マルチメディア工学
プログラム演習手引書
(ボリュームレンダリング)

教員　佐藤嘉伸　yoshi@image.med.osaka-u.ac.jp
TA　島田隆次　r-simada@ist.osaka-u.ac.jp

1　演習手順

1. Web ページからサンプルプログラムをダウンロードする．
2. VolumeRendering.c の TODO コメントの部分を埋めてコードを完成させる．
3. パラメータを変化させて生成される CG 画像を比較する．
4. 比較結果を載せたレポートを作成する．

2　プログラム

2.1　サンプルプログラムの入手

マルチメディア工学の Web ページからサンプルプログラム VolumeRenderingSampleCode.zip をダウンロードし，任意のディレクトリに展開する．

ファイル構成：

```
VolumeRenderingSampleCode/
├readme.txt
└src/                    ソースディレクトリ
  ├VolumeRendering.c     超楕円体の生成とレンダリングのメインルーチン
  ├Point3D.h             三次元空間上の一点を表す
  ├Vector3D.c            三次元ベクトルを表す
  ├Vector3D.h                    〃
  ├Quaternion.c          四元数を表す
  ├Quaternion.h                 〃
  ├Image2D.c             二次元ビットマップ画像を表す
  ├Image2D.h                     〃
  ├Image3D.c             三次元ビットマップ画像を表す
  ├Image3D.h                     〃
  ├GradientImage3D.c     3 次元ベクトルを画素値にもつ三次元画像
  ├GradientImage3D.h                〃
  ├Makefile              gcc 用の Makefile
  └Makefile.vc           VC++ 用の Makefile
```

VolumeRendering.c がメインのソースコードである．main 関数では以下の処理をしている．

1. コマンドライン引数を解析
2. makeRotater 関数で回転のための四元数を作成
3. initializeVolumeImage 関数で超楕円体の三次元画像を作成
4. smooth 関数で三次元画像を繰り返し平滑化
5. calcGradientImage 関数で三次元画像の勾配を計算
6. render 関数で三次元画像 (ボリュームデータ) を二次元画像に描画
7. 二次元画像をファイルに保存

同梱されているヘッダファイルには VolumeRendering.c で利用している各関数のプロトタイプ宣言と簡単な説明が書いてあるので，一度目を通しておくこと．

2.2 プログラムの作成

サンプルプログラムは VolumeRendering.c の一部が未完成となっているので，その部分を埋めてプログラムを完成させる．該当箇所はコメントに**「TODO: (番号) コード作成内容」**と書いてある．以下，各番号に対応したコード作成内容について説明する．

2.2.1 適当な不透明度関数を作る

画素値 value(0〜1) から不透明度 (0〜1) を求める不透明度関数を作る．「画素値をそのまま不透明度とする」「画素値が 0.5 以上なら不透明度 1，それ以外なら不透明度 0」など不透明度関数の内容は自由に決めてよい．授業スライド「コンピュータグラフィックス（ボリュームグラフィックス）」の 35 スライド目を参考に．

2.2.2 問題 P2-1 で求めた超楕円体の陰関数

30 スライド目の問題 P2-1 で求めた超楕円体の陰関数表現を C 言語で記述する．陰関数表現は $F(x, y, z) = 1$ の形で表されるが，ここでは $F(x, y, z)$ の部分だけを書けばよい．

2.2.3 q2 = conj(rotater) * q * rotater の計算

回転を表す四元数 rotater で q を逆回転した結果を q2 に求める．通常の回転とは rotater と conj(rotater) が入れ替わっている点に注意．conj(rotater) は事前に c に求めてあるのでそれを利用すればいい．なお conj は共役複素数を表す．四元数同士の乗算は Quaternion_mult 関数を使えばよい．

2.2.4 逆回転後の座標 (v2.x, v2.y, v2.z) における陰関数 (superellipsoid) の値を求め，¡=1 なら 1 を，¿1 なら 0 を value にセット

逆回転後の座標 (v2.x, v2.y, v2.z) と超楕円体の各種パラメータを引数として superellipsoid 関数を呼び出す．その値が 1 以下なら 1 を，1 より大きければ 0 を value に代入する．31 スライド目を参考に．

2.2.5 (x,y,z) における勾配を計算して v にセット

すぐ上にある smooth 関数を参考に，三次元画像 image の座標 (x, y, z) における勾配を v に求める．式が長くなるので，勾配の x 成分，y 成分，z 成分をそれぞれ計算してから v にセットするとよい．値を v にセットするには Vector3D_set マクロを使用すればよい．三次元画像から画素値を取得するには Image3D_getPixel 関数を使用すればよい．33 スライド目を参考に．

2.2.6 dest の計算

原点と (x,y,z) を通る直線を軸として theta だけ回転させるための四元数を dest に求める．授業スライド「コンピュータグラフィックスの基礎（座標系，剛体変換，3 次元回転の数理，投影）」の 23 スライド目に式があるので，それを C 言語で記述すればよい．dest に値を設定するには Quaternion_set マクロを使う．

2.2.7 ボクセル (x,y,z) の画素値を元に不透明度関数を呼び出して alpha に不透明度を得る

三次元画像 image のボクセル (x,y,z) の画素値を取得し，その画素値を引数として不透明度関数 alphaFunc を呼び出せばよい．

2.2.8 光線方向 (lightDir) と勾配 (g) の内積から角度 (陰影係数 cosTheta) を求める

陰影係数 cosTheta を求める．光線方向 lightDir と勾配 (の法線ベクトル)g の内積を求めればよい．ただし lightDir は正規化されている (長さが 1) が，g は正規化されていないので，内積を計算したあと g の長さで割る必要がある．g の長さは Vector3D_length 関数で求めることができる．光線方向と勾配 (の法線) と陰影係数の関係は 34 スライド目の図を参考に．

2.2.9 透過光強度 (transparent)，不透明度 (alpha)，陰影係数 (cosTheta) からボクセル (x,y,z) の寄与を求める式

透過光強度 (transparent)，不透明度 (alpha)，陰影係数 (cosTheta) から，そのボクセルの画素値への寄与を求める．寄与は透過光強度と不透明度と陰影係数の積であるので，それを記述すればよい．授業スライド「コンピュータグラフィックス（ボリュームグラフィックス）」の 34 スライド目の式を参考に．色は最後に掛けるのでここでは不要，単色として扱えばよい．

2.3 コンパイルと実行

サンプルプログラムには Makefile が付属している．gcc を使用している場合は「make」で，VisualC++ を使用している場合は「nmake /f Makefile.vc」でコンパイルすればよい．

コンパイルできたら不透明度関数，超楕円体の 5 つのパラメータ，回転パラメータ，光源の方向，平滑化回数などを変化させて画像を作成し，比較する．画像サイズを大きくすると大量のメモリと長い処理時間が必要になるので，最初は小さい画像サイズ (128 × 128 × 128 程度) で試すのがよい．

コマンドライン引数は以下のとおり．

```
$ VolumeRendering a1 a2 a3 ε1 ε2 rotaterX rotaterY rotaterZ rotaterTheta
    smoothIteration lightDirX lightDirY lightDirZ
    lightColorR lightColorG lightColorB
    imageSizeX imageSizeY imageSizeZ filename

  a1 a2 a3 ε1 ε2: 超楕円体のパラメータ
  rotaterX rotaterY rotaterZ: 原点と (z,y,z) を通る直線を回転軸とする
  rotaterTheta: 回転角
  smoothIteration: 平滑化の繰り返し回数
  lightDirX lightDirY lightDirZ: 平行光源の方向
  lightColorR lightColorG lightColorB: 光源の色
  imageSizeX imageSizeY imageSizeZ: 三次元画像のサイズ
  filename: 二次元画像の保存ファイル名
```