

本稿は [Linux Japan 誌](#) 2001 年 12 月号に掲載された記事に補筆修正したものです。

簡易動画作成

前回の数式処理 Maxima の紹介の最後にグラフィック出力の話をしました。その時「そういえば GIF アニメーションや MPEG への変換はどうなってたかなあ」と気になってしまい、今回はその話題をまとめておきたいと思います。ただ、ネットワークで動画をリアルタイムに配信するのが当たり前の時代に、MPEG-1 や GIF アニメーションは古いかもしれません。しかし、動画を教材の一部として用いるような用途には十分なので、その先の話はまたの機会ということで...

セル画像の準備

アニメーションに必要な画像は今や定番のレイトレーシングツール POV-Ray^[1]^[W³] で作成します。物理教材では、複雑な形状の物体を扱いません。問題は位置や角度を正確に指定できることで、POV-Ray のスクリプトはこの目的に適しています。あまりにも単純かもしれませんが、水平に投げ出された球の放物運動を題材にします。プログラム parab から出力された位置をスクリプトに取り込み POV-Ray で PNG 画像に変換するシェルスクリプトを作成します。

POV-Ray

シェルスクリプト thrown.sh の説明をしましょう。“echo |povray ... -I'-'... <<EOF” はヒアドキュメントで、行の先頭に EOF が現れるまで、echo します。そのパイプ出力先が POV-Ray で、POV-Ray 自身の入力を標準入力 '- ' に指定します。すなわち、POV-Ray に直接コマンドを送り込んでいる訳です。そして、球の位置をスクリプト末尾の “sphere{ <\$2,\$3,\$4>...}” の部分でスクリプトに与えられた引数により指定しています。全体の流れは以上ですが、その他の項目について POV-Ray の使い方のおさらいを兼ねて、簡単に延べます。

リスト 1 parab.c

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

main(int argc, char **argv)
{
    double x;
    x=atof(argv[1]);
    printf("%f %f %f\n", x, 36-(0.1*x)*(0.1*x), 0.5*x);
}
```

カメラの設定：camera z 軸 (奥行き) の手前に引いて設置し、画角を絞っています (up,right で調整)。望遠撮影になりますから、周辺の歪みのない画像が得られます。物理教材では正確な位置を示すことが重要なので、つまらないですがこうしています。なお、臨場感を出すならば、カメラを寄って撮影すると良いでしょう。カメラの位置を移動させる設定ができるともっと面白くなるはずで、そういうツールもあります。

光源の設置：light_source 右斜めに設置しています。遠くに離して全体が照らされるように調節しました。

オブジェクトの配置 被写体であるオブジェクトを置きます。今回の例では、チェッカー盤模様の水平床 (plain) と、黄色い球です。繰り返しますが、球は放物運動をしますので、時間に応じた x, y, z 座標を parab というコマンドで計算させ、その結果を取り込んでいます。リスト 2 thrown.sh

```
#!/bin/sh

echo "Rendering $1..."
echo |povray -W320 -H240 -d0 -I'-' -O"$1.png"
    2> /dev/null << EOF
#include "colors.inc"
#include "shapes.inc"
#include "textures.inc"
#include "finish.inc"
#include "stones.inc"

#version 3.0

global_settings {
    assumed_gamma 1.6
    max_trace_level 5
}

background { color Black }

light_source { <50, 200, -200> color White }

camera {
    up <0,0.09,0>
    right <0.12,0,0>
    location <30, 10, -800>
    look_at <30, 20, 0>
}

plane { <0, 1, 0>, -0.1
    pigment {
        checker color White color Blue
        scale 3
    }
    finish { ambient.2 }
}

sphere {
    <$2, $3, $4>, 2
    pigment { color Yellow }
    finish { Metal ambient 0.3}
}
EOF
```

この thrown.sh にセル画像の通し番号と parab から
の計算結果を引数として渡すためのシェルスクリプト
を以下にします。

```
#!/bin/sh

for i in `seq $2`; do
  j=`printf "%04d" $i`
  ./thrown.sh $j `seq $1 $i`
done;
```

番号が 0001, 0002... となるように, printf で書式を
指定して \$i から \$j を生成しています。

以下のように実行して, 放物運動の位置計算のプ
ログラム parab を用いて, 60 枚の画像を作成します。

```
./anim.sh ./parab 60
```

これで, コマを収めた一連のセル画像 0001.png,
0002.png ... が出来ました。

GIF アニメーション

アニメーション化

一連のセル画像をまとめてアニメーションにする
には, ImageMagick[2][W³] の convert が便利です。

```
convert -delay 3 *.png thrown.gif
```

-delay で画像の切り替え時間を 1/100 秒単位で指定
できるようになっています。今回の放物運動のアニメ
ーションは全部で 60 枚のコマがあり, convert でまと
めると 4.7MB になってしまいました。これでは使えませ
ん。そこで, 他の gif 専用ツールを使うことにします。

gifsicle[3][W³] は元画像が gif である必要がありま
すが, 最適化オプション -O2 を用いると, convert より
も相当小さなファイルを作成します。-d は delay, -k
は色数を指定するオプションです。

```
gifsicle -d3 -k256 -O2 *.gif >thrownGS.gif
```

コマ毎の gif 画像ファイル残したくないならば, gifsicle
は filter としても働きますから, convert と組み合わ
せて

```
convert *.png gif:- |gifsicle
-d3 -k256 -O2 >thrownGS.gif
```

などとすれば旨いきます。gifsicle で変換したら
なんとたった 18kB です。convert に比べて驚異の 1/200。
convert で作成した thrown.gif にも最適化をかけて
変換すると同程度の小さなサイズになります。

```
gifsicle -d3 -k256 -O2 thrown.gif >thrownS.gif
```

この最適化が何を意味するかは GIF Animation につ

いて調べないといけません [4][W³]。今回の例題アニメ
に関しては whirlgif[5][W³] というツールはもっと小
さなサイズのもを生成してくれました。

再生

GIF アニメーションは Netscape などの WWW ブラ
ウザーで標準的にサポートされているので, 特許の
問題があるにもかかわらず今でも頻りに用いられて
いるのだと思います。netscape でどのように表示され
るかは

```
netscape thrown.gif
```

とすれば判りますが, わざわざ重たい netscape を起動
するのもなんですから, ImageMagick の animate を
使ってみましょう。



図 1 ImageMagick の animate のメニュー

```
animate -delay * thrown.gif
```

オプション -delay は convert に同じで, 再生時にも
コマの切り替え時間を指定できるというわけです。さら
に, 画像をクリックしてメニューを表示させ, Animate
で Auto Reverseなどをさせると楽しいです。これ以
外にも xanim[6][W³] などの定番ツールなどがあって,
お好みでということでしょうか。



図 2 Xanim のコントロールボタン

MPEG

GIF アニメーションは LZW で圧縮しないと結構大きなファイルができてしまいますし、音声を含ませることができません。そこで音声も入るし、画質に応じてサイズを小さくすることが可能な MPEG に変換してみましょう。ただし今回は音声を入れません。また、ネットワークでリアルタイムに流す目的ではないので、MPEG-1 と MPEG-2 の差はあまり問題になりません。ソフトウェア変換のツールは意外に少なく、MPEG-1 へはバークレイで開発された `mpeg_encode`^[7]^{W³}、MPEG-2 への変換は `mpeg2encode`^[8]^{W³} しか見当たりませんでした。再生ツールは SDL(Simple DirectMedia Layer)^[9]^{W³} ライブラリもあり、ある程度種類が豊富です。

エンコーディング

まず MPEG-1 への変換は `mpeg_encode` を用いて

```
mpeg_encode thrown.param
```

とします。次のようなパラメータファイル(上の例の `thrown.param`)に必要な事柄を書かなければならずちょっと面倒ですが、ソースを展開すると `example` というサブディレクトリにパラメータファイルの例が豊富にありますのでそれを眺めればおおよそ検討がつかます。少しだけ説明をしましょう。入力画像形式(`BASE_FILE_FORMAT`)は、YUV、PPM、PNM、JPEG、JMOVE ですが、最終的には YUV に変換されます。YUV は自身に画像の大きさの情報を持っておらず、外部から知らせる必要があります。動画作成のアルゴリズム `PSEARCH_ALG`、`BSEARCH_ALG` により、品質とサイズ(むろん両者は背反します)が決まります。がしかし、意外と品質の差が判りにくく、むしろ変換に要する時間の差が気になりました。

この例では、アルゴリズムは雛形そのまま、元画像形式を PPM に設定してあります。といっても POV-Ray の出力は PNG ですから、`INPUT_CONVERT` でたまたまや ImageMagick の `convert` のお世話になって PPM に変換しています。すなわち、予め PPM (PNG よりもかなりサイズが大きくなります) に変換しておく必要はないということです。`INPUT` から `END_INPUT` の間に、入力画像を番号の形式を含めて指定します。“*” がワイルドカードで “[]” 内で指定した番号に置き換わります。

他の動画を作成する場合、以上の説明部分をそのま

まにしておくことにすれば、本当に書き換えが必要なのは `OUTPUT`、`YUV_SIZE`、`INPUT` から `END_INPUT` の間、だけになります。

```
PATTERN IBBPBBPBBPBBPBB
OUTPUT thrown.mpg
GOP_SIZE 30
SLICES_PER_FRAME 1
BASE_FILE_FORMAT PPM
YUV_FORMAT UCB
YUV_SIZE 320x240
INPUT_CONVERT convert * ppm:-
INPUT_DIR ./

INPUT
*.png [0001-0060]
END_INPUT

PIXEL HALF
RANGE 10
PSEARCH_ALG LOGARITHMIC
BSEARCH_ALG CROSS2
IQSCALE 8
PQSCALE 10
BQSCALE 25
REFERENCE_FRAME DECODED
```

とにかく MPEG-1 に変換したいというならば、後述の `mpeg2encode` をインストールして `convert` を使しましょう。

```
convert *.png thrown.mpg
```

だけで済みます。ただ出来る上がるファイルのサイズは大きくなってしまいました。

つづいて MPEG-2 も筆者の知るところ唯一のツール、`mpeg2vidcodev_v12.tar.gz`^[8]^{W³} の `mpeg2encode` しかないようです。このツールも面倒なパラメータファイルを必要とします。すなわち

```
mpeg2encode MPEG-2.par thrown.m2v
```

のように実行するのですが、`MPEG-2.par` の中身がさっぱりわかりません(^^; 無責任とは思いますが、パラメータファイルの謎解きは読者の皆さんの宿題として、ここでは ImageMagick の `convert` に頼ってしましましょう。

```
convert *.png thrown.m2v
```

拡張子 `m2v` を指定すると `mpeg2encode` により MPEG-2 にエンコードしてくれます。ワーニングがかなりありますが、とにかく出来上がります。`convert` は内部で変換できない形式については外部ツールを縦横無尽に利用しています。単に“convert”とすると、利用している外部ツールの一覧が表示されます。

再生

再生用のツール、いわゆるプレーヤーは結構種類があります。MPEG-2 のものは MPEG-1 にも対応している場合が多いです。まあ順番に MPEG-1 のみに対応しているものから挙げていきましょう。まずは、mpeg_encode を開発しているパークレイのグループから提供されている mpeg_play[7][W³] です。デフォルトをフルカラーにするパッチも当てることをお勧めします [10][W³] 。



図 3 パークレイの mpeg_play のメニュー

```
mpeg_play thrown.mpg
```

残念ながら、音声は再生しません。メニューが別窓に現れ、必要最小限の機能があります(図 3)。SDL の上に構築された SMPEG (SDL MPEG Player Library)[11][W³] はビデオ再生部分をこの mpeg_play に基づいて開発したとあります。SMPEG は音声再生も可能であり、ライブラリ構築時にテストアプリケーション plaympeg (コマンドライン), gtv (GTK++ による plaympeg のラッパー), glmovie (フルスクリーン再生) が出来ます。xtheater[12][W³] は VCD の再生も可能です。最後に、一時これしかないと人気を博した mtv[13][W³] のコマンドライン版 mtvp は個人使用に関してはライセンスフリーなので安心して使えます。

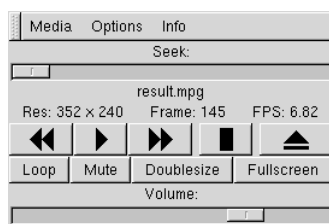


図 4 VCD も再生可能な xtheater

MPEG-2 についてですが、まず mpeg2encode と対をなしている mpeg2decode は MPEG-2 をフレームに分解するツールであってプレーヤーではありません。画面 (X11) に表示する MPEG-2 プレーヤーは他を探さなければなりません。SMPEG のページには XMPS[14][W³] というスキンを装備して凝った概観を持つことができるものへのリンクがあります。ただし、まだかなり不安定です。SMPEG から離れて探すと、Mplayer[15][W³] と Xmovie[16][W³] が見つかりま

した。Mplayer は GUI が未熟ですが、コマンドラインは安定しています。Xmovie は、最新のバージョン 1.9.1 が kernel 2.4 と XFree86-4 (XVideo エクステンション) を要求しているため、コンパイルが難しいです。図 6 は古いバージョン (1.5.3) のものです。

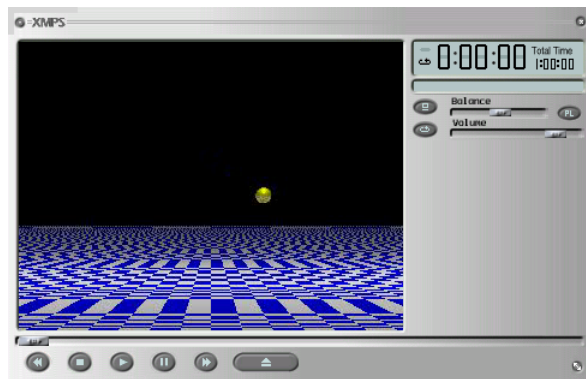


図 5 凝った概観の xmps



図 6 多種類のビデオ形式に対応している xmovie

参考文献

- [1] レイトレーシングの定番 Povray . [W³]
<http://www.povray.org/>
- [2] 毎度お馴染み ImageMagick . [W³]
<http://www.ImageMagick.org/>
- [3] Gifsicle の公式サイト . [W³]
<http://www.lcdf.org/gifsicle/>
- [4] “GIF Animation Studio” という評価の高い本の紹介ページ . [W³]
<http://www.webreference.com/content/studio/>

- [5] Whirlgif の開発者のページ . [W³](#)
<http://www.danbbs.dk/~dino/whirlgif/>
- [6] しばらく更新されていないですが , 定番 Xanim .
[W³](#)
<http://xanim.va.pubnix.com/home.html>
- [7] バークレーの MPEG 開発グループの FTP サイト . [W³](#)
<ftp://mm-ftp.CS.Berkeley.EDU/pub/mpeg/>
- [8] MPEG-2 ツールの FTP サイト . [W³](#)
<ftp://ftp.mpeg.org/pub/mpeg/mssg/>
- [9] Linux でゲームを作成するのに必須のサイト . [W³](#)
<http://www.libsdl.org/>
- [10] mpeg_play の色数のデフォルトを変更するパッチ .
[W³](#)
<http://phe.phyas.aichi-edu.ac.jp/~cyamauch/pl2.html#pkg>
- [11] SDL の MPEG 関連ライブラリ . [W³](#)
<http://www.lokigames.com/development/smpeg.php3>
- [12] VCD も再生できる Xtheater のソース配布サイト .
[W³](#)
<http://Xtheater.sourceforge.net/>
- [13] 非常に有名な MPEG 再生プレーヤーです . コマンドライン版 mtvp は個人使用に限りフリーです .
[W³](#)
<http://www.mpegtv.com/>
- [14] スキンを装備した見栄え重視の MPEG-1 プレーヤー . [W³](#)
<http://xmps.sourceforge.net/>
- [15] コマンドラインは安定している Mplayer . [W³](#)
<http://www.mplayerhq.hu/homepage/about.html>
- [16] MPEG-2 や Quick Time も再生できるプレーヤー Xmovie . [W³](#)
<http://www.heroinewarrior.com/xmovie.php3>