定し計算することによりオブジェクトを作成する。

構文は「sphere」である。

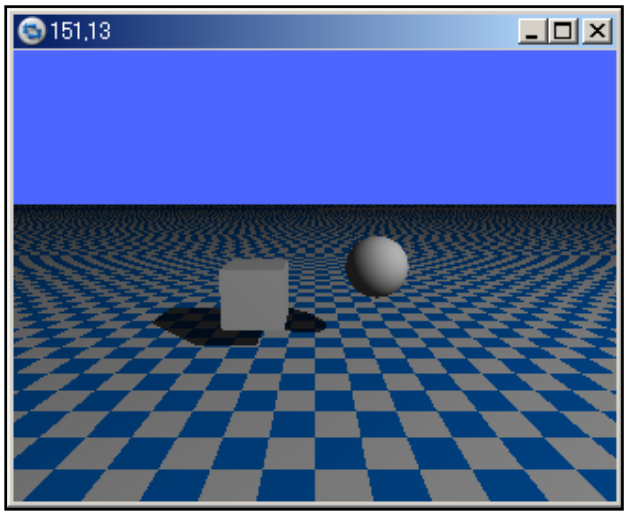

図７　球体オブジェクト

３.課題研究のまとめ

今回作成した作品は、アニメ「攻殻機動隊」と言うアニメ作品に登場する「タチコマ」と言う AI ロボット。

初めは POV-Ray でモデリングからレンダリングまで行って画像の完成させるつもりでしたが POV-Ray はレンダリングについてはとてもきれいに出来ますがモデリング作業には癖が多く細かい作業にはあまり向いていないと言うことがわかった。

なので、今回作ったタチコマは、メタセコイアというソフトを使ってモデリングして出来た画像を POV-Ray を使ってレンダリングする事にした。

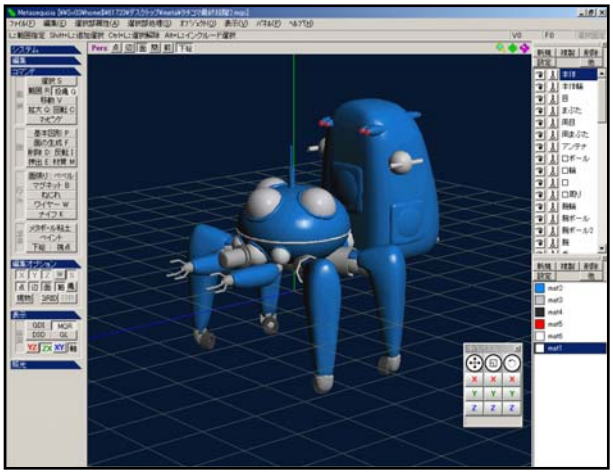

図８　メタセコイア

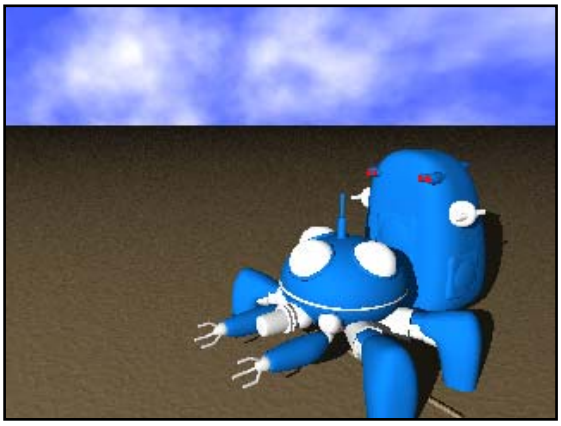

図９　POV-Ray レンダリング

４.感想

今回の課題研究では、フリーソフトでも上手に使えばきれいな作品を作成することが出来るということがわかった。

しかし、ソフトによって特性がありそのソフトの特性を生かすも殺すも使い手の力量しだいという事がとても心にしみた。

なので、まだまだ「POV-Ray」を使いこなせていると言えないのでソフトを使って色々な画像を作成し、使うソフトを最大限に生かせるようになることがこれからの課題だ。

また、家のパソコンでは何なく動くソフトが学校のパソコンではまったく動かなかったり動作にとても時間がかかったりと使うパソコンによって出来る作業が限られてくるのでパソコン環境の大切さも新ためて実感した。

そこでしたい作業にあったパソコンを使うことや使うパソコンにあった作業を行うことの大切さを学んだ。

今のところまだ不完全な作品しか出来上がっていないのでこれからも続けて作成してこうと思う。

５.参考文献

POV-Ray
The Persistence of Vision Raytracer
http://www.povray.org/

MetasequoiaLE
http://www.metaseq.net/

POVRAY station
http://nishimulabo.edhs.ynu.ac.jp/~povray/beginner/