マルチメディア工学３

コンピュータグラフィックス
形状表現、陰影表現

佐藤 嘉伸

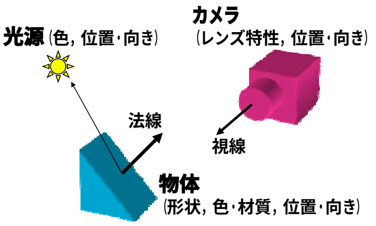
大阪大学 大学院医学系研究科
放射線統合医学講座
yoshi@image.med.osaka-u.ac.jp
http://www.image.med.osaka-u.ac.jp/member/yoshi/
講義ホームページ：日本語ページ → 授業の資料 → マルチメディア工学

ＣＧ (Computer Graphics)

物体形状表現
幾何学 Geometry

ＣＧ：３次元世界表現と表示

- 座標系，座標変換，回転の表現，投影(カメラ) モデル
- **物体形状表現 (幾何学 geometry)　(スプライン関数，超2次関数)**
- 物体陰影表現 (photometry)　（分光反射特性：拡散反射，鏡面反射）

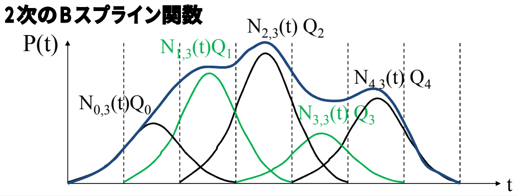


ＣＧ：　物体形状表現
Ｍ－１次のＢスプライン関数:

$$\mathbf{P}(t) = \sum_{j=0}^{N} \boxed{N_{j,M}(t)}\boxed{\mathbf{Q}_j}$$

Ｍ－１次基底関数 (Basis Function)　係数

２次のＢスプライン関数

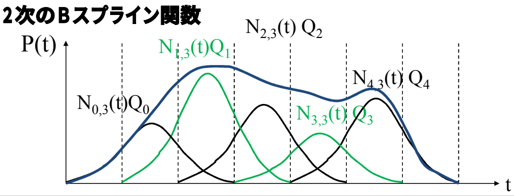


ＣＧ：　物体形状表現
Ｍ－１次のＢスプライン関数:

$$\mathbf{P}(t) = \sum_{j=0}^{N} \boxed{N_{j,M}(t)}\boxed{\mathbf{Q}_j}$$

Ｍ－１次基底関数 (Basis Function)　係数

２次のＢスプライン関数



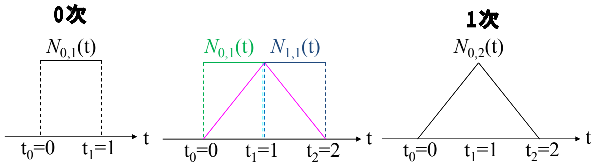
ＣＧ：　物体形状表現
Ｂスプライン関数:　Ｍ－１次 Ｂスプライン基底関数

$$N_{j,M}(t) = \boxed{\frac{t-t_j}{t_{j+M-1}-t_j}}\boxed{N_{j,M-1}(t)} + \boxed{\frac{t_{j+M}-t}{t_{j+M}-t_{j+1}}}\boxed{N_{j+1,M-1}(t)}$$

$$N_{j,1}(t) = \begin{cases} 1 & (t_j \le t < t_{j+1}) \\ 0 & (\text{else}) \end{cases}$$

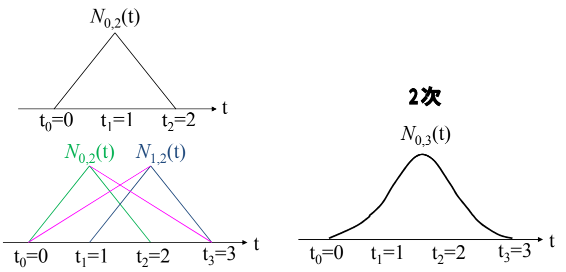
$$\mathbf{T} = [t_0, t_1, \cdots, t_{M+N-1}]$$

０次　**１次**



CG：物体形状表現
Bスプライン関数：M－1次Bスプライン基底関数

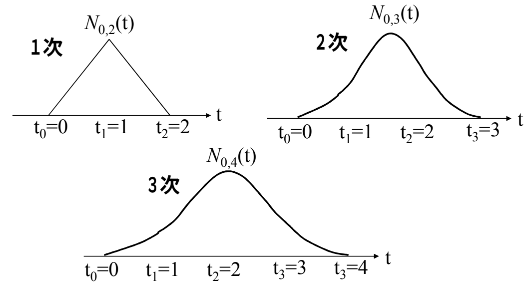
$$N_{j,M}(t) = \boxed{\frac{t-t_j}{t_{j+M-1}-t_j}} \boxed{N_{j,M-1}(t)} + \boxed{\frac{t_{j+M}-t}{t_{j+M}-t_{j+1}}} \boxed{N_{j+1,M-1}(t)}$$



CG：物体形状表現
Bスプライン関数：基底関数

(M－1)次スプライン関数

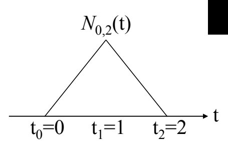
$$\mathbf{P}(t) = \sum_{j=0}^{N} N_{j,M}(t)\mathbf{Q}_j$$



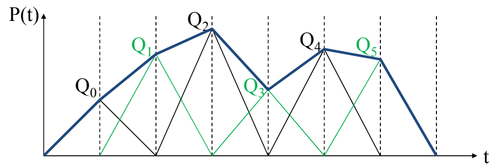
CG：物体形状表現
Bスプライン関数：1次の場合

1次スプライン関数

$$\mathbf{P}(t) = \sum_{j=0}^{N} N_{j,M}(t)\mathbf{Q}_j$$



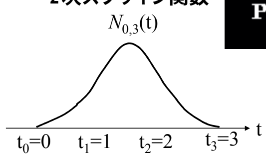
1次スプライン関数 ＝C^0連続



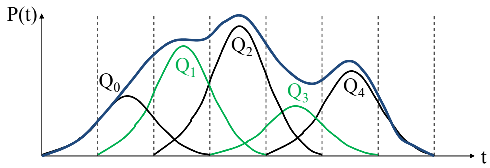
CG：物体形状表現
Bスプライン関数：2次の場合

2次スプライン関数

$$\mathbf{P}(t) = \sum_{j=0}^{N} N_{j,M}(t)\mathbf{Q}_j$$



2次スプライン関数 ＝C^1連続 (1次微分連続)



CG：Bスプライン関数

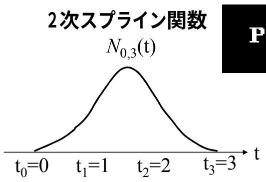
$$N_{j,M}(t) = \frac{t-t_j}{t_{j+M-1}-t_j}N_{j,M-1}(t) + \frac{t_{j+M}-t}{t_{j+M}-t_{j+1}}N_{j+1,M-1}(t)$$

$$N_{j,1}(t) = \begin{cases} 1 & (t_j \le t < t_{j+1}) \\ 0 & (\text{else}) \end{cases}$$

$$\mathbf{T} = [t_0, t_1, \cdots, t_{M+N-1}]$$

2次スプライン関数

$$\mathbf{P}(t) = \sum_{j=0}^{N} N_{j,M}(t)\mathbf{Q}_j$$



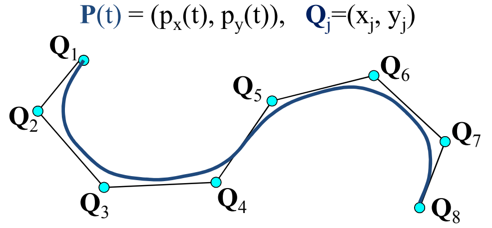
2次スプライン関数 ＝C^1連続 (1次微分連続)

ポイント：n＋1次Bスプライン関数がC^n連続である。

CG：物体形状表現
Bスプライン関数：曲線の表現 (パラメトリック曲線)

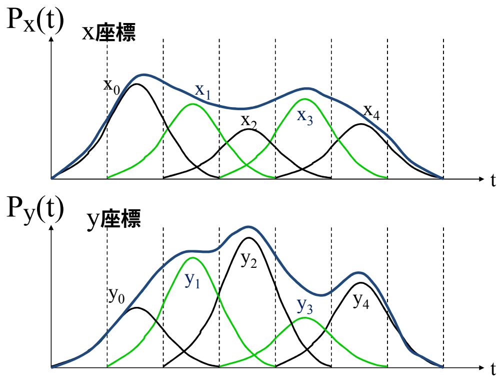
$$\mathbf{P}(t) = \sum_{j=0}^{N} N_{j,M}(t)\mathbf{Q}_j$$

$\mathbf{P}(t) = (p_x(t), p_y(t))$,　$\mathbf{Q}_j = (x_j, y_j)$



2次元空間にあるので、2つの座標値の表現式が必要であり、曲線は1次元的(1次元多様体)であることから、各表現式は、1つのパラメータ t で記述される。

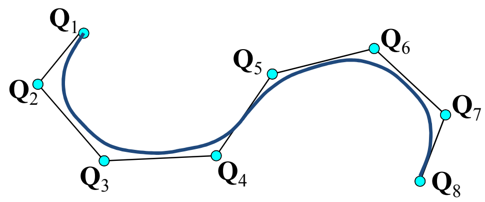
ＣＧ：物体形状表現
Ｂスプライン関数：２次の場合



ＣＧ：物体形状表現
Ｂスプライン関数：曲線の表現（パラメトリック曲線）

$$\mathbf{P}(t) = \sum_{j=0}^{N} N_{j,M}(t)\mathbf{Q}_j$$

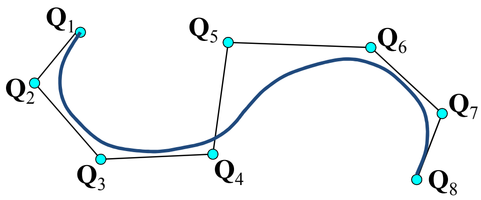
$\mathbf{P}(t) = (p_x(t), p_y(t))$,　$\mathbf{Q}_j = (x_j, y_j)$



ＣＧ：物体形状表現
Ｂスプライン関数：曲線の表現（パラメトリック曲線）

$$\mathbf{P}(t) = \sum_{j=0}^{N} N_{j,M}(t)\mathbf{Q}_j$$

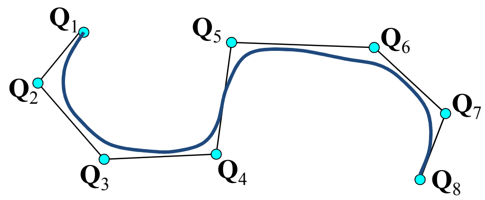
$\mathbf{P}(t) = (p_x(t), p_y(t))$,　$\mathbf{Q}_j = (x_j, y_j)$



ＣＧ：物体形状表現
Ｂスプライン関数：曲線の表現（パラメトリック曲線）

$$\mathbf{P}(t) = \sum_{j=0}^{N} N_{j,M}(t)\mathbf{Q}_j$$

$\mathbf{P}(t) = (p_x(t), p_y(t))$,　$\mathbf{Q}_j = (x_j, y_j)$

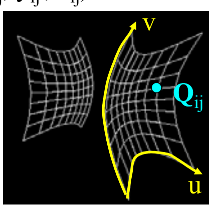


ＣＧ：物体形状表現
Ｂスプライン関数：曲面の表現（テンソル積曲面）

$\mathbf{P}(u,v) = \sum \sum N_{i,M}(u)\, N_{j,M}(v)\, \mathbf{Q}_{ij}$

$\mathbf{P}(u,v) = (p_x(u,v), p_y(u,v), p_z(u,v))$,

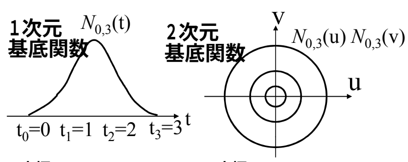
$\mathbf{Q}_{ij} = (x_{ij}, y_{ij}, z_{ij})$

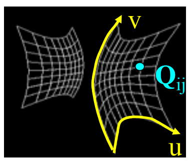


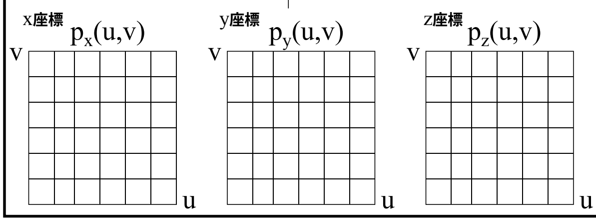
3次元空間中にあるので、3つの座標値の表現式が必要とあり、曲面は2次元的（2次元多様体）であることから、各表現式は、2つのパラメータ u,v で記述される。

Ｂスプライン関数：曲面の表現（テンソル積曲面）

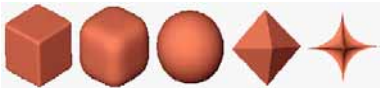
$\mathbf{P}(u,v) = (p_x(u,v), p_y(u,v), p_z(u,v))$



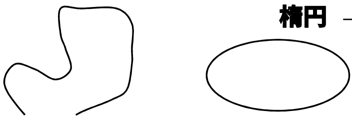




ＣＧ：物体形状表現
超2次関数 (Superquadrics)：超楕円体



楕円 → 超楕円



自由形状 vs 解析関数

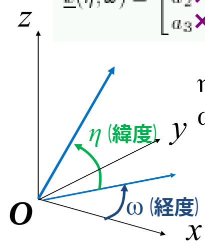
形状表現の多様さ vs 形状制御の容易さ

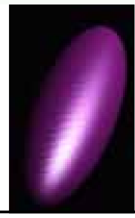
ＣＧ：物体形状表現
2次関数 (Quadrics)：楕円体

$$\left(\frac{x_1}{a_1}\right)^2 + \left(\frac{x_2}{a_2}\right)^2 + \left(\frac{x_3}{a_3}\right)^2 = 1$$

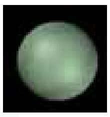
$$\underline{x}(\eta,\omega) = \begin{bmatrix} a_1 \times \cos\eta \times \cos\omega \\ a_2 \times \cos\eta \times \sin\omega \\ a_3 \times \sin\eta \end{bmatrix}, \quad \begin{array}{ccccc} -\pi/2 & \le & \eta & \le & \pi/2 \\ -\pi & \le & \omega & < & \pi \end{array}$$

η：緯度
ω：経度





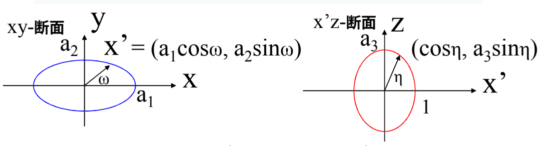
ＣＧ：物体形状表現
Spherical Products



円 $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r\cos\theta \\ r\sin\theta \end{pmatrix}$ 　楕円 $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a\cos\theta \\ b\sin\theta \end{pmatrix}$

$$\left(\frac{x_1}{a_1}\right)^2 + \left(\frac{x_2}{a_2}\right)^2 + \left(\frac{x_3}{a_3}\right)^2 = 1$$

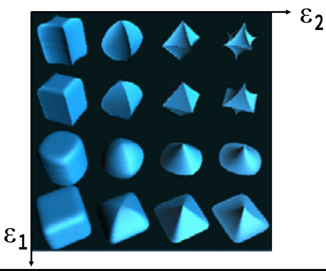
$$\underline{x}(\eta,\omega) = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \begin{bmatrix} \cos\eta \\ \cos\eta \\ \sin\eta \end{bmatrix} \begin{bmatrix} \cos\omega \\ \sin\omega \end{bmatrix}, \quad \begin{array}{ccccc} -\pi/2 & \le & \eta & \le & \pi/2 \\ -\pi & \le & \omega & < & \pi \end{array}$$



cosη ($a_1$cosω, $a_2$sinω, 0) + $a_3$sinη(0,0,1)

ＣＧ：物体形状表現
超2次関数 (Superquadrics)：超楕円体

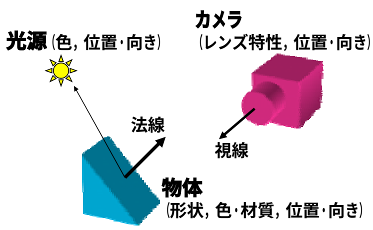
$$\underline{x}(\eta,\omega) = \begin{bmatrix} a_1 \times \cos^{\varepsilon_1}\eta \times \cos^{\varepsilon_2}\omega \\ a_2 \times \cos^{\varepsilon_1}\eta \times \sin^{\varepsilon_2}\omega \\ a_3 \times \sin^{\varepsilon_1}\eta \end{bmatrix}, \quad \begin{array}{ccccc} -\pi/2 & \le & \eta & \le & \pi/2 \\ -\pi & \le & \omega & < & \pi \end{array}$$

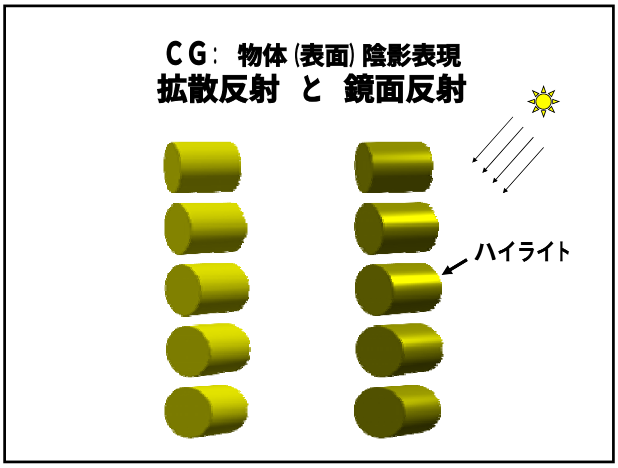


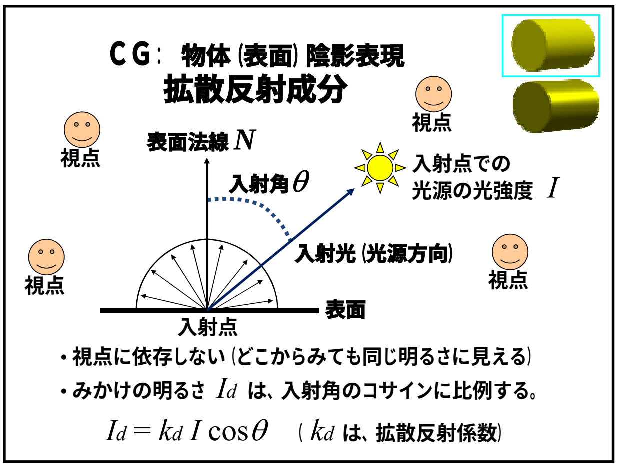
ＣＧ (Computer Graphics)
物体 (表面) 陰影表現
測光学 Photometry

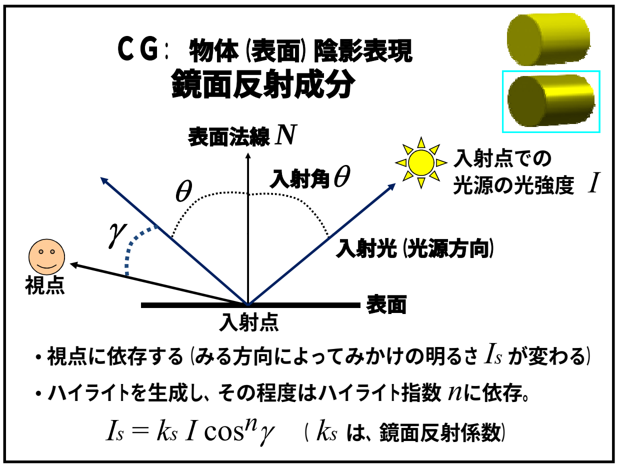
ＣＧ：3次元世界表現と表示

- 座標系，座標変換，回転の表現，投影(カメラ) モデル
- 物体形状表現 (geometry)　(スプライン関数，超2次関数)
- 物体 (表面) 陰影表現 (測光学 photometry)
  (分光反射特性：拡散反射，鏡面反射)









ＣＧ (Compuer Graphics)

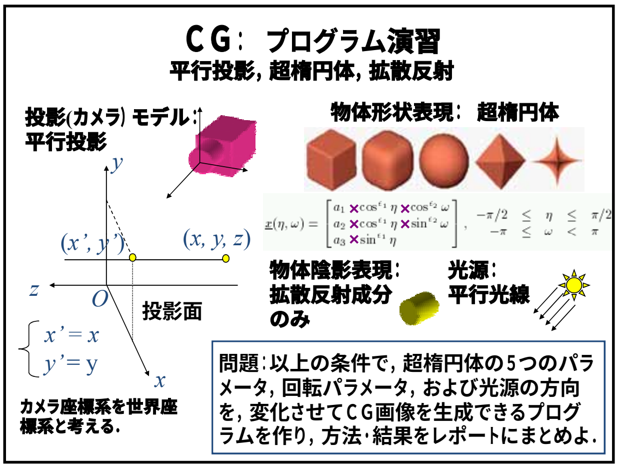
プログラム演習

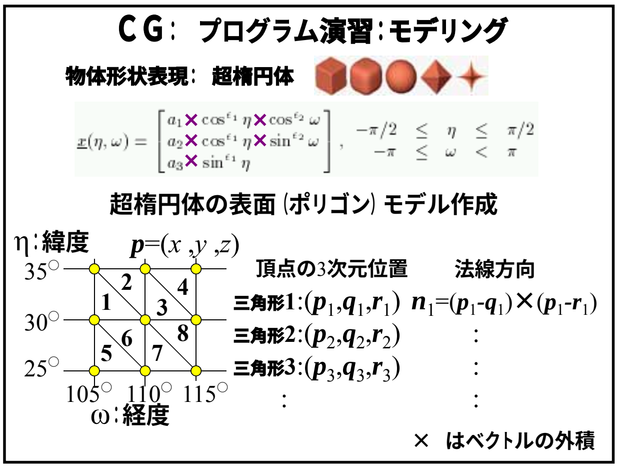
平行投影，超楕円体，回転，拡散反射

http://www.image.med.osaka-u.ac.jp/member/yoshi/

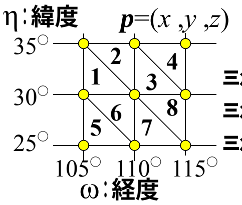
講義ホームページ：日本語ページ → 授業の資料 → マルチメディア工学

- サンプルコードを講義ホームページからダウンロードすること。
- サンプルコード中の　/***** TODO: 処理内容の記述 *****/ の部分のコードをかいて、デバッグ、実行すること。
- もちろん、すでにコードが提供されている部分も、自分で独自に新しく書きなおしてよいし、プログラム全体を自分で独自にＣ以外のＧＵＩを作りやすい言語で書きなおしてもよい。





CG: プログラム演習:レンダリング



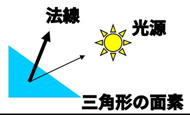
$\boldsymbol{p}=(x,y,z)$

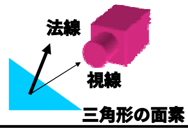
頂点の3次元位置	法線方向
三角形1:$(\boldsymbol{p}_1,\boldsymbol{q}_1,\boldsymbol{r}_1)$	$\boldsymbol{n}_1=(\boldsymbol{p}_1-\boldsymbol{q}_1)\times(\boldsymbol{p}_1-\boldsymbol{r}_1)$
三角形2:$(\boldsymbol{p}_2,\boldsymbol{q}_2,\boldsymbol{r}_2)$	:
三角形3:$(\boldsymbol{p}_3,\boldsymbol{q}_3,\boldsymbol{r}_3)$	:
:	:

超楕円体表面モデルの各三角形(ポリゴン)の輝度値決定

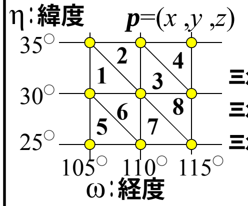
拡散反射の輝度値: 法線方向・光源方向の値

可視/不可視の判断: 法線方向・視線方向の符号(「・」はベクトルの内積)





CG: プログラム演習:レンダリング



$\boldsymbol{p}=(x,y,z)$

頂点の3次元位置	法線方向
三角形1:$(\boldsymbol{p}_1,\boldsymbol{q}_1,\boldsymbol{r}_1)$	$\boldsymbol{n}_1=(\boldsymbol{p}_1-\boldsymbol{q}_1)\times(\boldsymbol{p}_1-\boldsymbol{r}_1)$
三角形2:$(\boldsymbol{p}_2,\boldsymbol{q}_2,\boldsymbol{r}_2)$	:
三角形3:$(\boldsymbol{p}_3,\boldsymbol{q}_3,\boldsymbol{r}_3)$	:
:	:

各画素値の輝度値決定

平行投影は,z座標を無視して,x, y座標のみ取り出せばよい. よって, 各三角形は, 2次元三角形とみなして, 描画すればよい. (画像上の三角形塗りつぶしプログラムは提供されているはずである。)

可視化画像は, gif 形式などで画像ファイルとして保存し表示せよ. (ファイル保存の部分は、サンプルコードを提供されているはずある。)

実験項目および考察事項

- 超楕円体のパラメータ, 回転パラメータを変化させてCG画像を生成せよ.
- 光源の方向を変化させよ.
- 緯度と経度の刻み幅を変化させよ.
- その他, 各自, 創意工夫を加えて実験を行うこと.
- 以上の実験結果に考察を加えよ.
- プログラムが得意な学生は、使いやすいGUIも作成せよ。また、次に示す発展課題にも取り組むこと。

発展課題:陰面消去

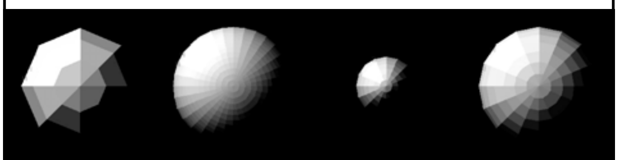
- 超楕円体がただ1つ存在し、かつ、凸形状の場合は、(法線方向と視線方向の内積判定により)視線上の可視表面は一意に定まるが、複数の超楕円体モデルが存在する場合、あるいは、凹表面の箇所がある場合、視線上に2つ以上の可視表面が存在する可能性があり、可視表面のみを描画(不可視表面(陰面)を消去する必要がある。
- 複数の超楕円体を生成して、陰面消去を行って描画せよ。陰面消去アルゴリズムとしては、"Zバッファ法"がある。以下のウェブページを参考にせよ。
  - http://chiyo.sfc.keio.ac.jp/cgsoft/Release/Textbook/zbuffer03.html
  - http://markun.cs.shinshu-u.ac.jp/learn/cg/cg10/index4.html

発展課題:運動生成

- 同じ形状パラメータで、異なる2つの回転(姿勢)をもつ超楕円体の画像をそれぞれ生成せよ。
- 一方の姿勢から他方の姿勢に、超楕円体の姿勢がなめらかに変化するように、アニメーション画像系列(2つの姿勢の中間姿勢を持つ複数毎の画像)を生成せよ。
  - 単位四元数により回転を表現し、四元数の補間式を用いて、2つの回転の間を、回転角度差5度程度になるよう等角度間隔サンプリングを行う。角度差は、単位四元数の内積のArcCosにより求められる。
  - 必要なら、単位四元数表現をマトリックス表現に変換して、画像生成を行う。
- 四元数に基づく補間とオイラー角に基づく補間の両方の画像系列を生成し、どのような違いがあるか調べよ。

Surface Rendering 出力例

形状パラメータ　形状パラメータ　光源方向



緯度と経度の刻み幅　形状サイズ　回転軸