

Computer Graphics

4. Shading

Numazu College of Technology
Dept. of Computer & Control Engineering
CAD/CAM・CG Laboratory

Version 1.1 2006.03.31

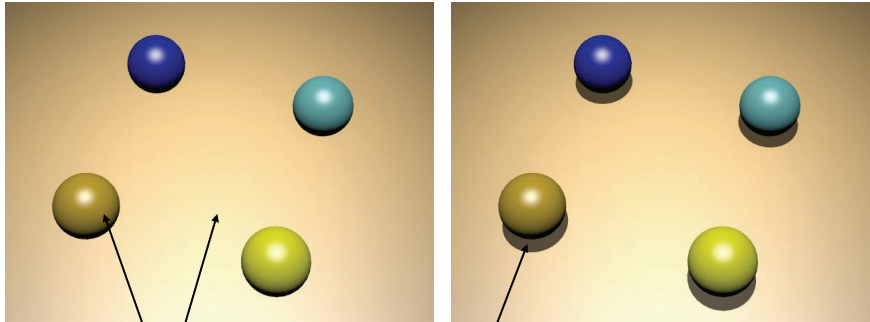
4. シェーディング(Shading)

- 4. 1 Shading
- 4. 2 レンダリング方程式(大域的照明モデル)
- 4. 3 局所的照明モデル(local shading model)
- 4. 4 補間シェーディング

Shading(濃淡づけ, 陰影付け)とは

光源と面の法線ベクトルとの関係から面の明るさを決定する計算

Shadingの例 1枚の平面の上に球を4個おいた場合



平面・球に濃淡が付いている

影づけ(Shadowning)を含めてShadingと呼ぶ

レンダリングとは

CG等を用いてシーン(絵)を作り出すこと

レンダリングの動向

Photo-Realistic -> Animation -> Motion -> Fusion

写真

アニメ

人の動き

実写との融合

レンダリングに関する技術

- ・Lighting (Illumination model : 照明モデル)
 - ・Global Illumination model(大域的照明モデル)
 - ・Local Illumination model(局所的照明モデル)
- ・Shading (Rendering Equation:レンダリング方程式)
 - ・Flat shading(コンスタントシェーディング)
 - ・Smooth shading(局所的照明モデル)
 - ・Gouraud shading
 - ・Phong shading
- ・Shadowing
- ・Mapping

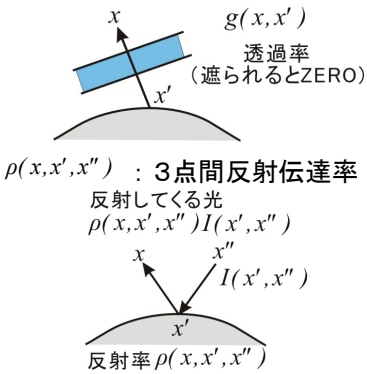
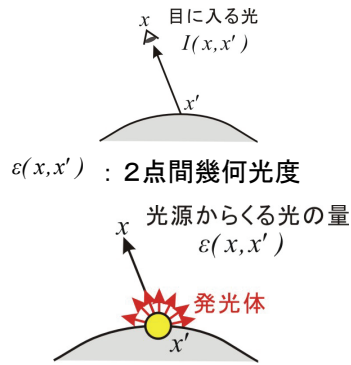
4.2 レンダリング方程式

大域的照明モデルのレンダリング方程式

$$I(x, x') = \underbrace{g(x, x')}_{\text{透過率}} [\underbrace{\varepsilon(x, x')}_{\text{発光}} + \underbrace{\int \rho(x, x', x'') I(x', x'') dx''}_{\text{世界中 反射光}}]$$

$I(x, x')$: 2点間伝達光度

$g(x, x')$: 2点間発光伝達率



$\rho(x, x', x'')$: 3点間反射伝達率

4.3 局所的照明モデル

- 4.3.1 局所的照明モデル
- 4.3.2 拡散反射(乱反射)
- 4.3.3 鏡面反射
- 4.3.4 環境光
- 4.3.5 透過光

4.3.1 局所的照明モデル

局所的照明モデルのレンダリング方程式
(シェーディングモデル)

$$L_i = L_{di} + L_{ri} + L_{ci} + L_t$$

ただし, $L_i(L_{ir}, L_{ig}, L_{ib})$

L_i :任意点の明るさ

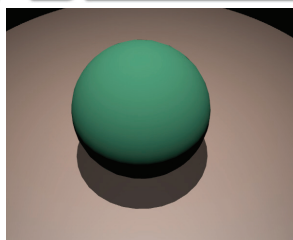
L_{di} :拡散反射(乱反射、散乱)による輝度
Diffuse reflection

L_{ri} :鏡面反射(正反射)による輝度
Specular reflection

L_{ci} :環境光による輝度
Ambient light

L_t :透過による輝度
Transparency

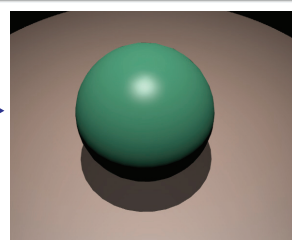
反射の例



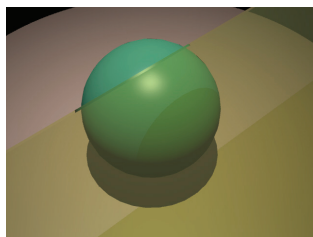
拡散反射のみ



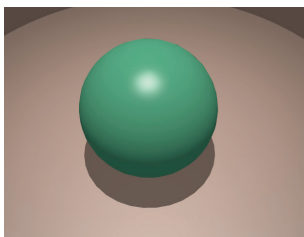
鏡面反射付加



透過追加

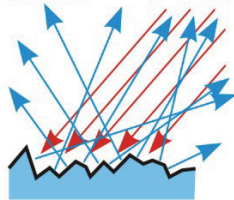


環境光強化



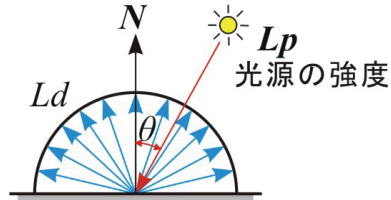
4.3.2 拡散反射(乱反射:Lambert model)

反射光 入射光



拡散反射

単位法線ベクトル



拡散反射のモデル (Lambert model)

$$L_{di} = K_{di} \cdot L_{pi} \cdot \cos \theta$$

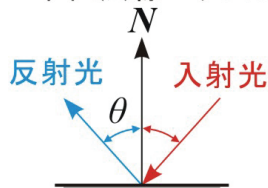
L_{di} : 拡散反射による輝度
 L_{pi} : 光源の強度

K_{di} : 拡散反射係数(0~1)
 θ : 面の法線と入射光のなす角度

拡散反射光は、見る方向によって明るさは変わらず、明るさは面の法線ベクトルと光の入射角によって決まる。その明るさは、入射角の余弦に比例する

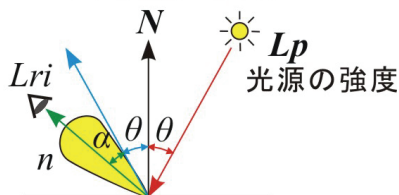
4.3.3 鏡面反射(Phong model)

単位法線ベクトル



鏡面反射

単位法線ベクトル



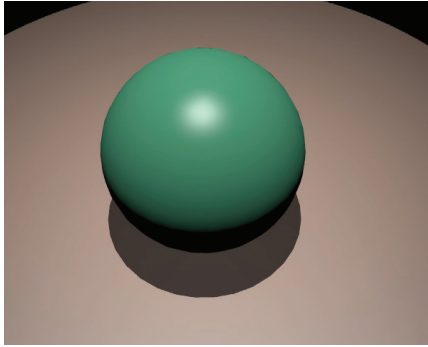
鏡面反射のモデル(Phong model)

$$L_{ri} = K_{ri} \cdot L_{pi} \cdot \cos^n \alpha$$

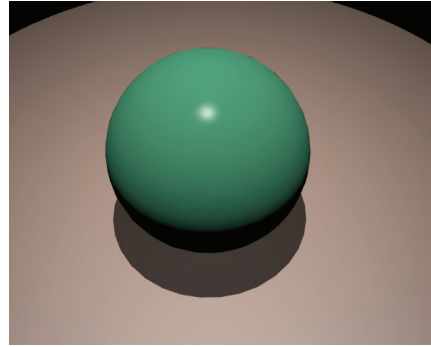
L_{ri} : 鏡面反射による輝度
 L_{pi} : 光源の強度
 n : 鏡面反射の強度(0~10で大きいほど広がりがシャープ)

K_{ri} : 鏡面反射係数($K_{ri} = f(\theta)$)
 α : 視線と正反射方向のなす角度

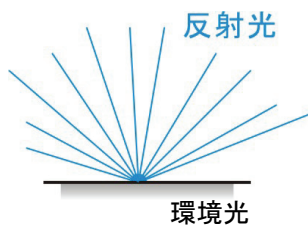
鏡面反射光は、見る方向によって明るさが異なる。明るさは面の法線ベクトル、光の入射角、視線ベクトルによって決まる。プラスチックや金属表面の質感(ハイライト)をあらわす



n:小



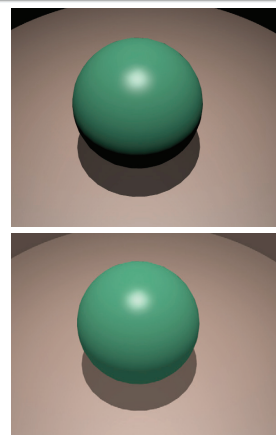
n:大



環境光の例

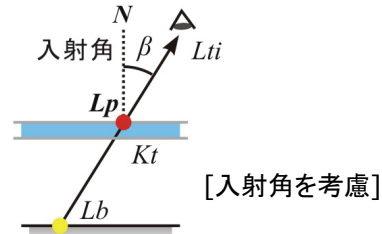
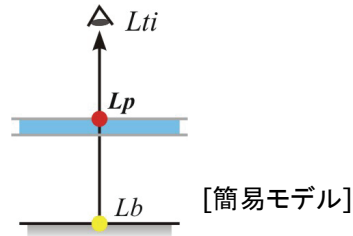
$$L_{ci} = \frac{1}{10}(L_{di} + L_{ri})$$

一般にはパラメータとして直接与える



直接光を受けない面も、2次的な反射光など環境全体によって与えられる明るさによって多少明るく見える。

4.3.5 透過による輝度



$$L_{ti} = K_t \cdot L_{bi} + (1 - K_t) \cdot L_{pi}$$

- L_{ti} : 透過による輝度
- K_t : 透過率(0~1: 1で不透明)
- L_{pi} : 物体表面の輝度
- L_{bi} : 背景の物体の輝度

$$K_t = K_0 - (K_0 - K_1)(1 - \cos \beta)^p$$

- K_0 : $\beta = 0$ 時の透過率(不透明に近い)
- K_1 : $\beta = 90^\circ$ 時の透過率(透明に近い)
- β : 入射角
- p : 透明感強度定数(大で透明感大)

ガラスなど透明感を表現するために透過率を用いて透明を表現する

4.4 補間シェーディング

4.4.1 Flat shading

4.4.2 Gouraud shading

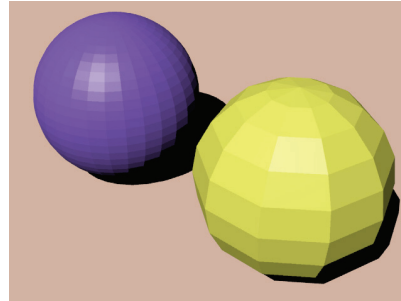
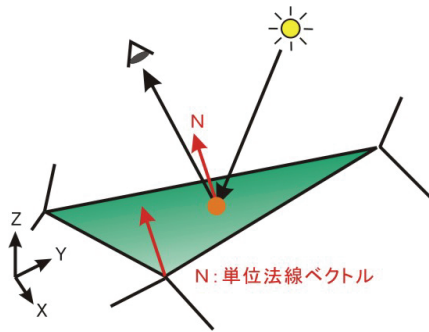
4.4.3 Phong shading

球などの曲面は一般に、ポリゴンに分割されて表現されるが、細かくすると面の数が増えて表示が遅くなる。しかし、荒くすると曲面に見えなくなる。これらの解決の手法が補間シェーディングである。

Shadingの種類

- Flat shading(コンスタントシェーディング)
- Smooth shading(局所的照明モデル)
 - Gouraud shading
 - Phong shading

4.4.1 Flat shading



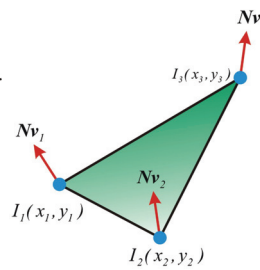
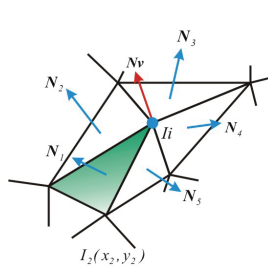
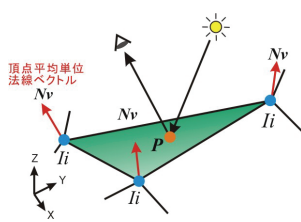
面の輝度を計算する際、常に面の代表法線ベクトルを用いる

フラットシェーディングで表現した球 (面の数が増えても不自然な表示)

フラットシェーディングは高速に描画できるが、マツハバンドがでてしまう。

4.4.2 Gouraud shading(1971年)

P点の輝度 I_p を求める



ひとつの頂点を共有する各面の法線ベクトルを求める

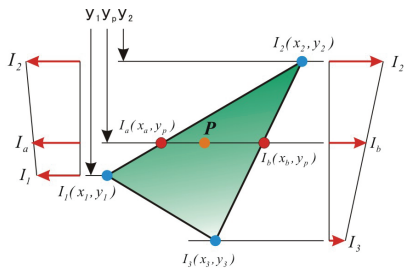
$$N_v = \frac{\sum_{i=1}^m N_i}{m}$$

頂点毎に法線ベクトルを平均化(N_{v1}, N_{v2}, N_{v3})



各頂点における輝度を (I_1, I_2, I_3) 求める

Gouraud shading

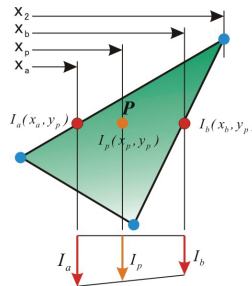


平面座標を2次元に投影

Pを通るスキャンラインと三角形の辺との交点における輝度(I_a, I_b)を線形補間により求める

$$I_a = I_1 \frac{y_p - y_2}{y_1 - y_2} + I_2 \frac{y_1 - y_p}{y_1 - y_2}$$

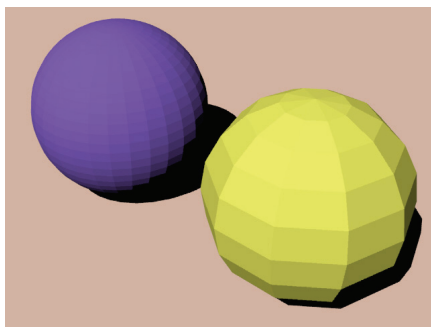
$$I_b = I_1 \frac{y_p - y_3}{y_1 - y_3} + I_3 \frac{y_1 - y_p}{y_1 - y_3}$$



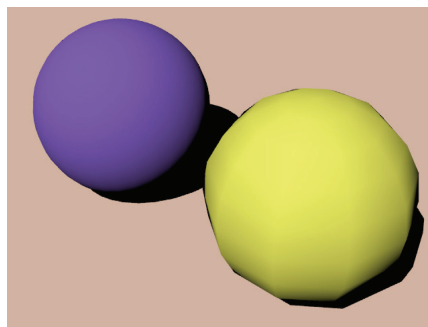
輝度(I_a, I_b)を用いてx方向に線形補間することでPの輝度を求める

$$I_p = I_a \frac{x_b - x_p}{x_b - x_a} + I_b \frac{x_p - x_a}{x_b - x_a}$$

Gouraud shading



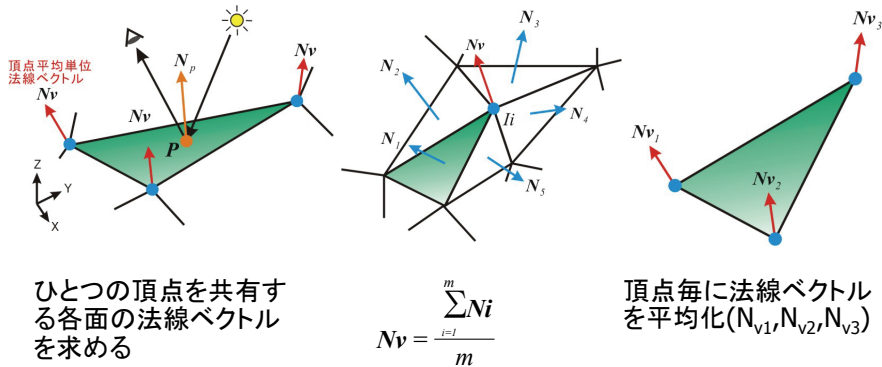
Flat shading



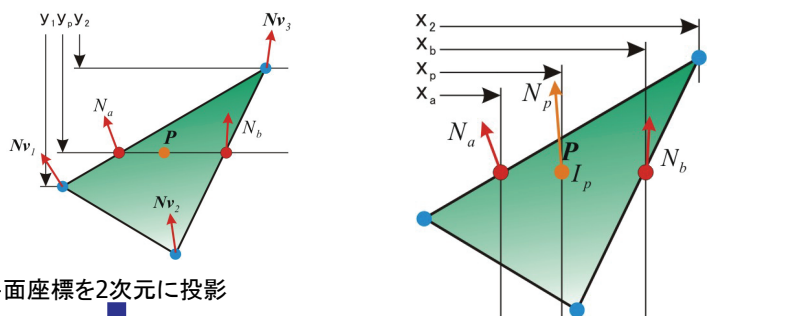
Gouraud shading

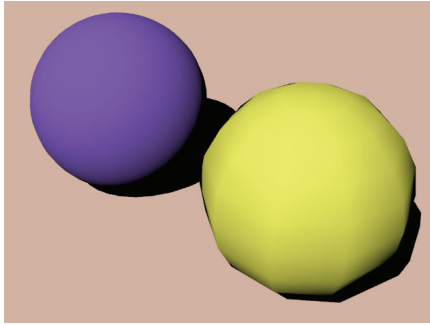
- ・計算が比較的に簡単のため、グラフィックワークステーションではハードウェア化されている。
- ・ハイライトが不自然であるとともに、曲面の輪郭部が丸くならない

4.4.3 Phong shading(1975年)

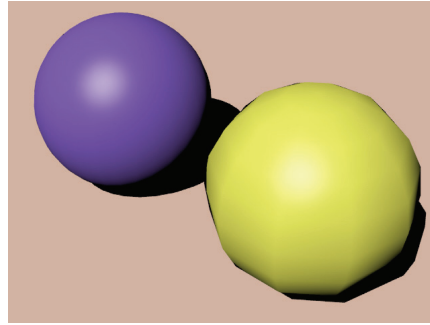


Phong shading





Gouraud shading



Phong shading

- ・計算量が多いため、処理に時間を要する。
- ・自然なハイライトを得ることができる。
- ・曲面の輪郭部が丸くならない