

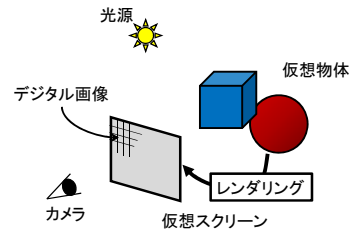
CGにおける画像生成

土橋 宜典

北海道大学大学院情報科学研究科
<http://ime.ist.hokudai.ac.jp/~doba>
doba@ime.ist.hokudai.ac.jp

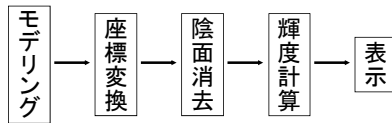
CGとは

- コンピューターを使って作成された画像



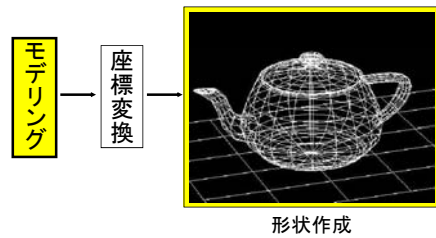
CG画像生成パイプライン

- 画像生成過程



CG画像生成パイプライン

- 画像生成過程



モデリング: 形状表現

- サーフェスモデル

- ポリゴンモデル
多角形の集合で表現
高速表示が可能
基本的に近似表現

- 曲面モデル

スプライン曲面パッチの集合で表現
正確な物体表現が可能
扱いが難しい→表示が遅い



曲線・曲面

- 陰関数表現とパラメトリック表現

$$a_1x^2 + a_2y^2 + a_3z^2 + 2a_4xy + 2a_5yz + 2a_6xz + 2a_7x + 2a_8y + 2a_9z + a_{10} = 0$$

(方程式を解かないと座標がわからない)

$$x = x(u, v), \quad y = y(u, v), \quad z = z(u, v)$$

(パラメータ(u,v)を与えれば座標がわかる)

- パラメトリック表現を用いる方が多い

- 双三次曲面、ベジェ曲面、Bスプライン曲面、NURBS(Non-Uniform Rational B-spline Surface)曲面

曲面・曲面

■ ベジエ(Bezier)曲線

$$x(u) = \sum_{i=0}^n x_i B_i^n(u)$$

$$y(u) = \sum_{i=0}^n y_i B_i^n(u)$$

$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i}$$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$



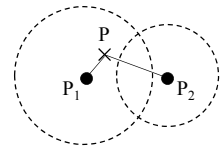
曲線・曲面:陰関数表現

■ メタボール

- 等電位面に似た考え方
- やわらかいものの表現に適する

$$\sum_{k=1}^n f(|P - P_k|) - T = 0$$

を満たす表面



モデリング:その他の形式

■ メタボール

- 等電位面に似た考え方
- やわらかいものの表現に適する



モデリング:その他の形式

■ メタボール

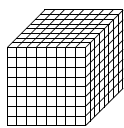
- 等電位面に似た考え方
- やわらかいものの表現に適する



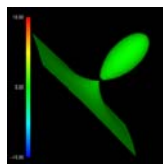
モデリング:その他の形式

■ ボリュームデータ

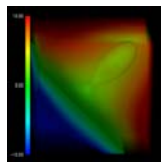
- CT・MRI等の医用画像データ



3次元格子



等値面表示



ボリュームレンダリング

モデリング:データ作成

■ 直接入力

- 基本形状の組み合わせ

■ 形状計測

- 専用装置の利用

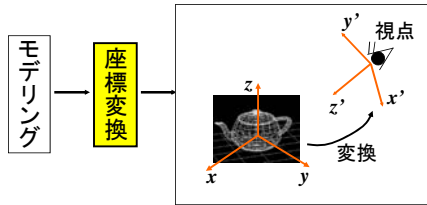
■ 自動生成

- フラクタルなど



CG画像生成パイプライン

- 画像生成過程



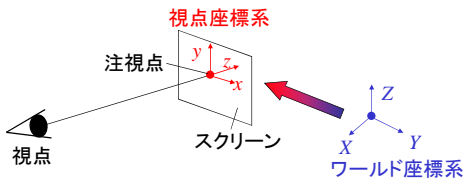
座標変換

- 視野変換
 - ワールド座標系から視点座標系
- クリッピング
 - 視野外の部分の切り取り
- 投影変換
 - 平行投影: z座標を0にするだけ
 - 透視投影: 遠近感を表現
- ビューポート変換

処理の流れ

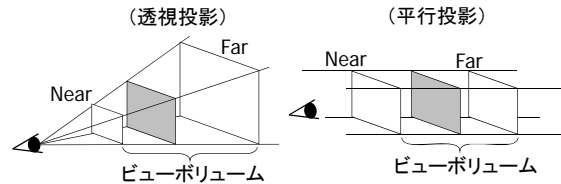
視野変換

- ワールド座標系から視点座標系への変換
- 注視点→視点をz軸とする座標系
- 平行移動・回転によって変換



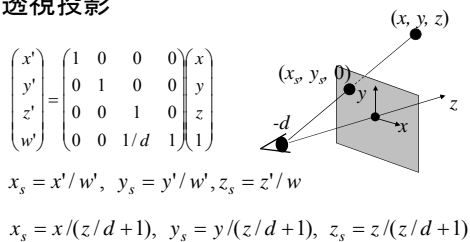
クリッピング

- ビューボリューム外を取り除く
- 正規化座標系への変換(平行移動・拡大/縮小)
 - Far Clipping Plane: $z = -1$
 - Screen: $-1 < x, y < 1$



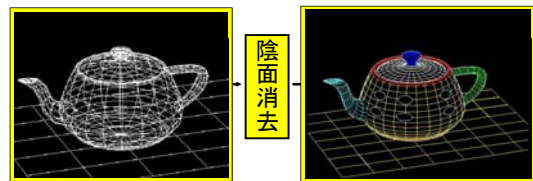
投影変換

- 平行投影: z座標を0にするだけ
- 透視投影



CG画像生成パイプライン

- 画像生成過程

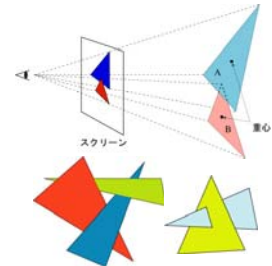


代表的な方法

- Zソート法
- Z-Buffer法
- スキャンライン法
- レイトレーシング法

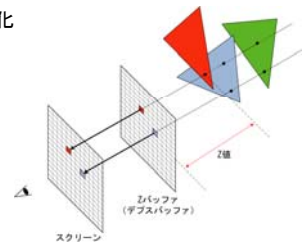
Zソート法

- 全てのポリゴンについて (例えば) 重心位置のZ値を計算
- Z値の大きい順にソート
- 各ポリゴンについて
 - 透視変換
 - 描画 (各画素の内容を上書き)
- **単純だがうまくいかない場合がある**



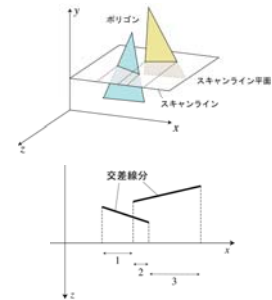
Zバッファ法

- Zバッファを無限遠で初期化
- 各ポリゴンについて
 - 透視変換
 - 各画素(i,j)について
 - Z値の計算
 - $If(z < Z_{buff}(i,j))$
 - 描画
 - $Z_{buff}(i,j) = z$
- ほとんどのハードに搭載
- **ポリゴンならどんな場合もOK**



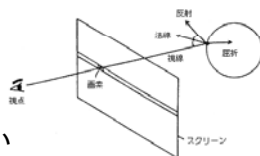
スキャンライン法

- 全てのポリゴンを透視変換
- 各スキャンラインについて
 - スキャンライン平面とポリゴンの交線算出
 - 陰線消去して描画
- 次元の問題に帰着
- **ポリゴンならどんな場合もOK**



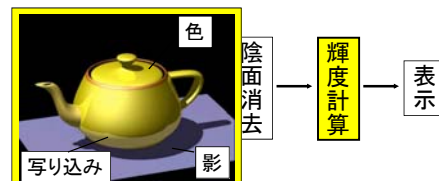
レイトレーシング法

- 各画素について
 - 視線を算出
 - 視線と全物体の交点計算
 - 視点に最も近い交点を算出
 - 描画
- ポリゴンでなくともよい
- 反射/屈折を表現
- **一般には、とても遅い**



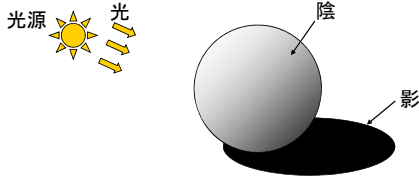
CG画像生成パイプライン

- 画像生成過程



輝度計算

- 陰付け(シェーディング)
- 影付け(シャドウイング)



輝度計算

- 反射モデルと照明モデルに依存
 - 反射モデル(物体表面の属性)
 - 拡散反射・鏡面反射
 - 照明モデル(光源の種類)
 - 点光源・平行光源・線光源・面光源
- 局所照明モデルと大域照明モデル

輝度計算

- 基礎的なシェーディングモデル

$$I = \underbrace{k_a I_{amb}}_{\text{環境光}} + \underbrace{k_d I_l \cos \theta}_{\text{拡散反射光}} + \underbrace{k_s I_l \cos^n \gamma}_{\text{鏡面反射光 (フォンのモデル)}}$$

k_a, k_d, k_s : 係数

反射モデル

- 拡散反射光

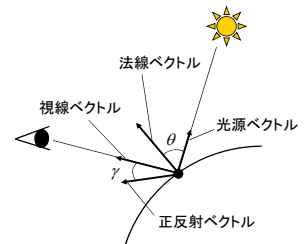
$$I_{out} = \rho_d I_{in} \cos \theta$$

↓
拡散反射率

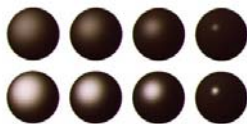
- 鏡面反射光(フォンのモデル)

$$I_{out} = I_{in} w(\theta) \cos^n \gamma$$

↓
鏡面反射率(一定値を用いることが多い)



反射モデル



ハイライトの違い



アルミ 鉄
金属反射のモデル

照明モデルと影付け

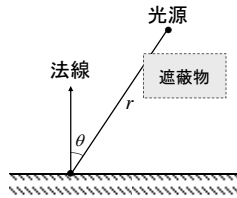
- 点光源
- 平行光源
- 線光源
- 面光源

点光源

- 光を放つ点
- 距離の2乗に比例して光は減衰

$$I_{out} = \frac{\rho_d I_l \cos \theta}{r^2}$$

- 電球
- スポットライト

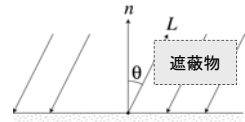


平行光源

- 無限遠にある点光源
- 距離による減衰はない

$$I_{out} = \rho_d I_l \cos \theta$$

- 太陽光

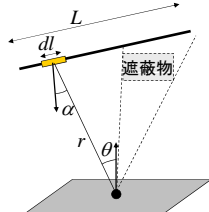


線光源

- 光を放つ線(無限小の点光源の列)
- 線上の微小要素による輝度を積分

$$I_{out} = \rho_d I_l \int_0^L \frac{\cos \theta}{r^2} \cos \alpha dl$$

- 蛍光灯

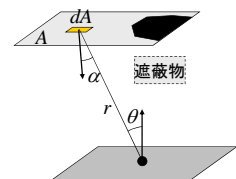


面光源

- 光を放つ面(無限に並んだ線光源)
- 微小面素による輝度を積分

$$I_{out} = \rho_d I_l \int_A \frac{\cos \theta}{r^2} \cos \alpha dA$$

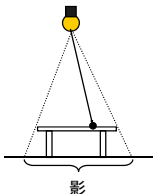
- パネル光源



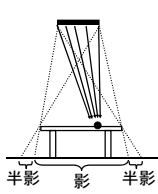
照明モデルと影付け

- 照明モデル

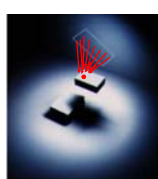
•点光源



•線光源



•面光源



大域的照明モデル

- 相互反射を考慮すること
 - 物体による照り返し。多重反射。
 - 照明・反射モデルを統合した計算が必要
- 二つのアプローチ
 - 有限要素法(ラジオンシティ法)
 - モンテカルロ法

大域的照明モデル

- Rendering Equation

$$\frac{L_{out}(\theta_r, \phi_r)}{\text{反射光}} = \int \frac{f(\theta_{in}, \phi_{in}, \theta_{out}, \phi_{out}) L_{in}(\theta_{in}, \phi_{in}) \cos \theta_{in} d\omega_{in}}{\text{反射分布関数 (BRDF)} \quad \text{入射光}}$$

- 難しさ: 反射光 = 他の点への入射光

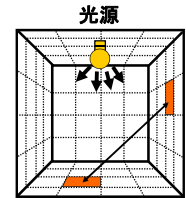
ラジオシティ法

- 拡散反射
- 連立一次方程式

$$B_i = E_i + \rho_j \sum_{j=1, j \neq i}^n F_{ij} B_j$$

$$F_{ij} = \frac{1}{A_i} \int_{A_j} \int_{A_i} \frac{\cos \theta_i \cos \theta_j}{\pi r_{ij}^2} V_{ij} dA_j dA_i$$

B_i : 放射輝度 (ラジオシティ)
 F_{ij} : フォームファクタ
 ρ_j : 拡散反射率
 V_{ij} : パッチ間の可視性を表す関数



計算例

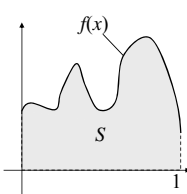


相互反射計算法

- 相互反射とは
 - 物体による照り返し。多重反射。
 - 照明・反射モデルを統合した計算が必要
- 二つのアプローチ
 - 有限要素法 (ラジオシティ法)
 - モンテカルロ法

モンテカルロ法

- 積分計算を行う手法の一つ



$$S = \int_0^1 f(x) dx$$

$$\approx \langle f \rangle$$

$$= \frac{1}{N} \sum_i f(x_i) \quad (0 < x_i < 1)$$

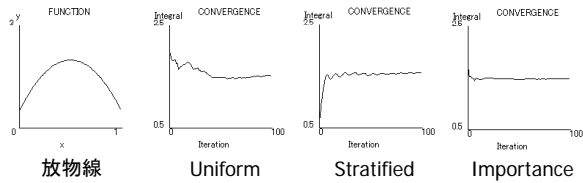
(x_i : 一様乱数)

モンテカルロ法

- Uniform sampling
 - 全区間に対して一様乱数を用いる方法。最も収束性が悪い。
- Stratified sampling
 - 積分区間を有限個の小区間に分割し、各区间ごとに乱数を発生する方法
- Importance sampling
 - 被積分関数の性質を利用して、結果に対する寄与が大きな部分により多くのサンプル点を生成する方法

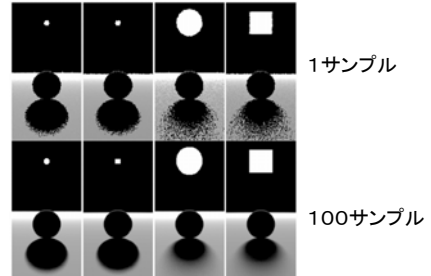
モンテカルロ法

■ 計算例



モンテカルロ法と相互反射

■ 直射光を計算した例



モンテカルロ法と相互反射

■ Rendering Equation

$$\frac{L_{out}(\theta_r, \phi_r)}{\text{反射光}} = \int \frac{f(\theta_{in}, \phi_{in}, \theta_{out}, \phi_{out}) L_{in}(\theta_{in}, \phi_{in})}{\text{反射分布関数 (BRDF)} \quad \text{入射光}} \cos \theta_{in} d\omega_{in}$$

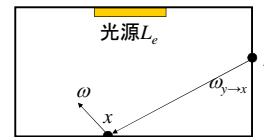
- 一種の積分計算(積分方程式)
- フレドホルム型積分方程式

モンテカルロ法と相互反射

■ フレドホルム(Fredholm)型積分方程式

$$\phi(x) = g(x) + \int_R K(x, y) \phi(y) dy$$

$$L_o(x, \omega) = L_e(x, \omega) + \int_R f_r(\omega, \omega_{y \rightarrow x}) G(x, y) L_o(y, \omega_{y \rightarrow x}) dy$$



モンテカルロ法と相互反射

■ フレドホルム型積分方程式の解

- 以下の無限級数となる

$$\phi(x) = g(x) + I_1(x) + \dots + I_k(x) + \dots$$

$$I_k(x) = \int_R \dots \int_R K(x, x_1) K(x_1, x_2) \dots K(x_{k-1}, x_k) g(x) dx_1 dx_2 \dots dx_k$$

- I_k は光源の光のk回反射光と等価
- 多重積分であり評価が難しい
→モンテカルロ法を利用

モンテカルロ法と相互反射

■ 代表的な方法

- パストレーシング法
- 双方向パストレーシング法
- メトロポリス法



モンテカルロ法と相互反射

- Photon Map法
- 光源からの光を追跡
- フォトンマップとして記憶
- 鏡面反射・コースティクス・相互反射などを統一的に扱えるモデル
- 実装も比較的簡単
- 視点位置に依存しない