

Graphics with Processing



2010-12 レンダリング技術

<http://vilab.org>

塩澤秀和

12.1 隠面消去(1)

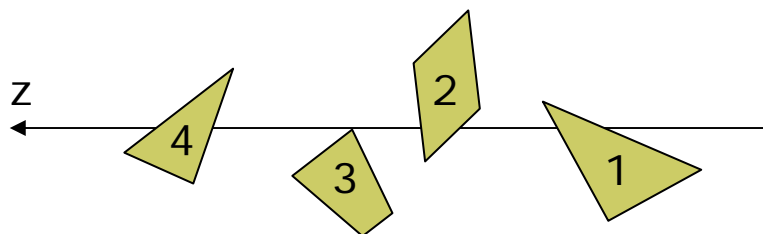
隠面消去(隠線・隠面処理)

□ 隠面消去とは

- 他の物体などに隠れて見えない物体(の全部または一部)を描画しない処理
- 弱点を補い合ういくつかの手法を組み合わせることもある

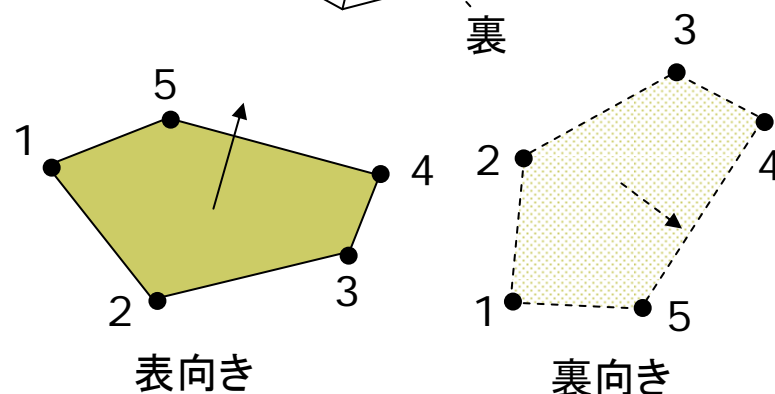
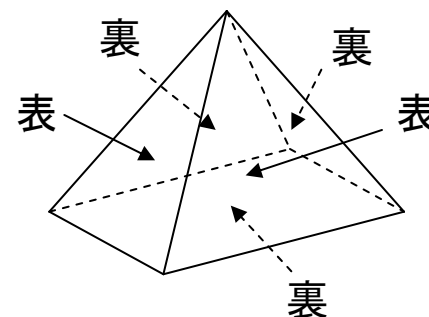
□ 奥行きソート法(p.102)

- ポリゴンをz座標(視点座標)で並び替え, 遠くから順に描画
- 細長いポリゴンで問題が生じる



□ バックフェースカリング(p.100)

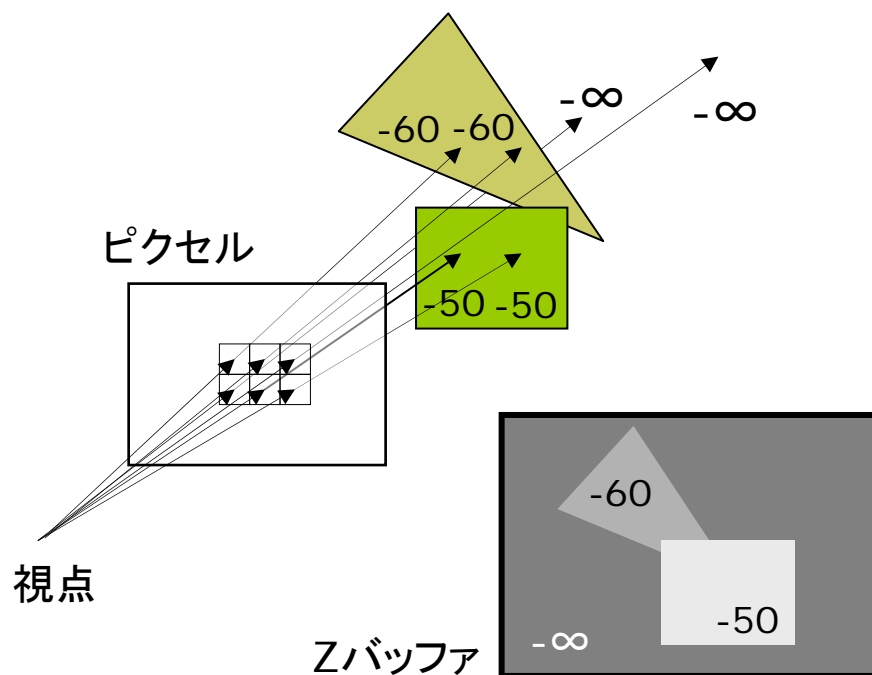
- ポリゴンに表裏を設定し, 裏側を向いているポリゴンを描画しない
- 表裏はポリゴン作成時の頂点の順序(右回り・左回り)で指定
- 1つの凸多面体上の隠面消去



12.2 隠面消去(2)

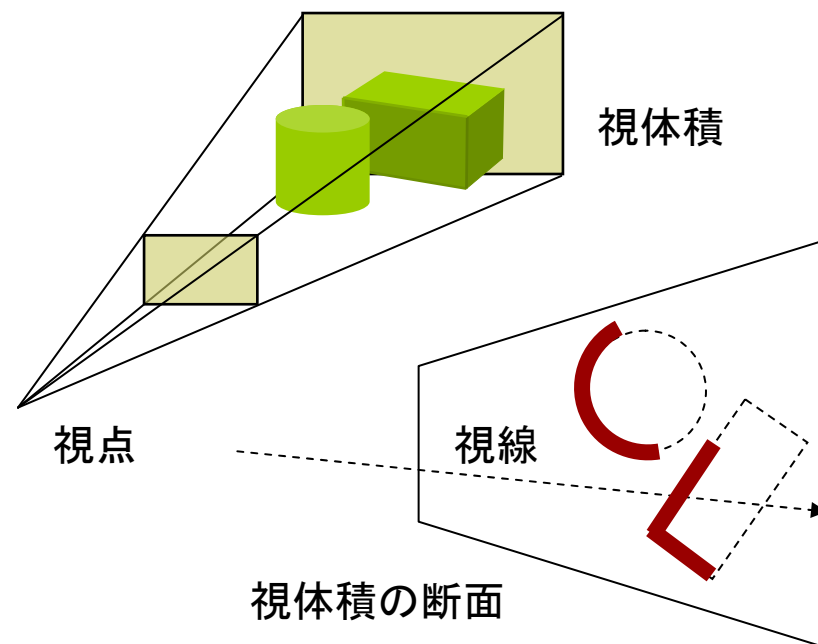
□ Zバッファ法(p.107)

- 画面上の全ピクセルにz座標を持たせ, 1点1点描画するとき遠近関係をチェックする
- 単純&高速 ⇒ ハードウェア化
- 半透明の重なるの処理に難点



□ スキャンライン法(p.105)

- ピクセル横1行(スキャンライン)ごとにポリゴンの断面の重なりを数学的に計算し, 描画する
- 計算は複雑だが, 使用メモリが少ない



12.3 レイトレーシング (p.110)

レイトレーシング法

□ 概要

- Ray Tracing = 光線追跡
- 各ピクセルに届く光の軌跡を、視点から光源にさかのぼるように追跡するレンダリング技術

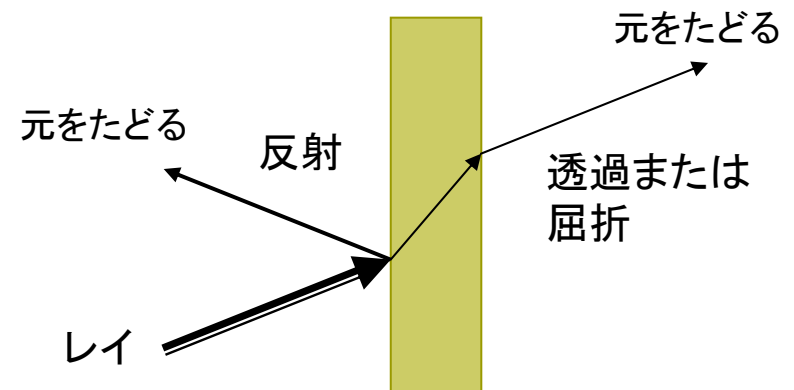
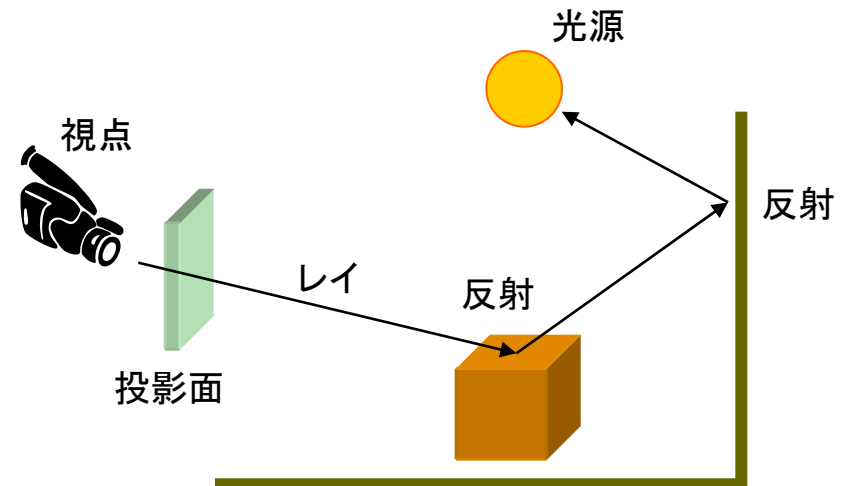
□ 高品質

- 3DCGの初期からあるが、より正しく光学現象を再現するように研究され続けている
- 原理的に隠面消去される
- 透明, 影, レンズも自然に表現

□ 用途

- リアルだが時間がかかるので、まだゲームなどには向かない
- 映像作品(映画)製作で一般的

□ 光線追跡の概念図



12.4 影付け (p.136)

影の種類

- 本影と半影
 - 点光源・平行光線 ⇒ 本影のみ

影付けのアルゴリズム

- レイトレーシング法
 - 原理的に影がつくが遅い
- スキャンライン単位の方法
 - スキャンライン法において、影の区間も計算によって求める
- 投影マッピングによる方法
 - (物体空間における2段階法)
 - **視点を光源**にして、物体のシルエットを描画すると影が分かる
 - 例えば、それを影の画像として、各物体の上にテクスチャ投影

□ シャドウボリューム法

- 各物体が光源からの光をさえぎってできる影の空間を計算し、物体データに「シャドウポリゴン」として付け加える
- レンダリング時に、各ポリゴンと影の交差を判定しながら描画
- 「ステンシルバッファ」を用いると、高速に実現できる

□ シャドウマップ法

- (Zバッファを用いた2段階法)
- 視点を光源にしてZバッファだけを描画すると、光源から光のあたる部分までの距離が分かる
- ピクセルをレンダリングするときに、このシャドウマップも参照し、光のあたっている点だけを描画する

12.5 大域照明モデル(1)

大域照明モデル

□ 間接光の計算

- 環境光モデルでは空間で均一としている間接光を, よりリアルに計算する手法
- 特に室内の陰影がより自然

ラジオシティ法 (p.141)

□ 概要

- Radiosity = 放射発散度
- すべてのポリゴンをパッチ(より小さなポリゴン)に分割する
- 2つのパッチの位置関係から, 相互反射による“光の授受”(フォームファクタ)を計算
- 全パッチ間での光の放射発散の平衡状態を計算する

□ 特徴

- 照明工学の技術の応用
- 輪郭の柔らかい自然な影(ソフトシャドウ)などが表現できる
- 光源が変化しても, フォームファクタは再計算の必要がない

□ ラジオシティ方程式 (p.158)

$$B_i = E_i + \rho_i \sum_{j=1}^n F_{ij} B_j$$

n シーン全体のパッチ数

B_i パッチiのラジオシティ

ρ_i パッチiの反射率

F_{ij} パッチの間のフォームファクタ

- 本質的には「連立一次方程式」
⇒ ガウス・ザイデル法など

12.6 大域照明モデル(2)

フォトンマッピング(p.142)

□ 概要

- Photon=光子(光の粒子)
- 光源から出る大量の光子を考え, その軌跡をシミュレーションする(レイトレーシングとは逆方向)
- すると, シーン全体の間接光の分布が概算できる
- 最後に, 求めた間接光を環境光の代わりにして, 視点からのレイトレーシングを行う

□ 特徴

- レイトレーシング法と相性がよい
- 集光現象(凸レンズや凹面鏡で光が集まる効果)が表現できる
- 着想は簡単だが, アルゴリズムは複雑で, 膨大な時間がかかる

フリーソフトによるレンダリングの例

□ POV-Ray

<http://www.povray.org>
→ Hall of Fame

□ Blender+Yafray

<http://www.blender.org>
→ Feature & Gallery
<http://www.yafaray.org>
→ Gallery

□ Sunflow

<http://sunflow.sourceforge.net>
→ Gallerly

□ Art of Illusion

<http://www.artofillusion.org>
→ Art Gallery

12.7 その他のレンダリング技術

ぼかし(ボケ)系

- CG画像の違和感
 - すべてがはっきりくっきりしすぎ
 - 現実感を出すために、「はっきり見えなくする」ことも必要
- アンチエイリアシング
 - ドットのギザギザが目立たないように、輪郭を中間色でぼかす
- フォグ(霧)
 - 水蒸気やチリなどによる空気の「濁り」を再現する
 - 遠くにあるものがかすんでいき、色が落ちていく効果を与える
- 被写界深度(DOF)
 - レンズの効果を実況し、ピントが合っていないところをぼかす

□ モーションブラー

- 速く動くものに見える残像(ボケ)をわざと表示する
- 軌跡の画像を重ね合わせる

ノンフォトリアリスティック(非写實的)レンダリング

□ 概要

- 現実のマネではないレンダリング
- 例) 油絵風, 手書きタッチの再現, 製図風, 2次元アニメ, 芸術作品

□ 背景

- 既に, 写實的(フォトリアリスティック)なCG技術はかなり完成
- 芸術などへのCG利用の広がり
- アニメーション製作への利用

12.8 演習

Processingでレイトレーシング

- P5Sunflowのインストール
 - p5sunflow-fix.zip を講義ページからダウンロードし, 展開する
 - p5sunflow という名前のフォルダを選び, Processingのlibrariesフォルダの下にコピーする
 - さらにその下にlibraryができる
 - <http://hipstersinc.com/p5sunflow/>
- 使用例

```
import hipstersinc.*;
void setup() {
  size(300, 200,
    "hipstersinc.P5Sunflow");
  noLoop();
}
```

まとめテスト(7/6)

- 日時
 - 7月6日(火) 後半40分程度
 - 欠席者、低得点者は追試
⇒ 7/13, 7/20
- 形式
 - **配布したA4用紙のみ持込可**
(手書きで記入し, 必ず回収)
 - **ただし、追試は持込不可**
- 内容
 - CGにおけるコンピュータ処理の
手順と概要について
 - 基本的な用語と意味
 - 基本的な公式と計算
 - プログラミングそのものは出ない