メディアプログラミング演習―第１５回（第６テーマ４日目）―

３Dグラフィックスの扱い

３次元の滑らかな形状表面（円錐、円柱、トーラス）を，三辺形または四辺形で近似し，各々をワイヤフレーム表示した．さらに，各々の３辺形および四辺形を，平行光線を想定した「明るさ」で塗り潰すことによりレンダリング表示も行った．

今回は、形状として「球」を、また、「点光源」を想定したレンダリングを行う。

演習６－７－１：球を描く

球面上の一点は　パラメータ *u, v* （$0 \leqq u \leqq 2\pi$, $-\pi/2 \leqq v \leqq \pi/2$）を使えば,

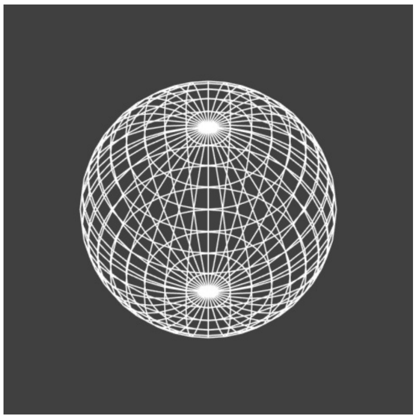
P(u,v)=(x(u,v),y(u,v),z(u,v))、ただし、

x(u,v)=r cos v cos u , *y(u,v)=r cos v sin u, z(u,v)=r sin v*

と表現される．したがって、そのワイヤフレーム表示は

```
for(float u=0;u<360.0;u+=d){
      for(float v=-90;v<90.0;v+=d){
        // P(u,v), P(u+d,v), P(u+d,v+d), P(u,v+d)の４点を求め
        // これらの四辺形を描く
       }
    }
```

となる。



＊md_sphere 内の関数（drawSpherewf() ）完成させ、球をワイヤフレームで表示しなさい。

演習６－７－２：点光源でのレンダリング

平行光線を想定したランバート反射については、第１４回で学んだ。

基本式は、面法線ベクトル *N*、光の方向を指すベクトル *L* の内積として

I=L・N

である。すなわち、光の方向ベクトルと面法線の内積で定まるモデルである。

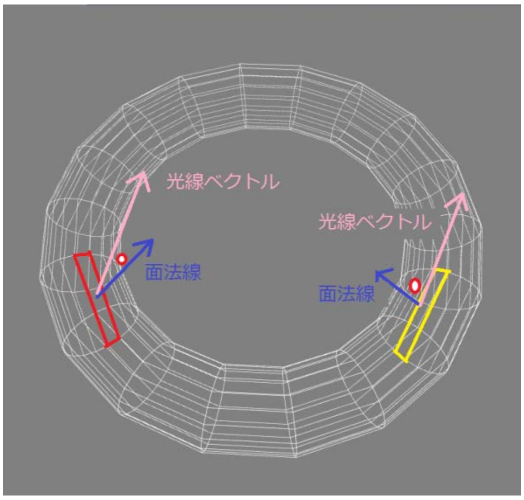
平行光線を想定すると、形状のどの面であっても光の方向ベクトルは一定である（下図、(a)のピンクのベクトル）。一方、点光源を想定すると曲面の位置によって、光の方向は異なる（下図、(b)のピンクのベクトル）。

すなわち、光線ベクトルは一定ではなく、曲面上の点から光源の位置に向かうベクトル　*PL-P* となる。従って、点光源を想定したランバート反射モデルにより輝度は、正

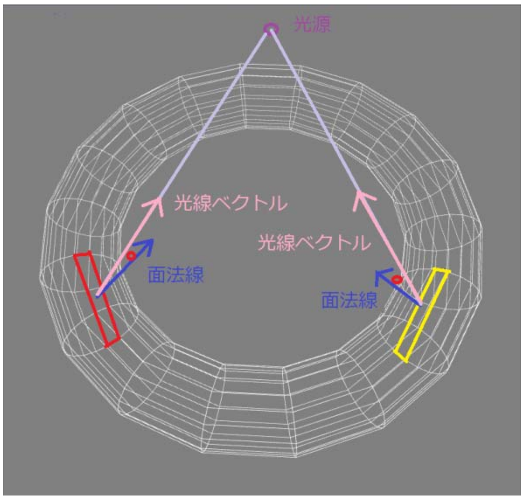
規化された光源方向のベクトルと面法線の内積

$$I=(PL-P)/|PL-P|\cdot N$$

となる。ここで、***PL*** は点光源の位置、***P*** は塗潰すパッチ内の一点とする。



(a) 平行光線のモデル



(b) 点光源のモデル

トーラスの点光源でのレンダリング表示

前回完成させた *drawToruswf()*を作り換え（*drawTorusRend_point()*とする）、また、点光源での明るさを求める関数を ***Bright_point()***として、点光源でレンダリングされたトーラスを生成せよ(***md-tourus-rend-point.txt*** を参照)。

ヒント：たとえば、光源位置を

```
float[] P_Light={0,0,1};
```

のように定義する。この時、

関数 **Bright()**において、

```
br=(N1[0]*Light[0]+N1[1]*Light[1]+N1[2]*Light[2]);
```

のかわりに

```
D1[0]=P_Light[0]-P1[0];D1[1]=P_Light[1]-P1[1];
D1[2]=P_Light[2]-P1[2];
r=sqrt(D1[0]*D1[0]+D1[1]*D1[1]+D1[2]*D1[2]);
D1[0]/=r;D1[1]/=r; D1[2]/=r;
br=(N1[0]*D1[0]+N1[1]*D1[1]+N1[2]*D1[2])*256;
```

(PL-P)/|PL-P| を求める

輝度を求める

とすればよい。

＊光源をトーラスの中心に置きレンダリング表示しなさい。