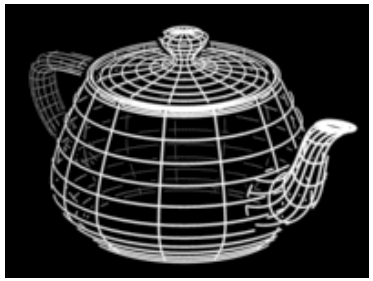
コンピュータグラフィックス基礎

第8回 レンダリング (1)

3DCG表示



モデリング

レンダリング

- 対象物を計算機内で表現する
  - 形の定義
  - 表面の材質
  - 光源

- 対象物をディスプレイに表示する
  - 投影（座標変換）
  - 照明（反射・屈折の計算）

今回の主なテーマ

学習の目標

- 3次元立体を画面に描画するための、様々なアプローチを概観する
- レイトレーシングの原理を理解する
- 物体表面での光の反射モデルを理解する
- レイトレーシングを用いて球体を表示するプログラムを開発できるようになる

レンダリングとは？

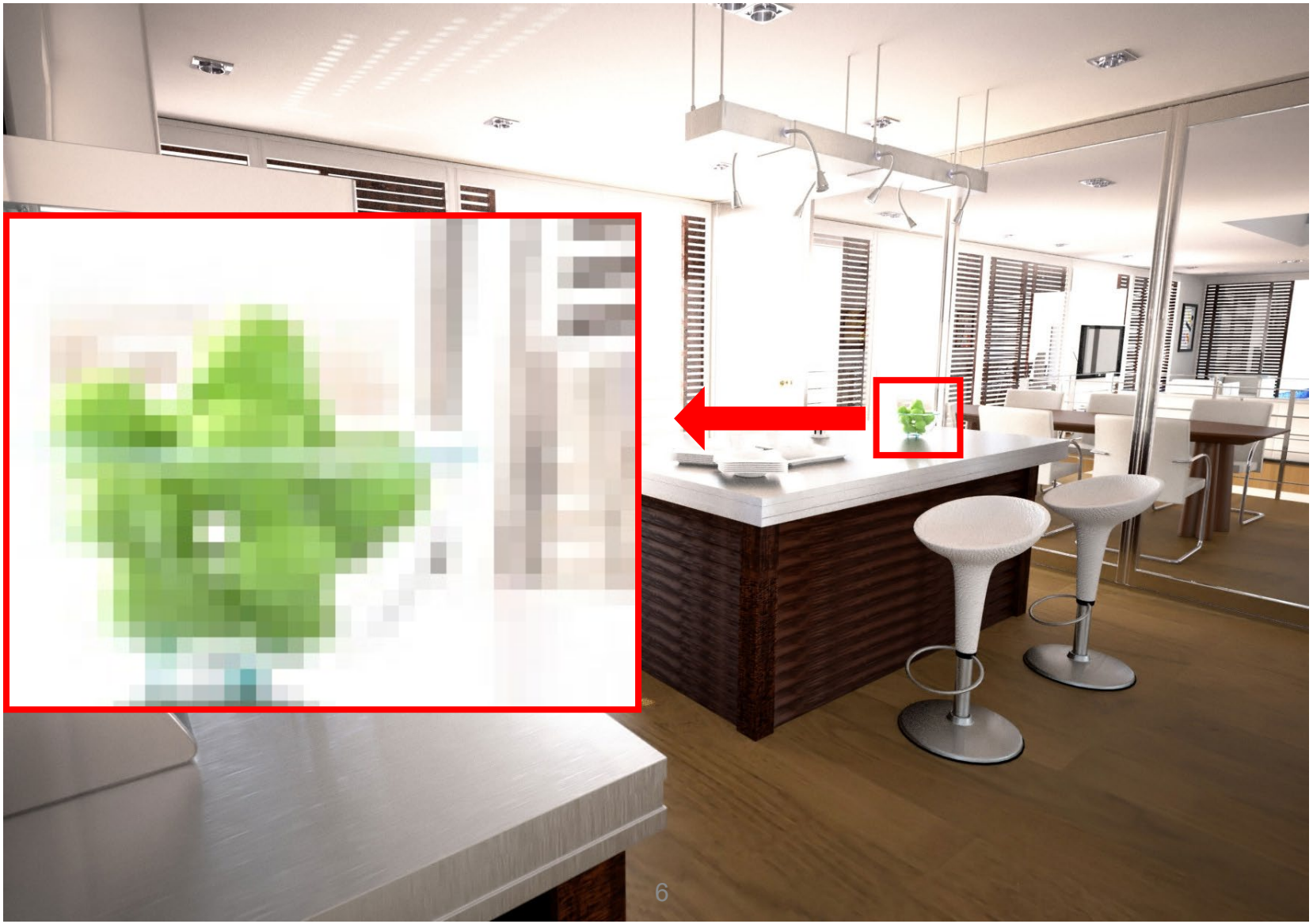
- モデリングによって定義された3次元形状を、画面に描画する

- 画面を構成する要素（画素）1つ1つに対して色（RGB値）を割り当てる操作

- レンダリング結果を定める要素
  - カメラの位置や視野角（透視投影変換）
  - 光源の位置、方向、色
  - 物体表面の材質（光の反射の仕方）

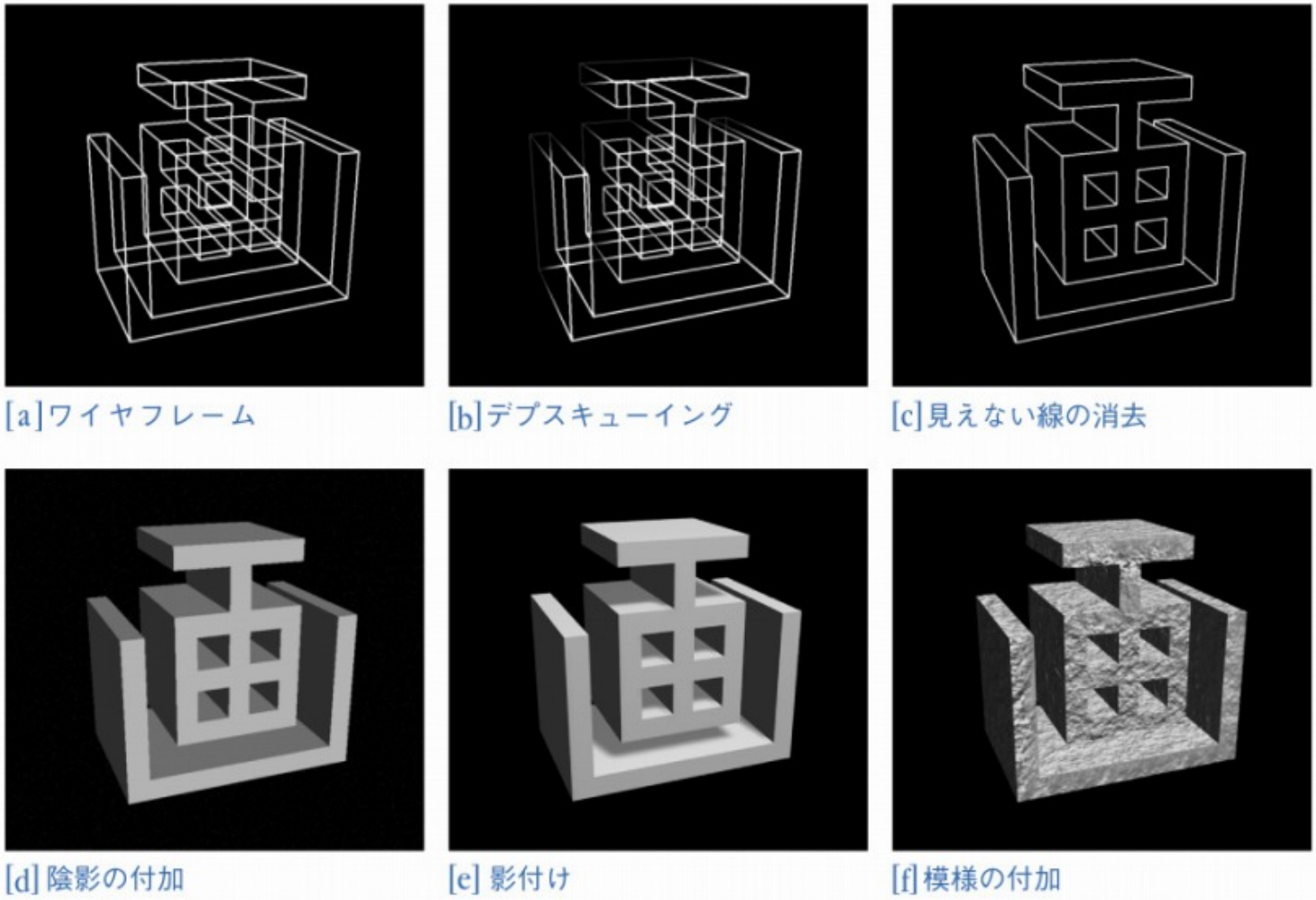
レンダリング（Photo Realistic Rendering）

レンダリング（Photo Realistic Rendering）



3Dモデルの表示

■図4.1——3次元モデルの写実的表現のモデル



[a]ワイヤフレーム　[b]デプスキューイング　[c]見えない線の消去

[d]陰影の付加　[e]影付け　[f]模様の付加

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

リアリティの要素

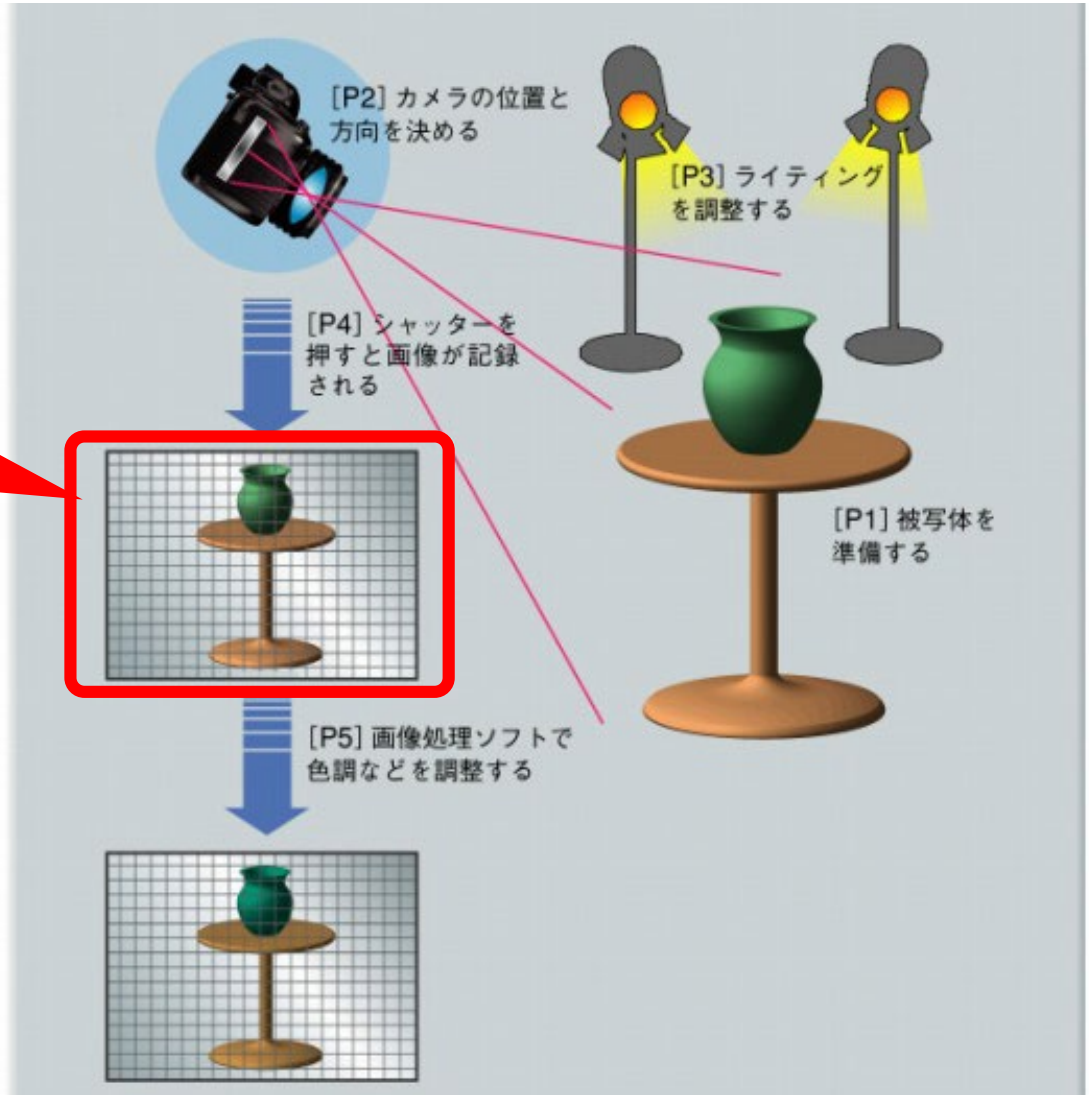
- 遠近感
  - 透視投影変換（遠：小さい　近：大きい）
  - デプスキューイング
    （遠：暗い、ぼやけ　近：明るい、鮮明）

- 可視面表示
  - 隠線消去法
  - 隠面消去法

- 表面の明るさ（陰影付け），影の表示
  - 物体属性：材質，表面の滑らかさ，色（反射特性）
  - 反射光計算（直接光＋間接光）

本物のカメラで撮影するときは？

大別して２通りの方法あり

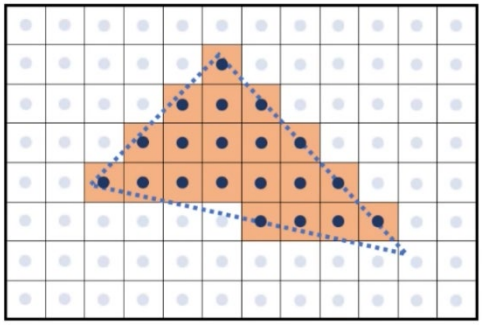


レンダリングの 2 つの枠組み

- ラスタライズベース
  - 三角形などの基本図形を 1 つずつ画面に塗りつぶす (rasterize)

  ○ハードウェア (GPU) で高速処理可能
  → ゲームなどのリアルタイム処理

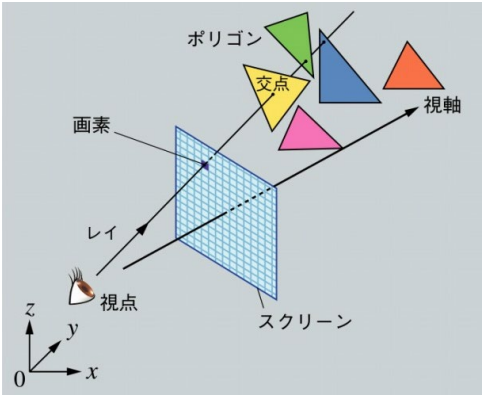
  ×複雑な光学現象を扱うには近似が必要



- レイトレーシング
  - 光の経路を 1 本ずつ追跡

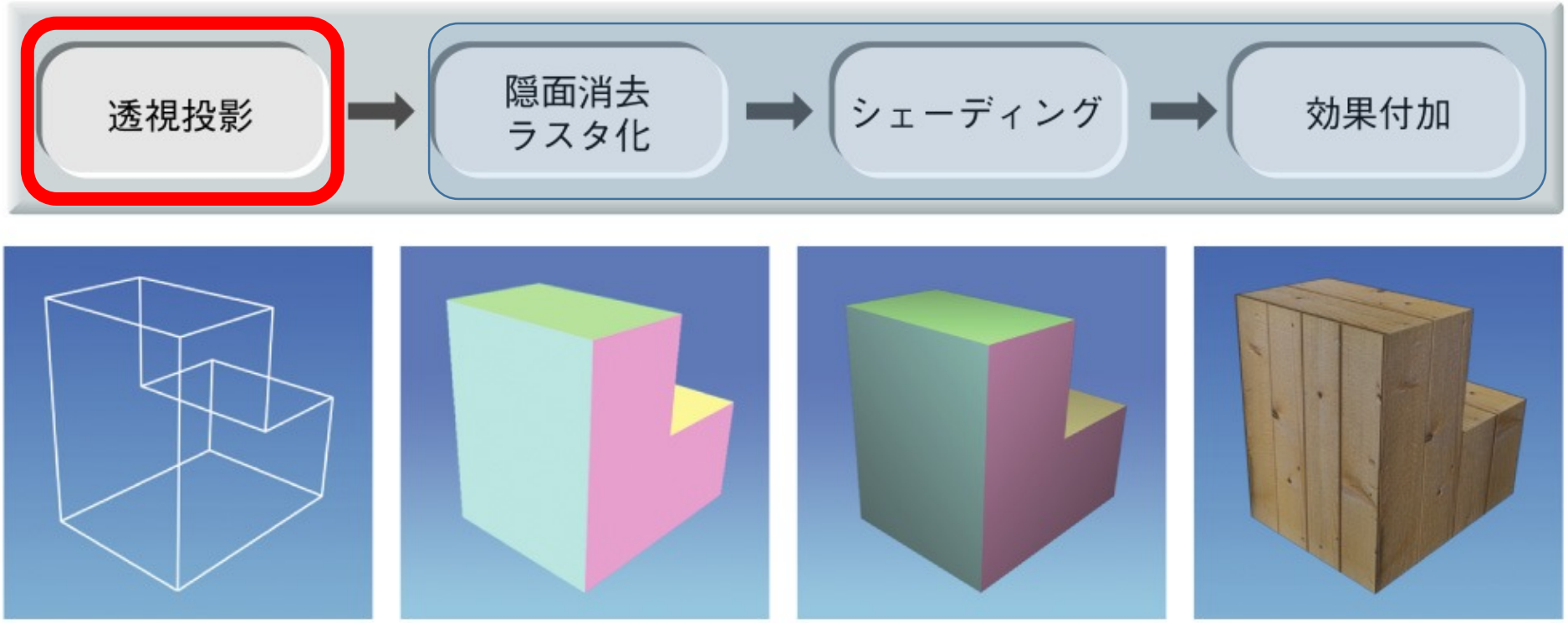
  ○複雑な光学現象を自然に扱え、高品質
  → 映画などのオフライン処理

  ×計算コストが高い
  → GPU サポート (NVIDIA RTX シリーズ)



ラスタライズベースの処理手順

■図4.2——レンダリングを構成する処理過程



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

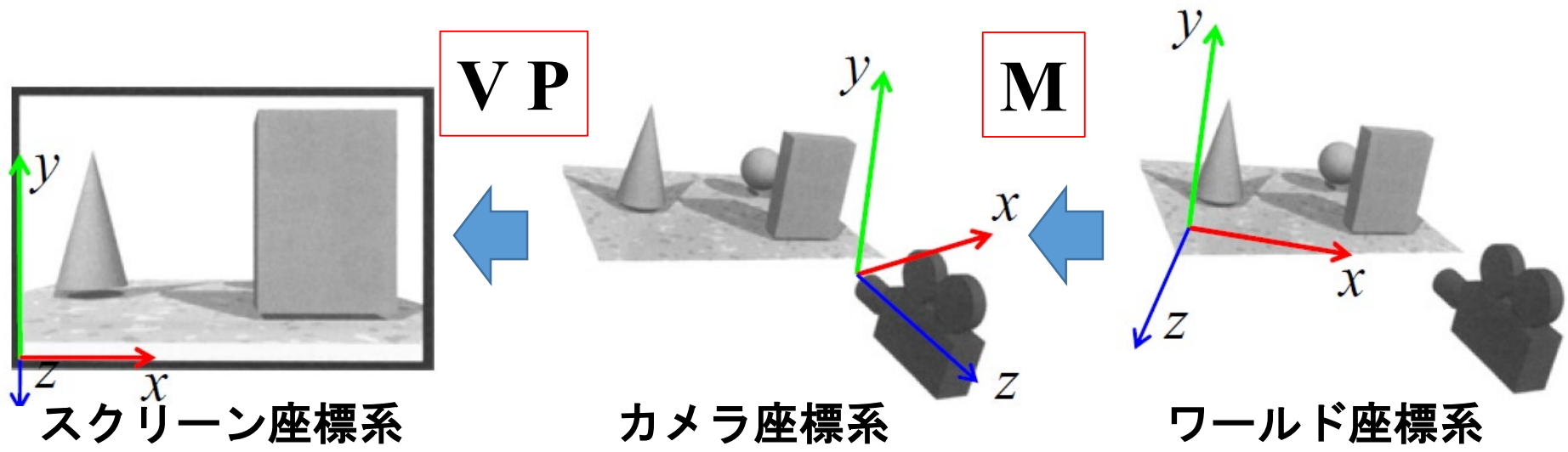
復習: 座標変換

- 座標変換はベクトル・行列で記述できる
  - ただし 4 成分のベクトルと 4×4 行列 (同次座標)

ビューポート変換行列　モデルビュー行列

$$\mathbf{p}_s = \mathbf{V}\,\mathbf{P}\,\mathbf{M}\,\mathbf{p}_w$$

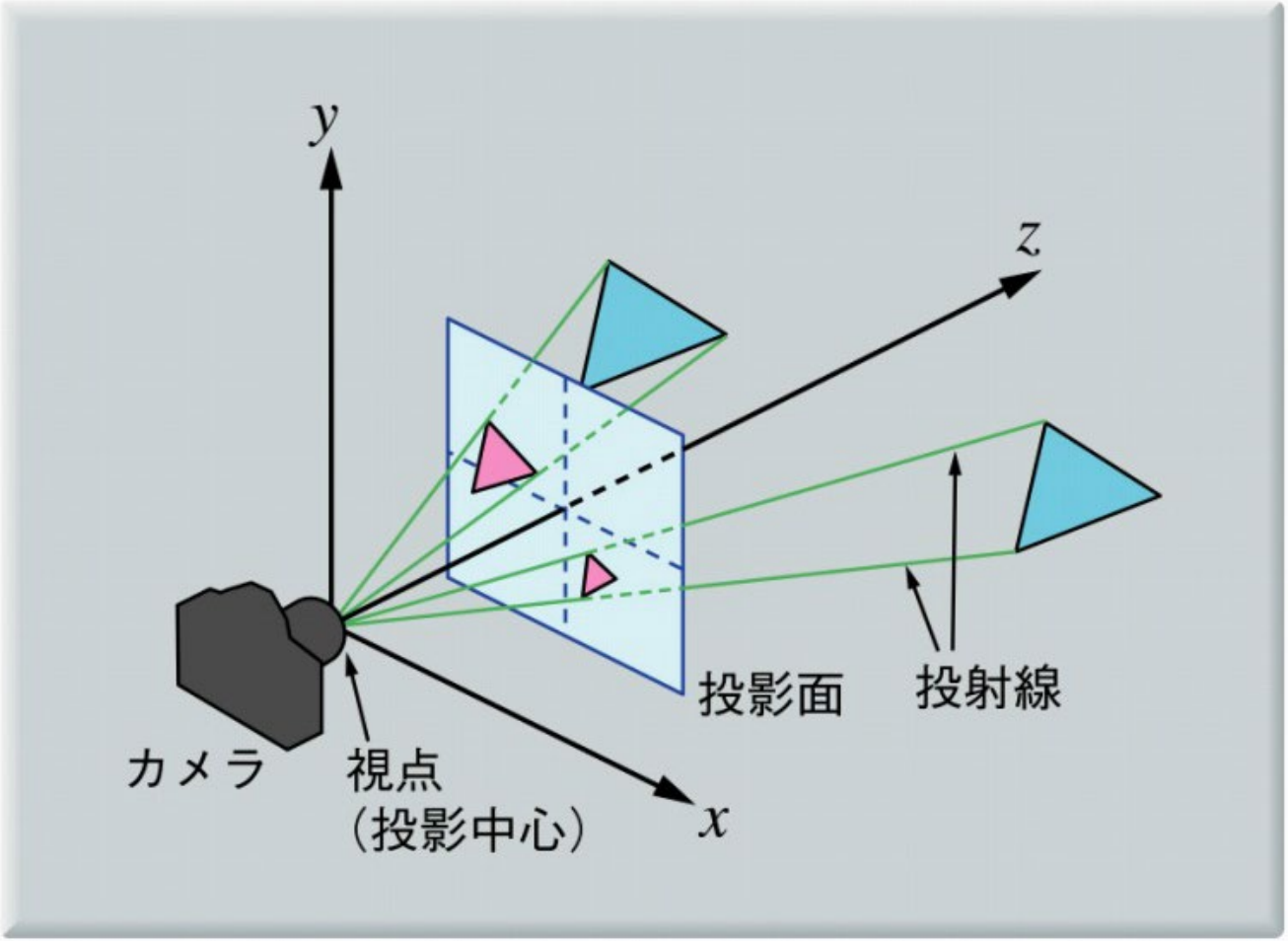
スクリーン座標　透視投影行列　ワールド座標



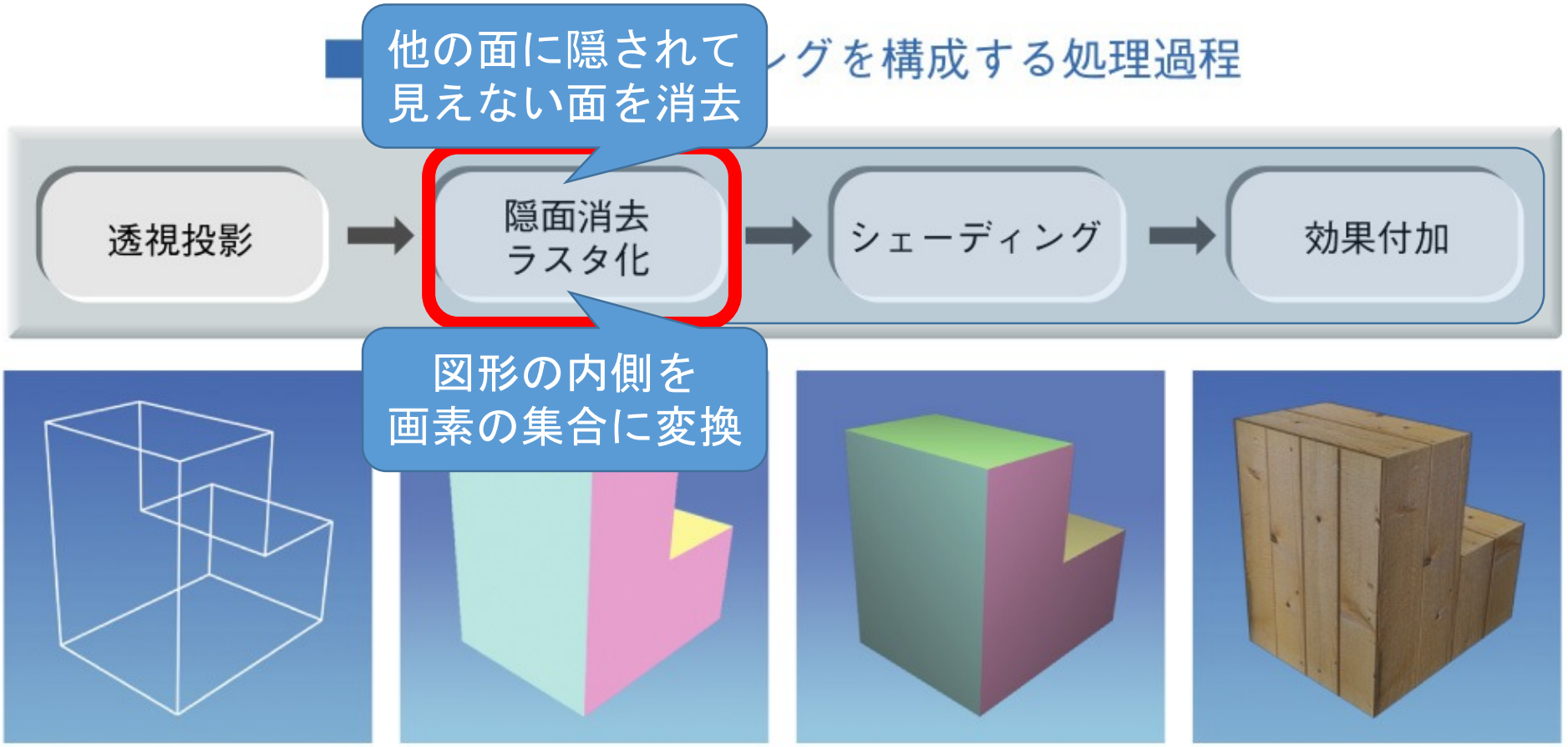
スクリーン座標系　カメラ座標系　ワールド座標系

復習: 透視投影 (遠近法)

- 遠くのものは小さく、近くのものは大きく表示される
  - 直線は直線のまま

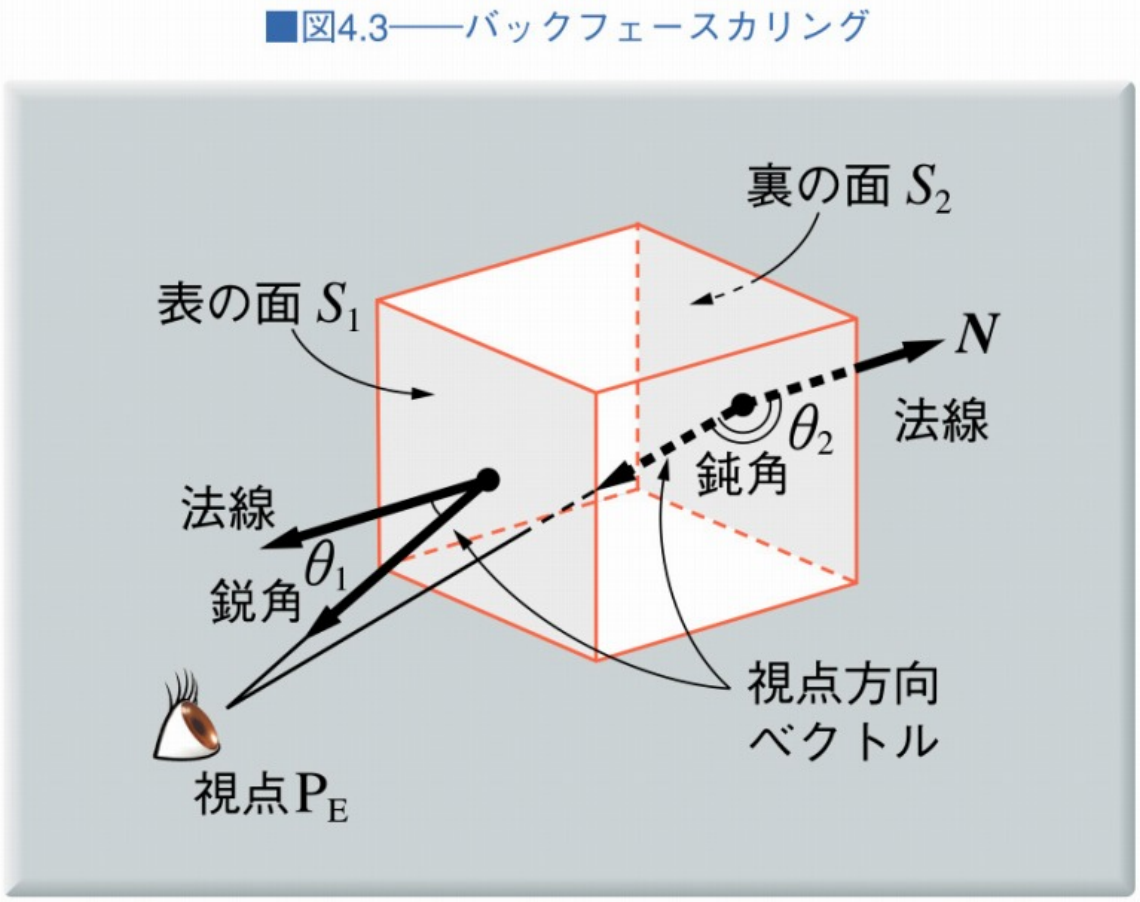


ラスタライズベースの処理手順



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

Back-face culling による隠面消去

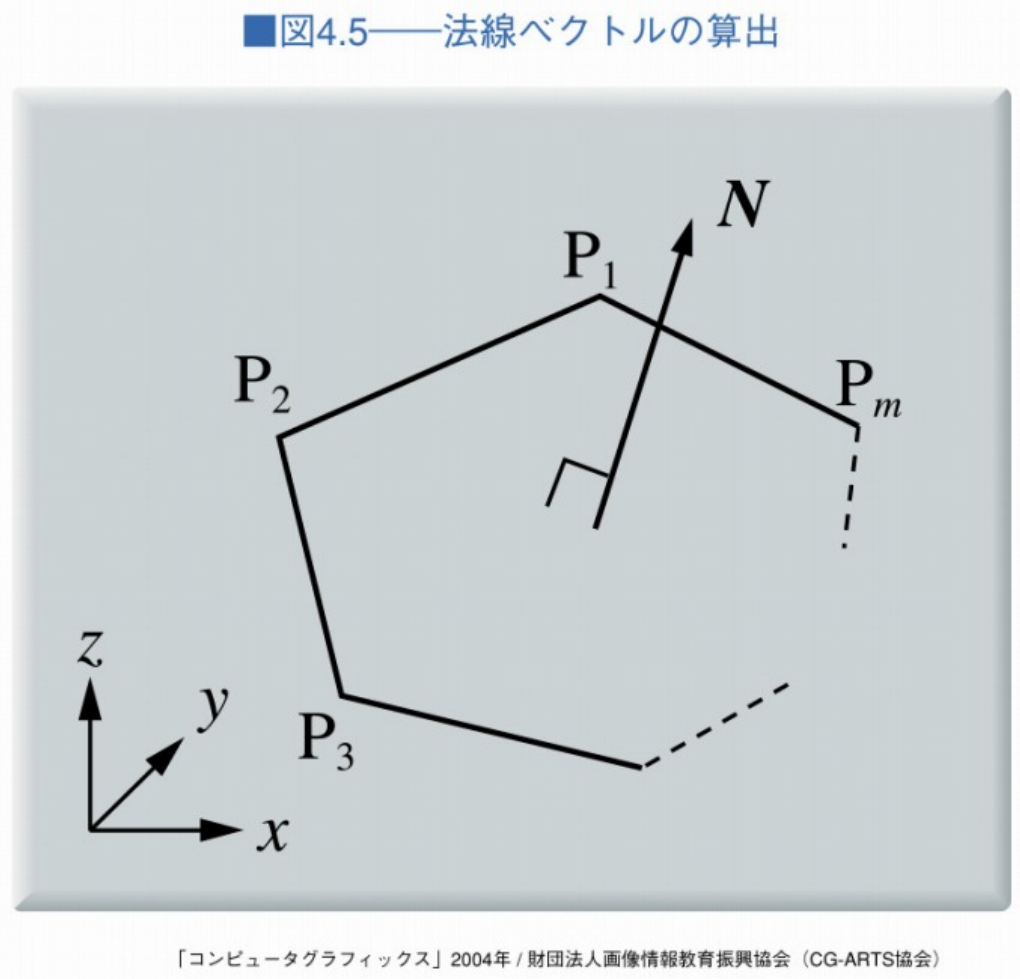


視線ベクトルと法線の内積

D = V • N

D > 0 ：見える
D < 0 ：見えない

法線ベクトル



法線ベクトルは面を
構成する3頂点から
算出される

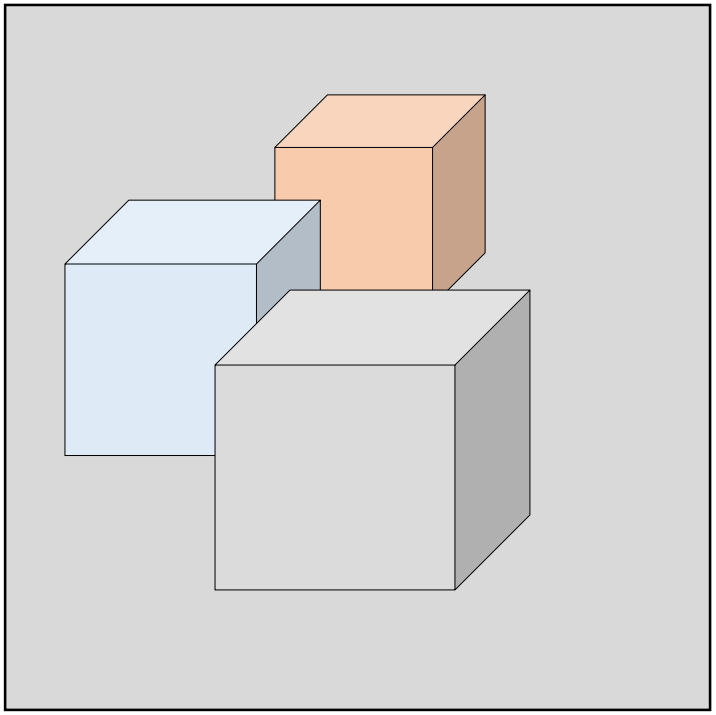
$$N = (P_2 - P_1) \times (P_3 - P_2)$$

（記号「×」は外積）

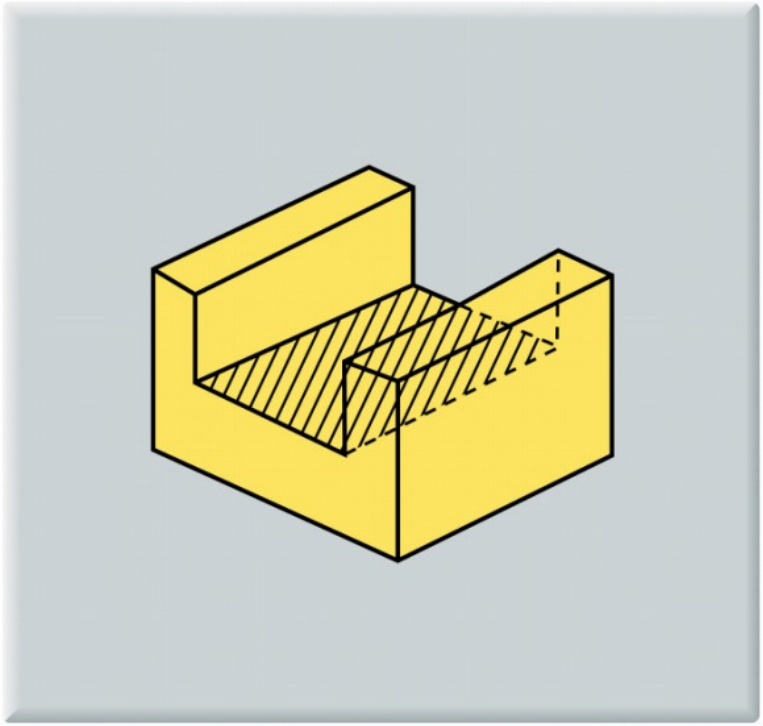
Back-face culling の限界

- 複数物体、凹形状は扱えない
- 追加の処理が必要

複数の物体



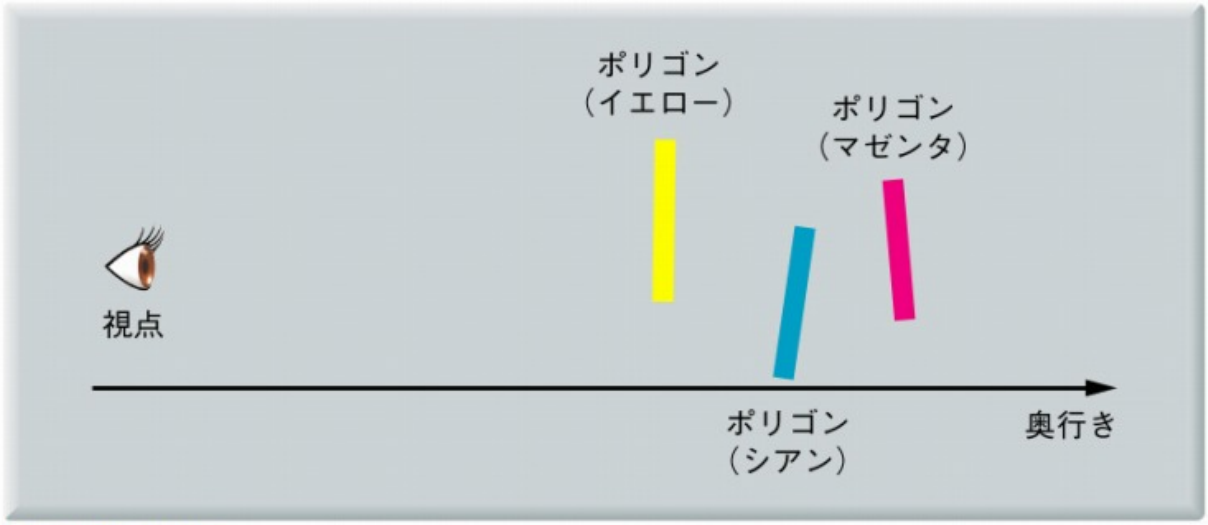
■図4.4——凹多面体を構成する面の可視性



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

優先順位アルゴリズム
奥行きソート法

■図4.6——奥行きソート法の描画過程



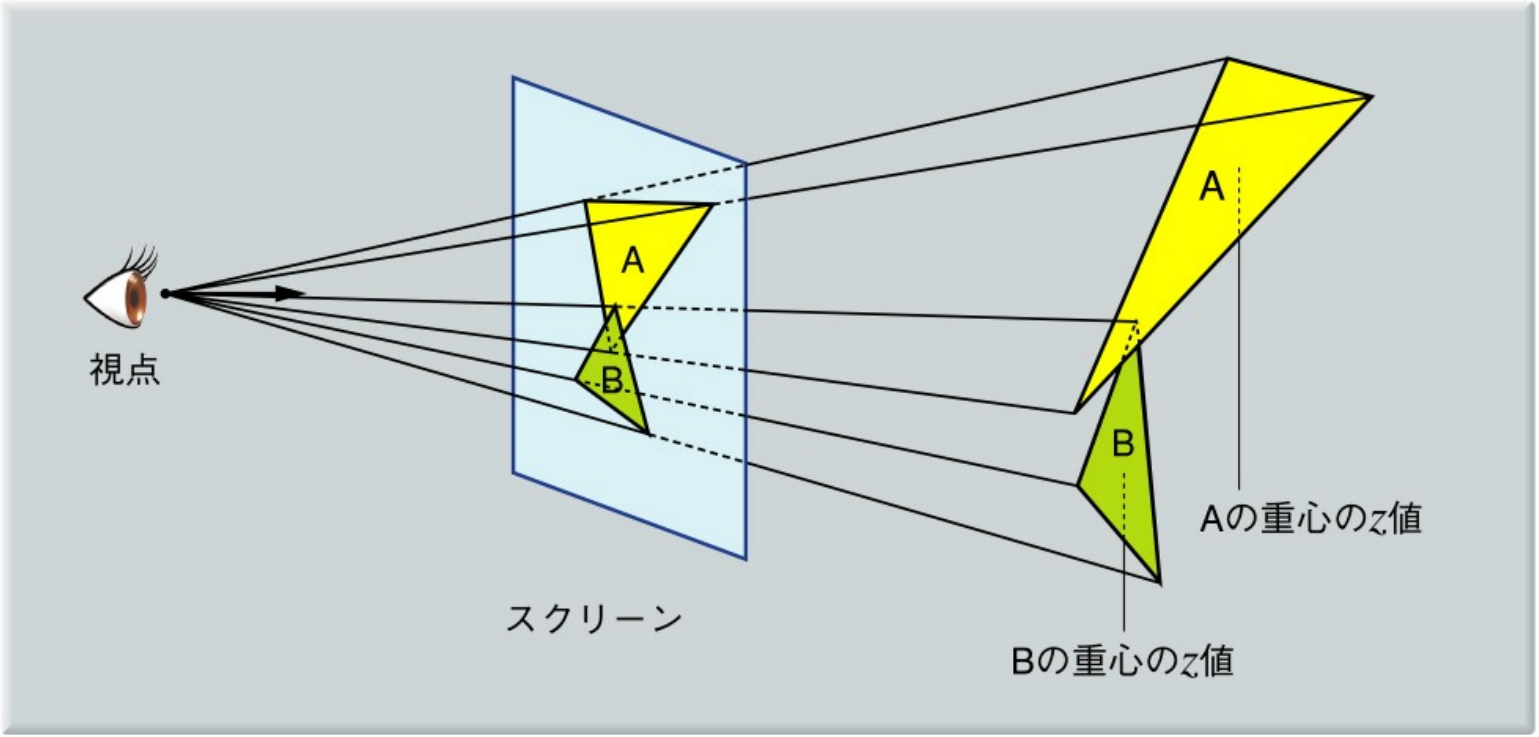
[a] ポリゴンの配置

[b] 描画過程（左から右の図へ描画が進行する）

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

奥行きソート法の限界

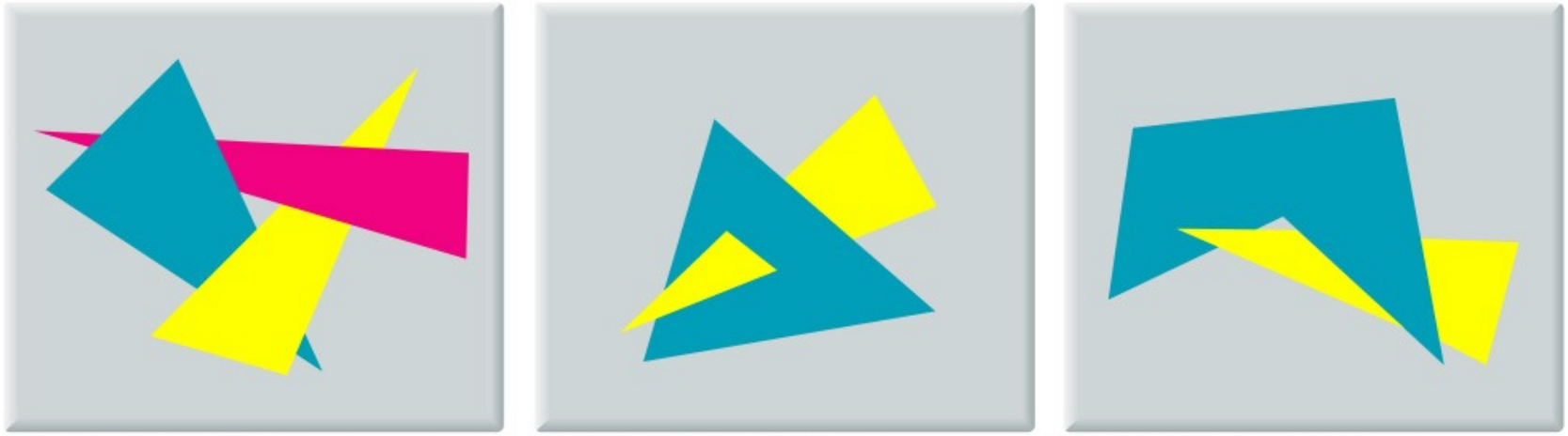
■図4.7——優先順位をポリゴンの重心で定めた場合に隠面消去に失敗する例



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

奥行きソート法の限界

■図4.9——可視性の優先順位が決定できない場合

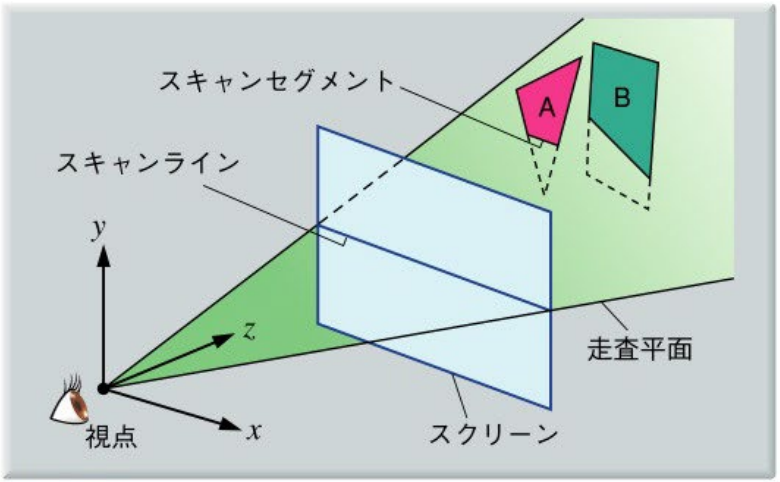


[a] 三すくみの場合　[b] 貫通している場合　[c] 凹ポリゴンの場合

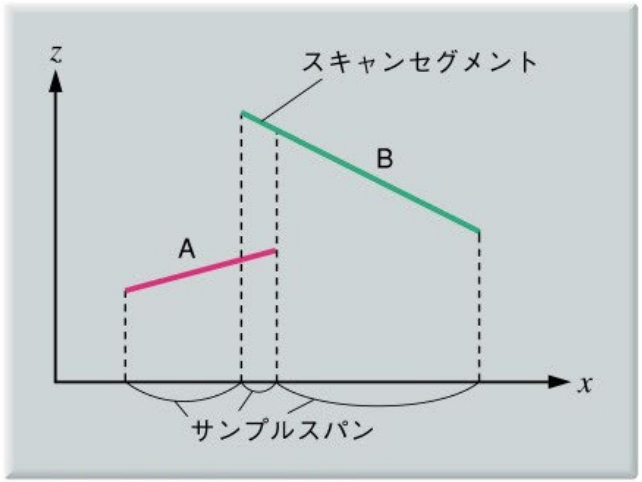
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

スキャンライン法

- スキャンライン（走査）に基づいて隠面消去を行う



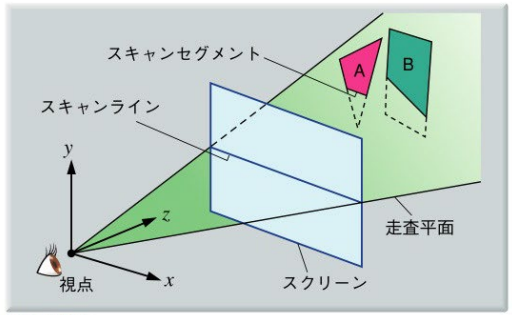
[a] 走査平面



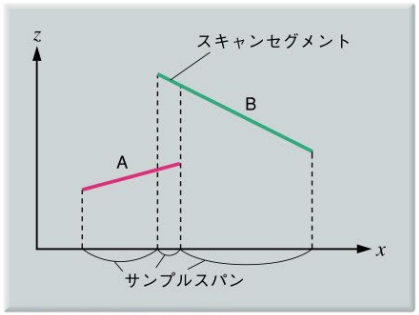
[b] 奥行き判定（走査平面断面図）

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

スキャンライン法のアルゴリズム



[a] 走査平面



[b] 奥行き判定（走査平面断面図）

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

■アルゴリズム4.1——スキャンライン法

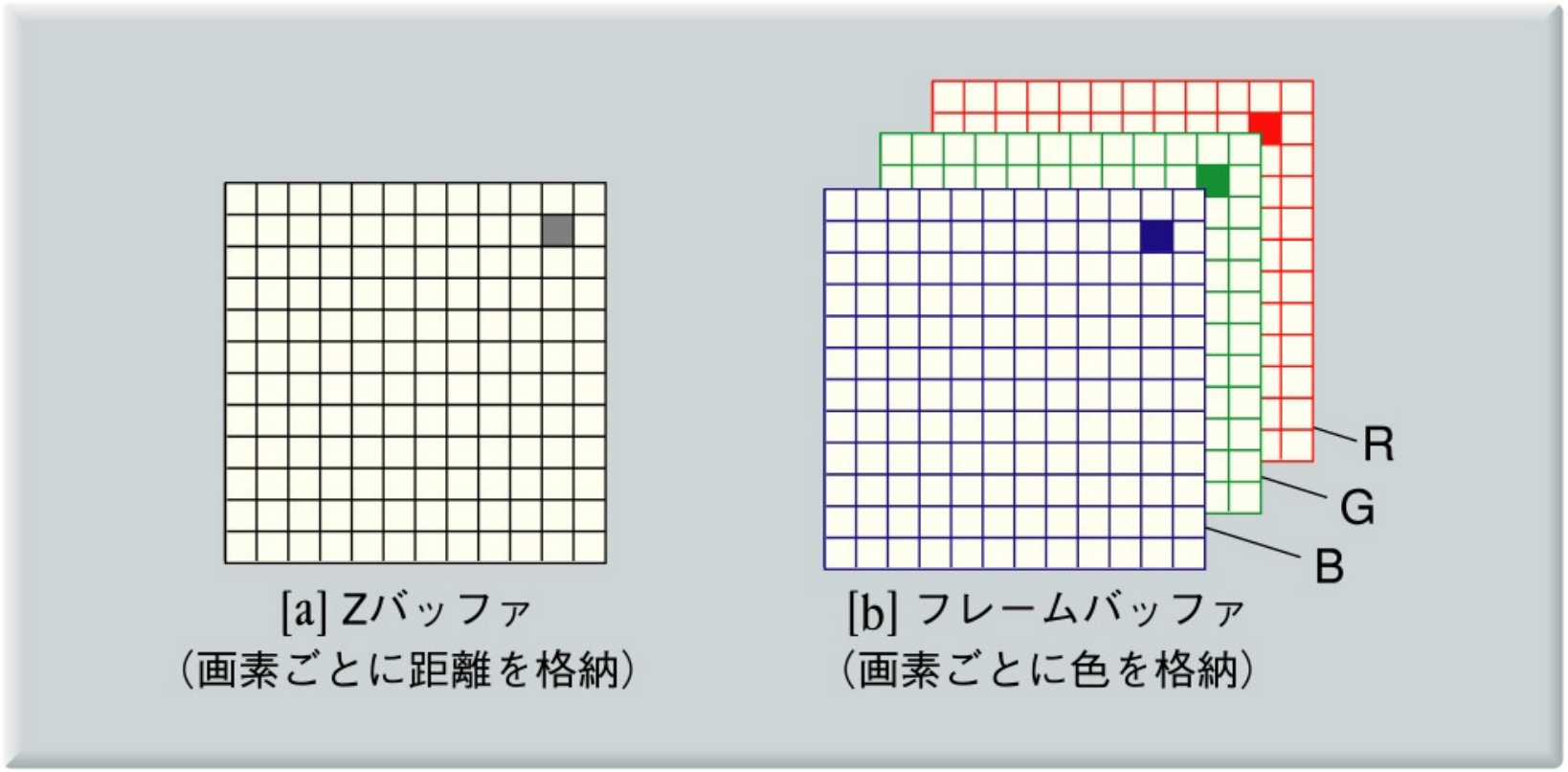
```
すべてのポリゴンを透視投影する;
スキャンラインを上から下へ順番に移動させ,
それぞれのスキャンラインについて{
    走査平面とポリゴンとの交差線分(スキャンセグメント)を求める;
    各スキャンセグメントの端点をx座標の小さい順にソートする(x軸ソート);
    スキャンセグメントの端点によりサンプルスパンを決定する;
    それぞれのサンプルスパンについて{
        奥行きの最も小さいスキャンセグメント(可視セグメント)を求める;
        そのサンプルスパンに可視セグメントの色を塗る;
    }
}
```

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

Z バッファ法

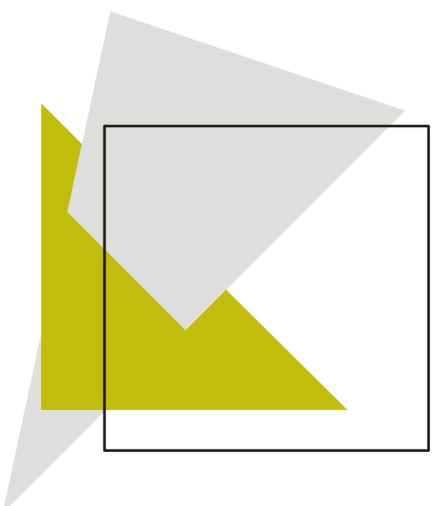
- Z バッファ：画素ごとの奥行き値（デプス値）を格納
- フレームバッファ：描画色を格納

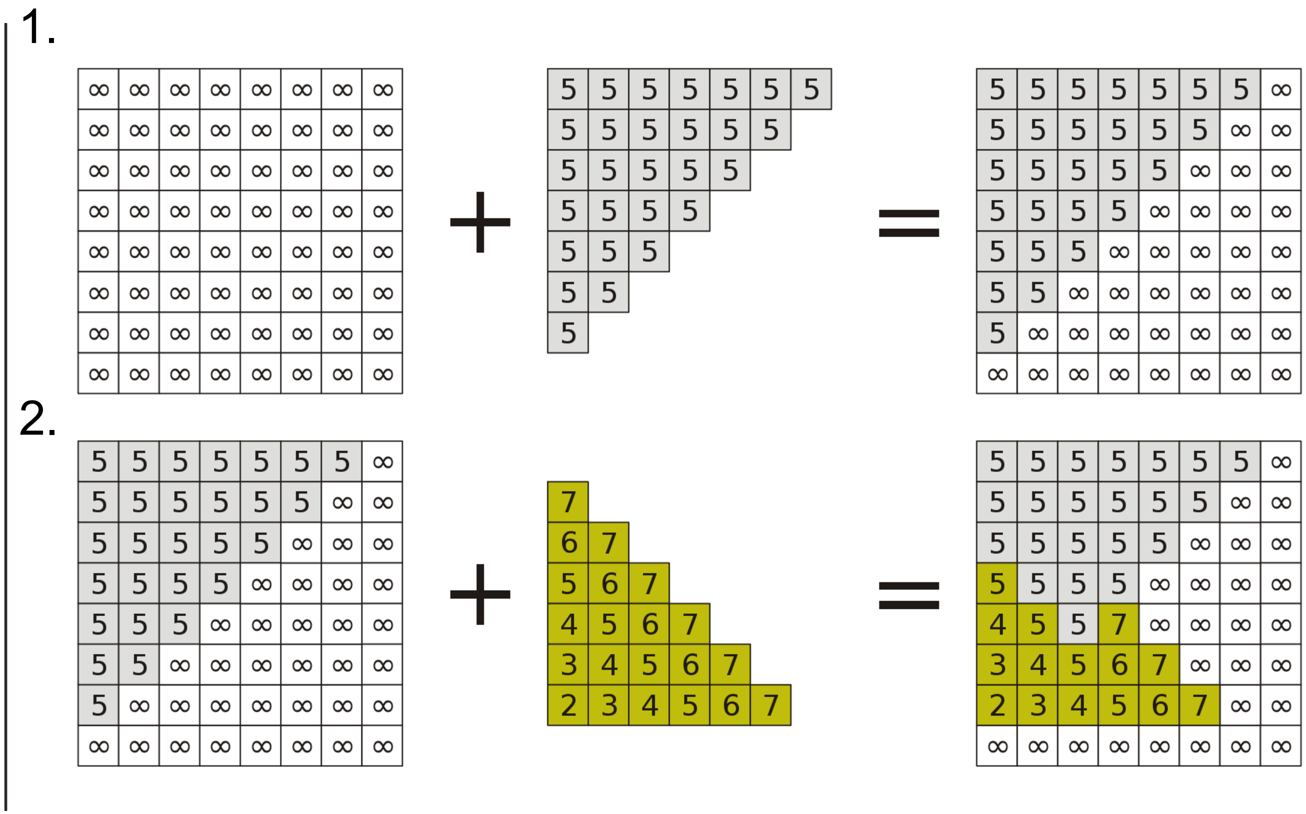
■図4.14——Zバッファとフレームバッファ



Zバッファ法のアルゴリズム

1. Zバッファを大きな奥行き値 (∞) で初期化
2. 各ポリゴンを塗りつぶす際に、Zバッファにすでに記録された値以下のときだけ記録





Z バッファ法のアルゴリズム

■アルゴリズム4.2——Zバッファ法

```
フレームバッファ F(i, j)のすべての画素を背景色で初期化する;
Zバッファ Z(i, j)のすべての画素を最遠点(無限大)で初期化する;
すべてのポリゴンを任意の順番でとり出し,
それぞれのポリゴンについて {
    ポリゴンを透視投影する;
    ポリゴンを走査変換する;
    ポリゴン内部の各画素(i, j)について {
        画素(i, j)に対応する点でのポリゴンの奥行き(z値)を求める;
        ポリゴンのz値とZ(i, j)を比較して,
        z < Z(i, j)を満たすならば {
            画素(i, j)に対応する点でのポリゴンの色をF(i, j)に格納する;
            Z(i, j)をポリゴンのz値で更新する;
        }
    }
}
```

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

Z バッファ法の特徴

- アルゴリズムが簡単でハードウェア化が容易
- OpenGL での描画では Z バッファ法が用いられる

例：第 3 回の OpenGL（GLUT）のソースコードより抜粋

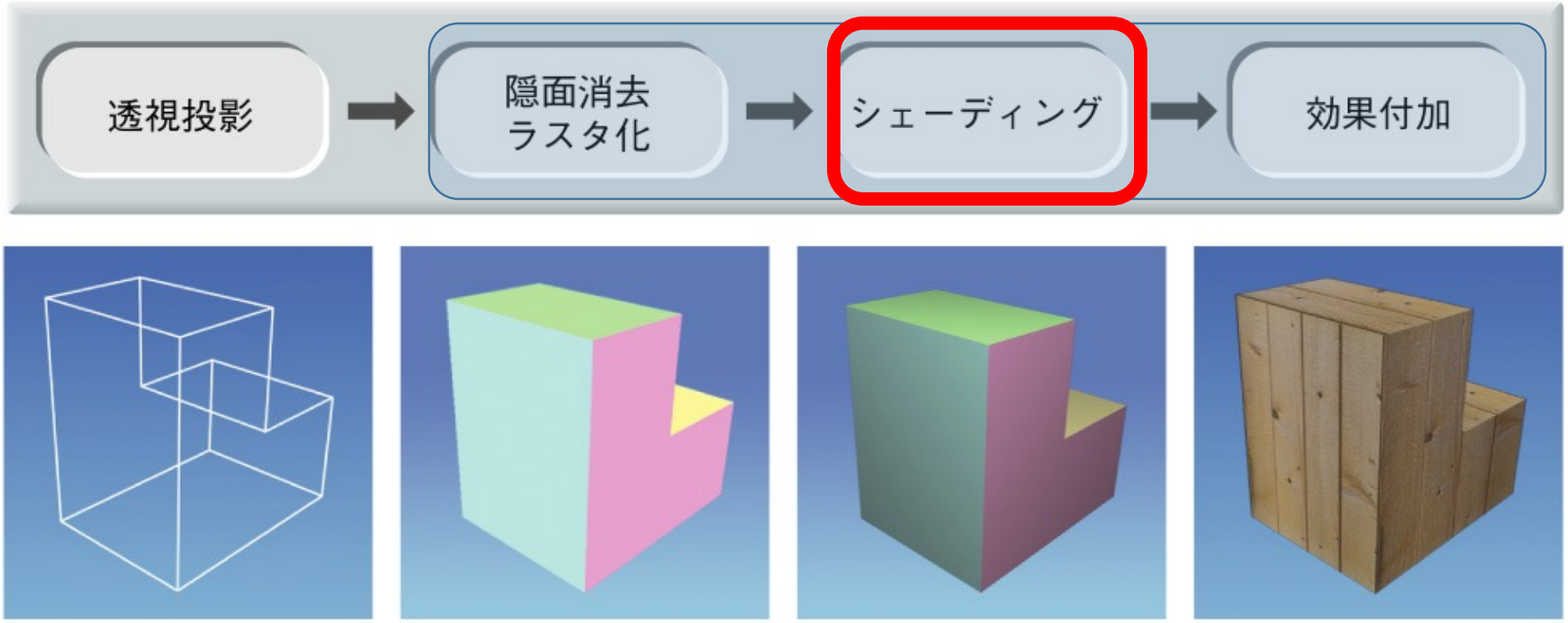
```
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);

glEnable(GL_DEPTH_TEST);
```

- 画素単位の計算 (塗りつぶし)さえできれば
  平面ポリゴンの他、曲面にも適用可能
- 制限：反射や屈折を扱えない
  - 後述のレイトレーシングなら扱える

ラスタライズベースの処理手順

■図4.2——レンダリングを構成する処理過程



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

シェーディングとシャドウイング

- シェーディング（Shading）
  - 表面の濃淡を光の当たり具合から計算
- シャドウイング（Shadowing）
  - 影付け、光が届くかどうかを判定

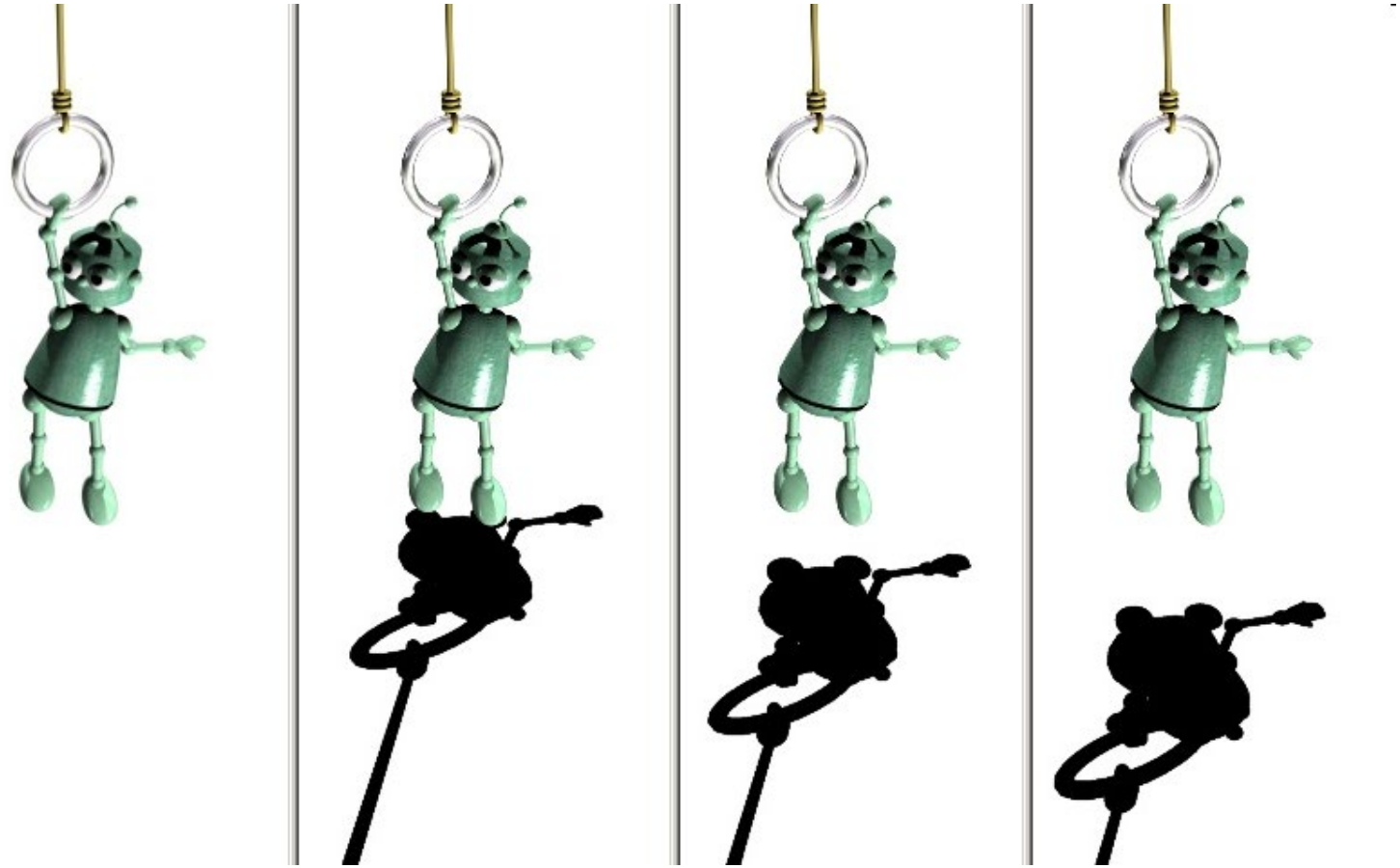
■図4.24——シェーディングと影付け

[a] シェーディングのみ　[b] 影付けのみ　[c] シェーディングと影付けの両方を施した場合

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

シャドウ (影)

- 光が遮られると影 (シャドウ) ができる
- 3 次元空間での位置関係を知るために重要

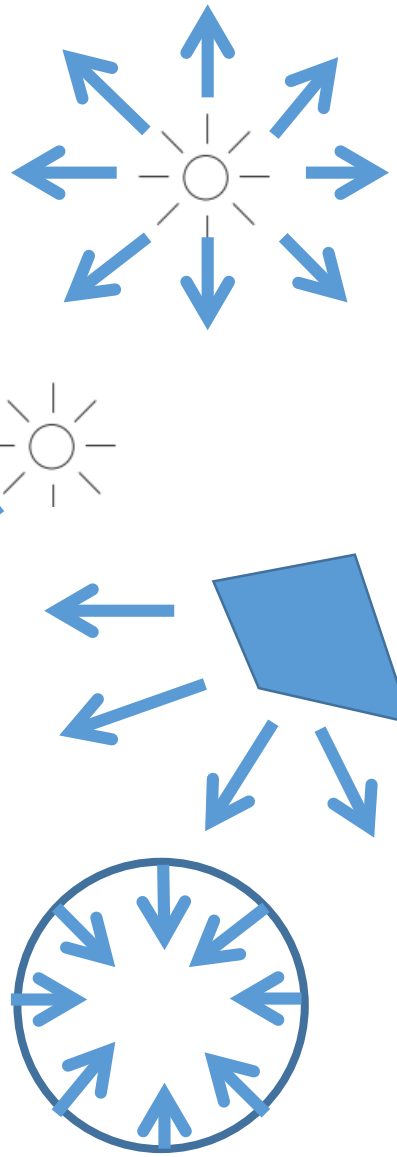


シャドウ (影)

- 光が遮られると影 (シャドウ) ができる
- 3 次元空間での位置関係を知るために重要

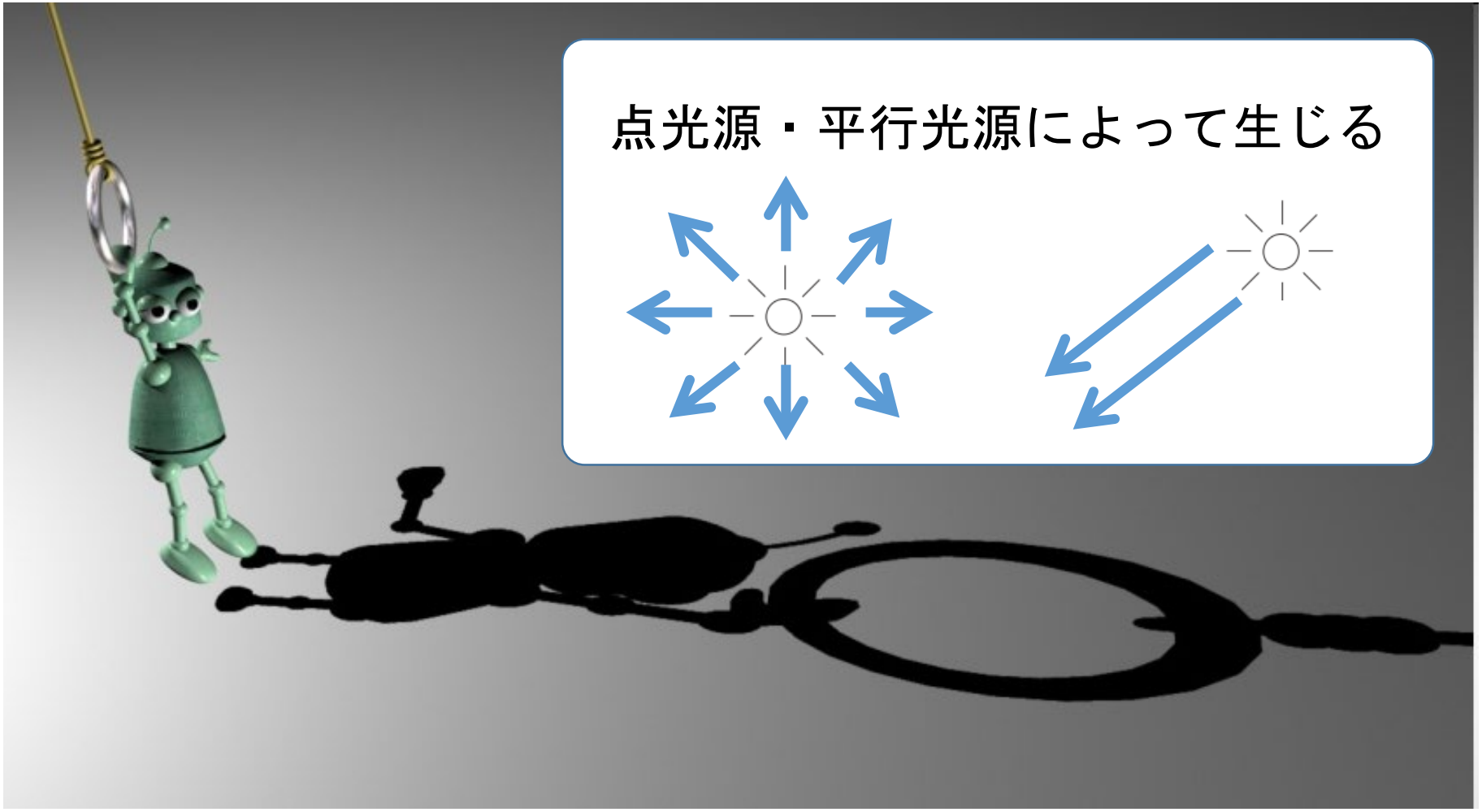
CG でよく使われる光源の例

- 点光源
  - 仮想的な 1 点から光が放射される
  - 距離の 2 乗に反比例して光の強さが減衰
- 平行光源
  - 無限遠から光が届く (例: 太陽光)
  - 光の減衰なし
- 面光源
  - 面積を持った光源 (点光源の集合)
- 環境光源
  - 物体の周囲から届く光 (平行光源の集合)



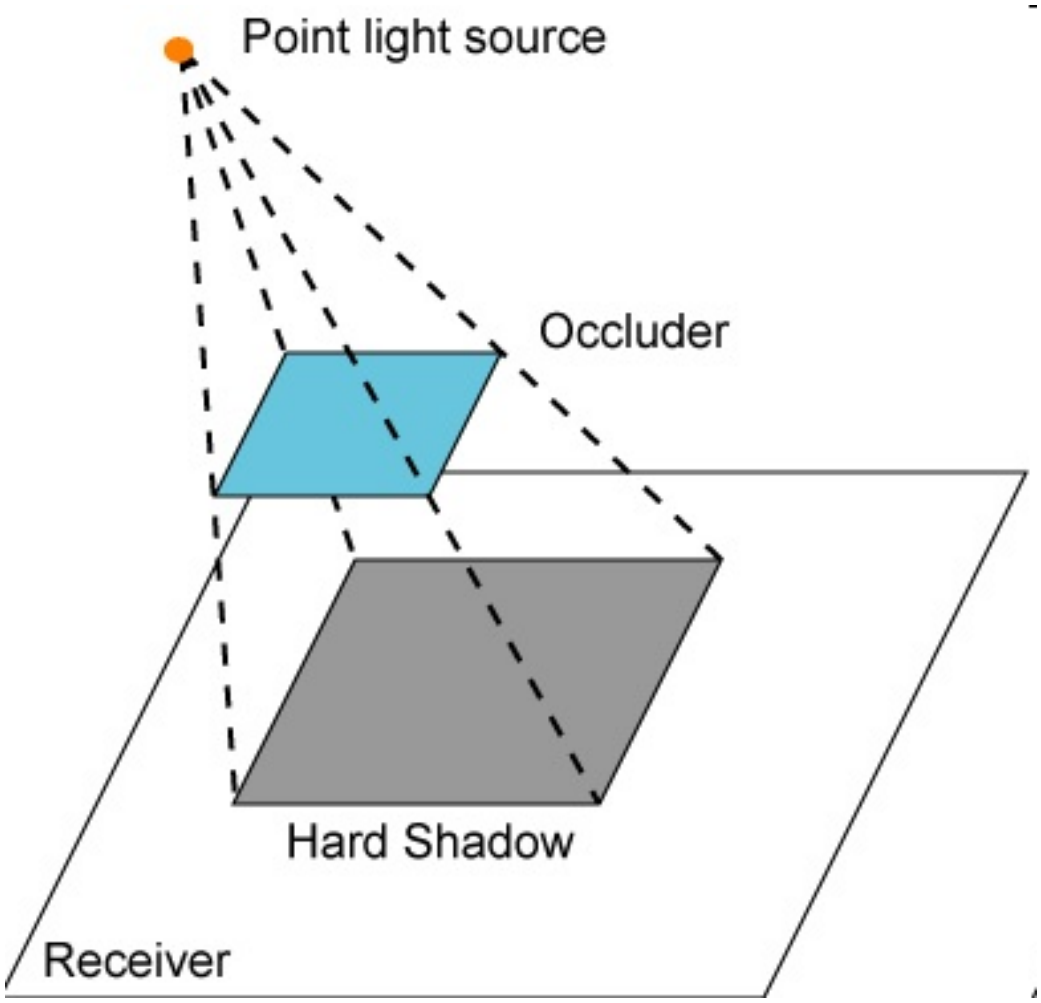
ハードシャドウ

- 輪郭のくっきりした影



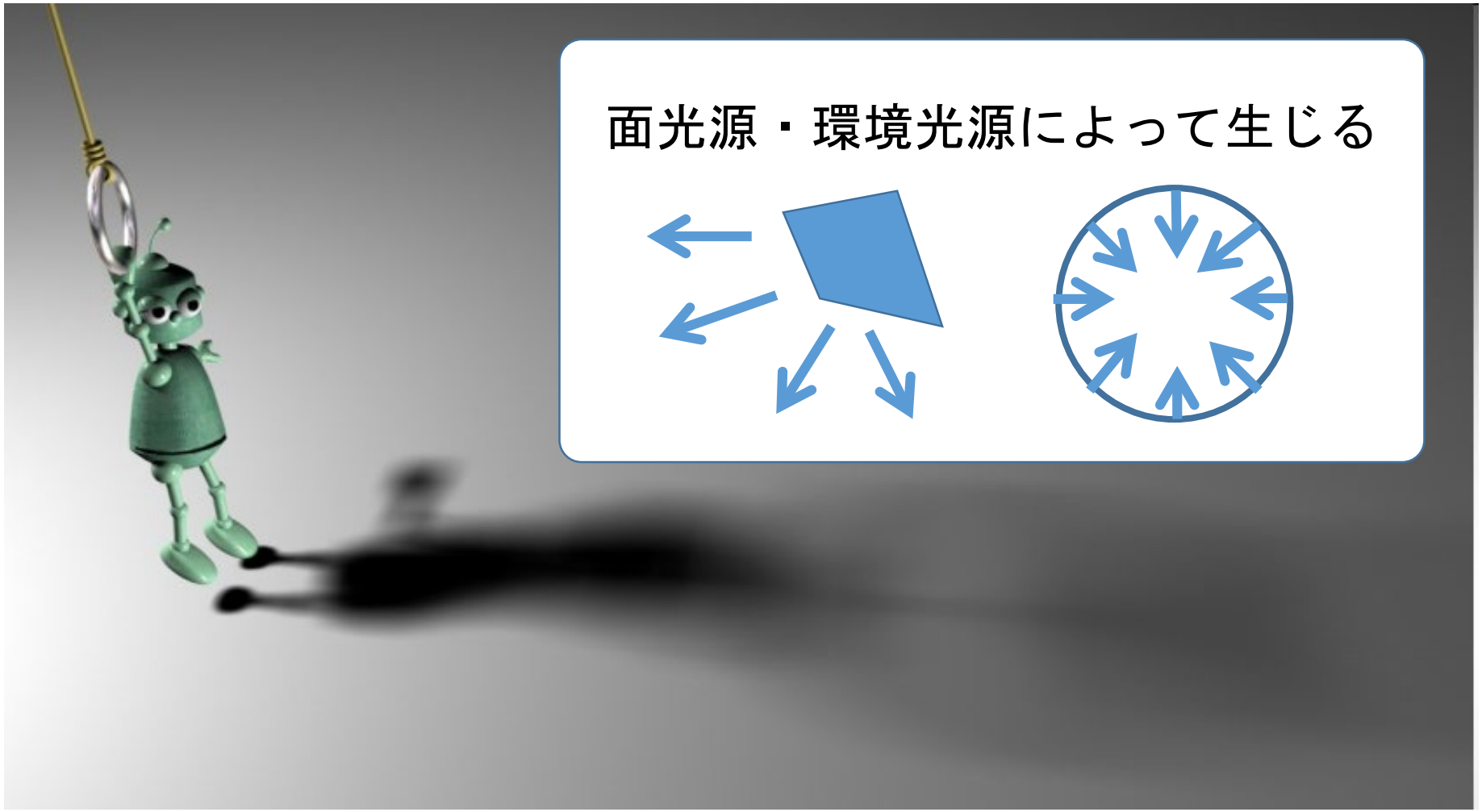
ハードシャドウ

- 輪郭のくっきりした影



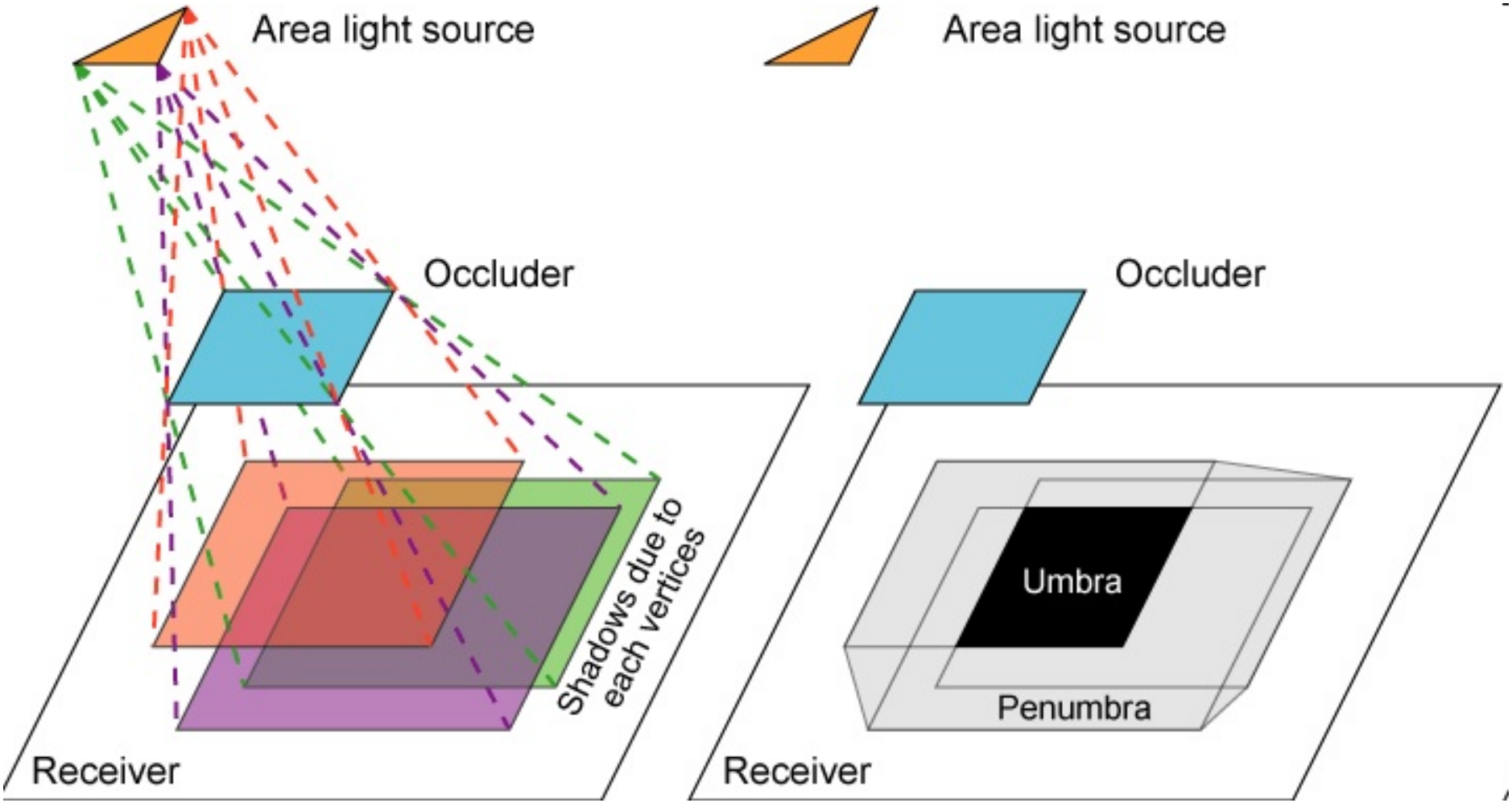
ソフトシャドウ

- 輪郭のぼんやりした影



ソフトシャドウ

- 輪郭のぼんやりした影

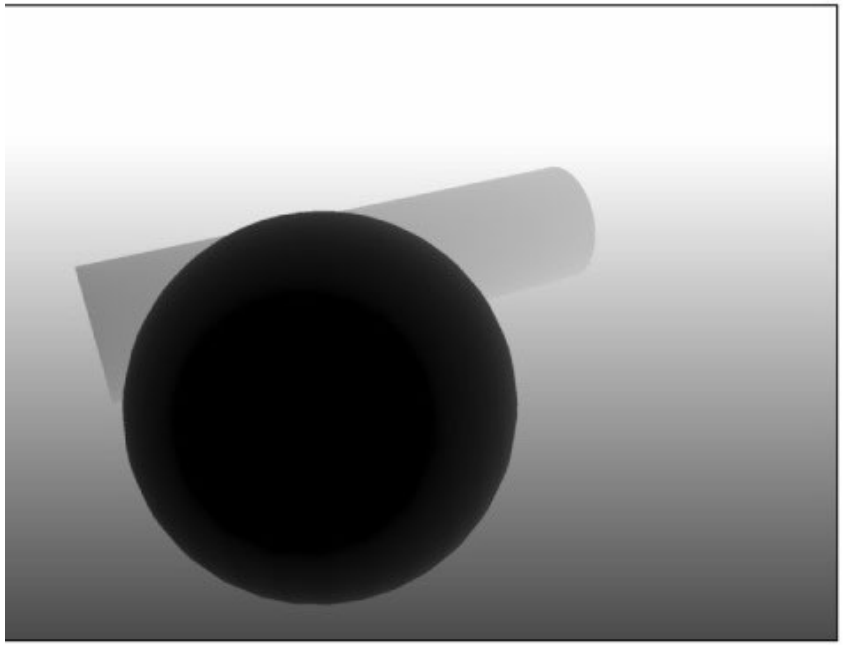


シャドウの計算方法

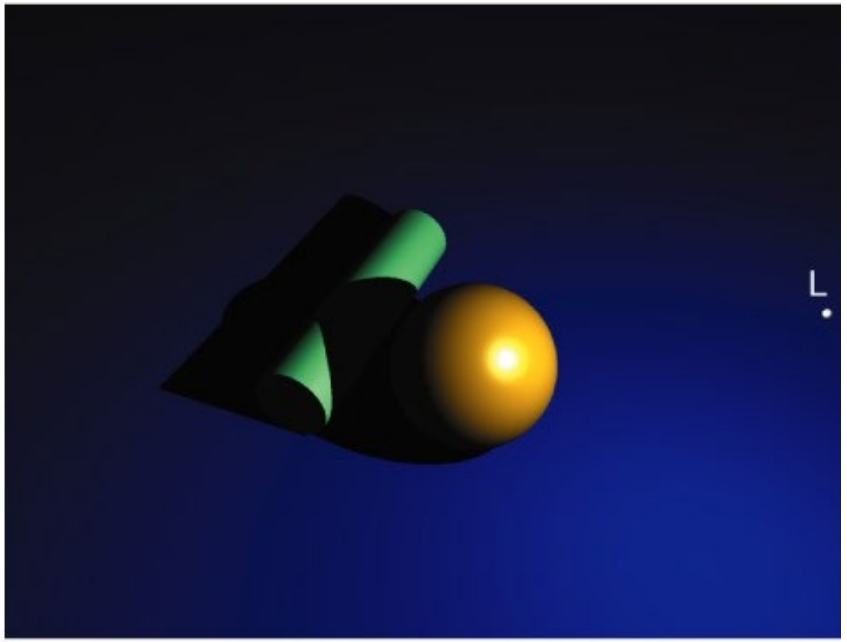
- ある点が影になるかどうか
  ↓
- 光源から見て、その点が「見えるかどうか」
  ↓
- 光源から見た隠面消去
  - ハードシャドウ
    - **シャドウマッピング**
    - シャドウボリューム
  - ソフトシャドウ
    - 単純な多角形の面光源＆ポリゴンメッシュの場合は解析解あり
    - シャドウマッピング＆フィルタリング

シャドウマッピング

1. 光源にカメラを配置してZバッファを計算
2. レンダリングする際に上記のZバッファを参照
   - 光源のZバッファの値の方が小さい (= 光源から遠い) なら光源から見えない → 光が当たっていない → 影



光源で撮影したZバッファ (**シャドウマップ**)

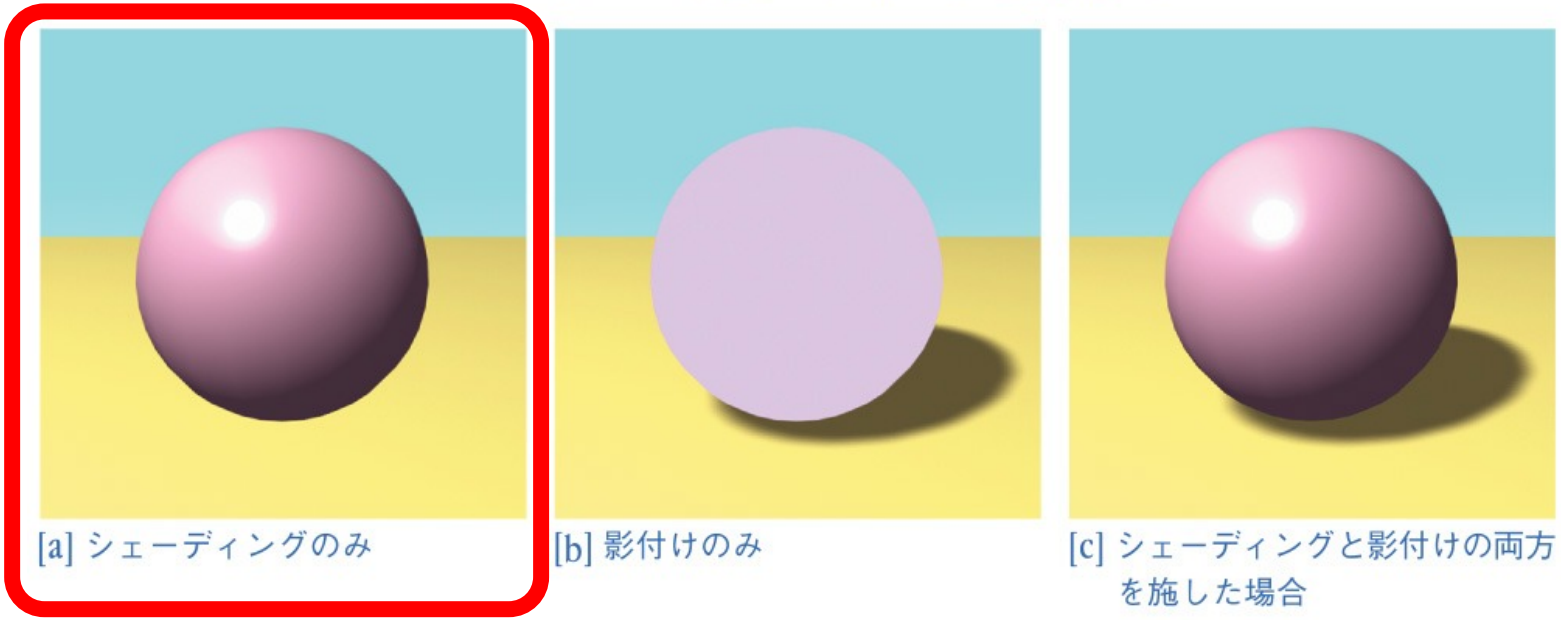


本来のカメラからレンダリングした結果

シェーディングとシャドウイング

- シェーディング（Shading）
  - 表面の濃淡を光の当たり具合から計算
- シャドウイング（Shadowing）
  - 影付け、光が届くかどうかを判定

■図4.24——シェーディングと影付け



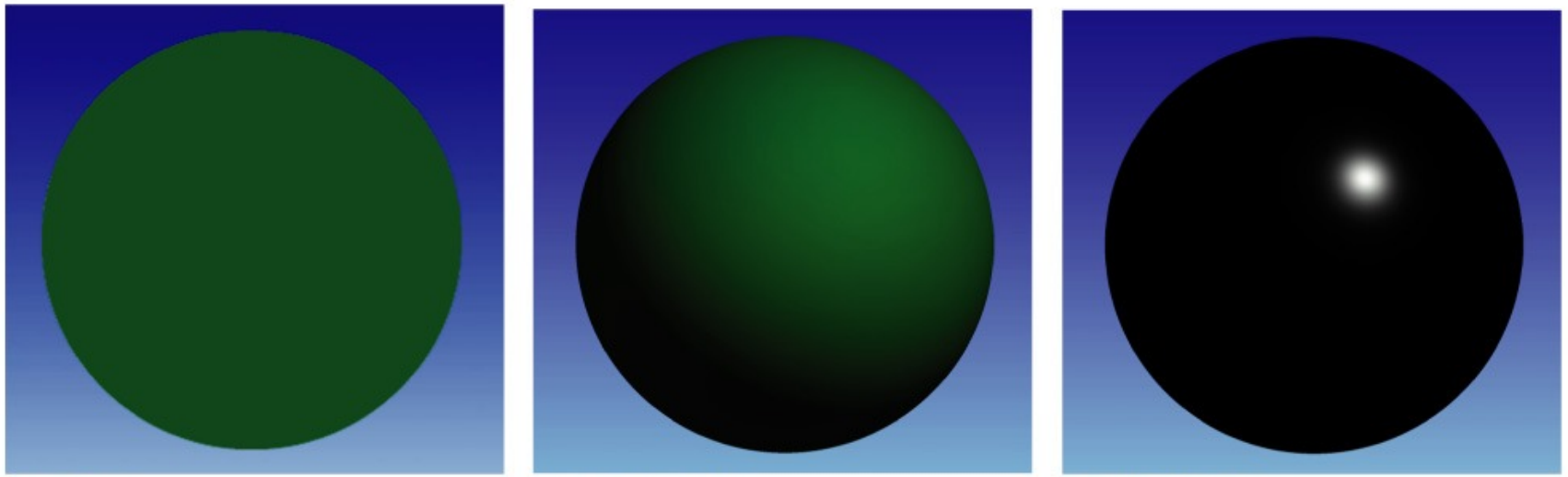
[a] シェーディングのみ　[b] 影付けのみ　[c] シェーディングと影付けの両方を施した場合

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

シェーディングモデル

- 環境光反射：一様に照らす
- 拡散反射：法線方向と光源方向の内積
- 鏡面反射：正反射方向と視線方向

■図4.29——環境光による反射，拡散反射，鏡面反射の各成分



[a] 環境光　[b] 拡散反射光　[c] 鏡面反射光

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

環境光

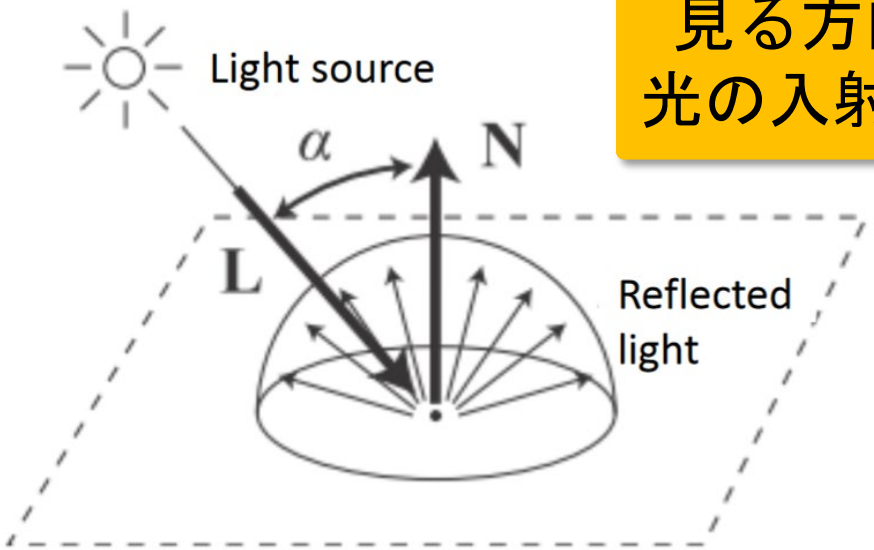
- 物体表面に一様に照らす光
  - 壁などで反射を繰り返して、空間を一様に照らす光を近似したもの
  - 直接は光の届かない陰や影の部分にも明るさを与える
    - （大域照明：間接光の相互反射を計算）

拡散 (diffuse) 反射モデル

（Lambert反射モデル）

- 面上の明るさI_dは、その位置での法線方向と光源方向との内積（cos α）に比例

$$I_d = I_0 k_d \cos\alpha$$



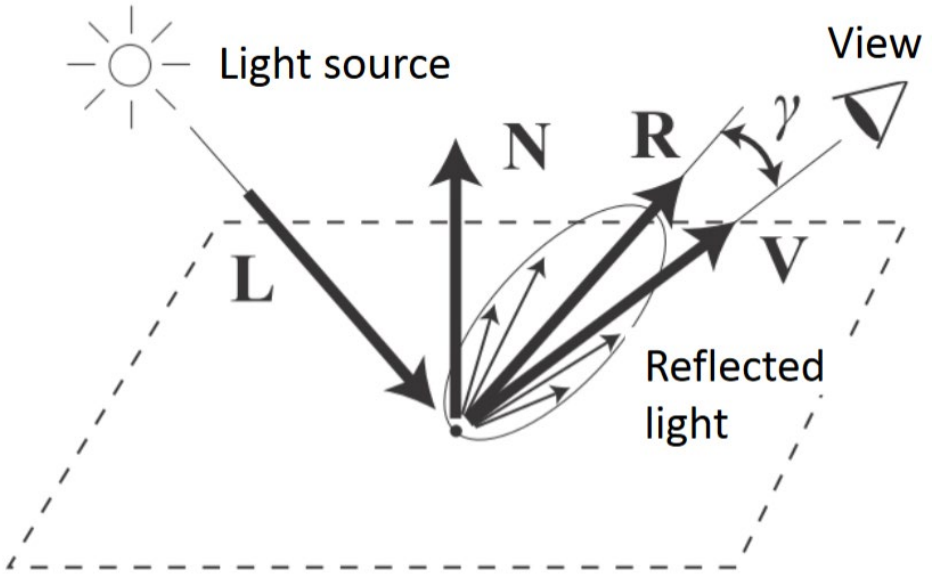
見る方向に依存しない
光の入射方向のみに依存

鏡面 (specular) 反射

（Phong反射モデル）

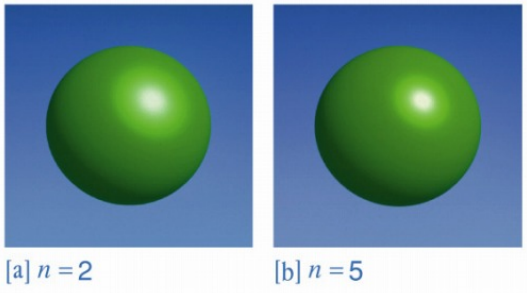
- 光が正反射する方向と視線方向との内積（$\cos\gamma$）のn乗に比例

$$I_s = I_0 k_s \cos^n \gamma$$



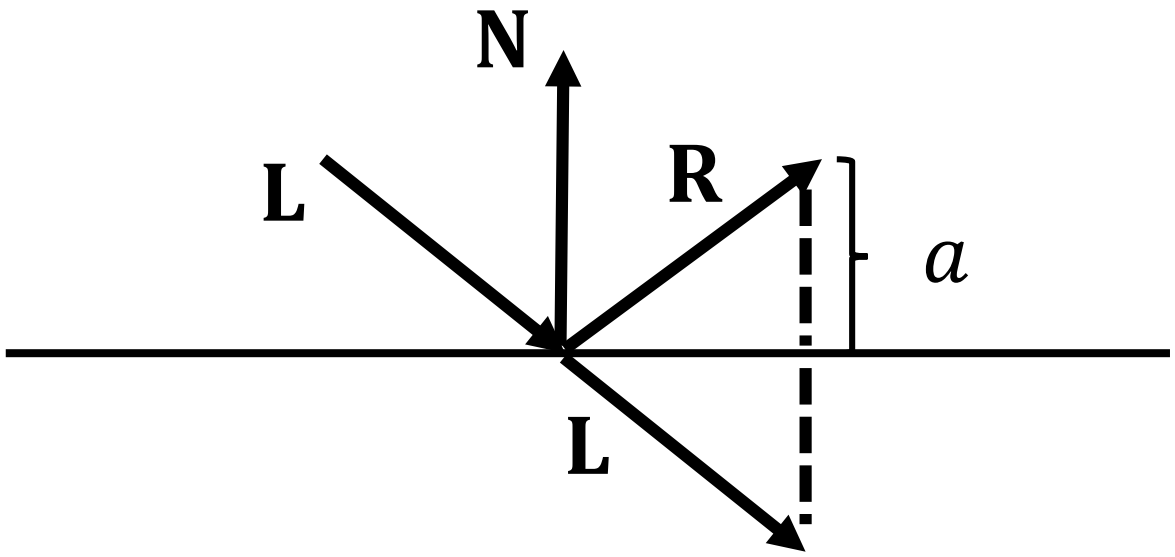
見る方向と光の入射方向の両方に依存

■図4.39——フォンの反射モデルによるハイライトの違い



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

正反射方向 R の計算



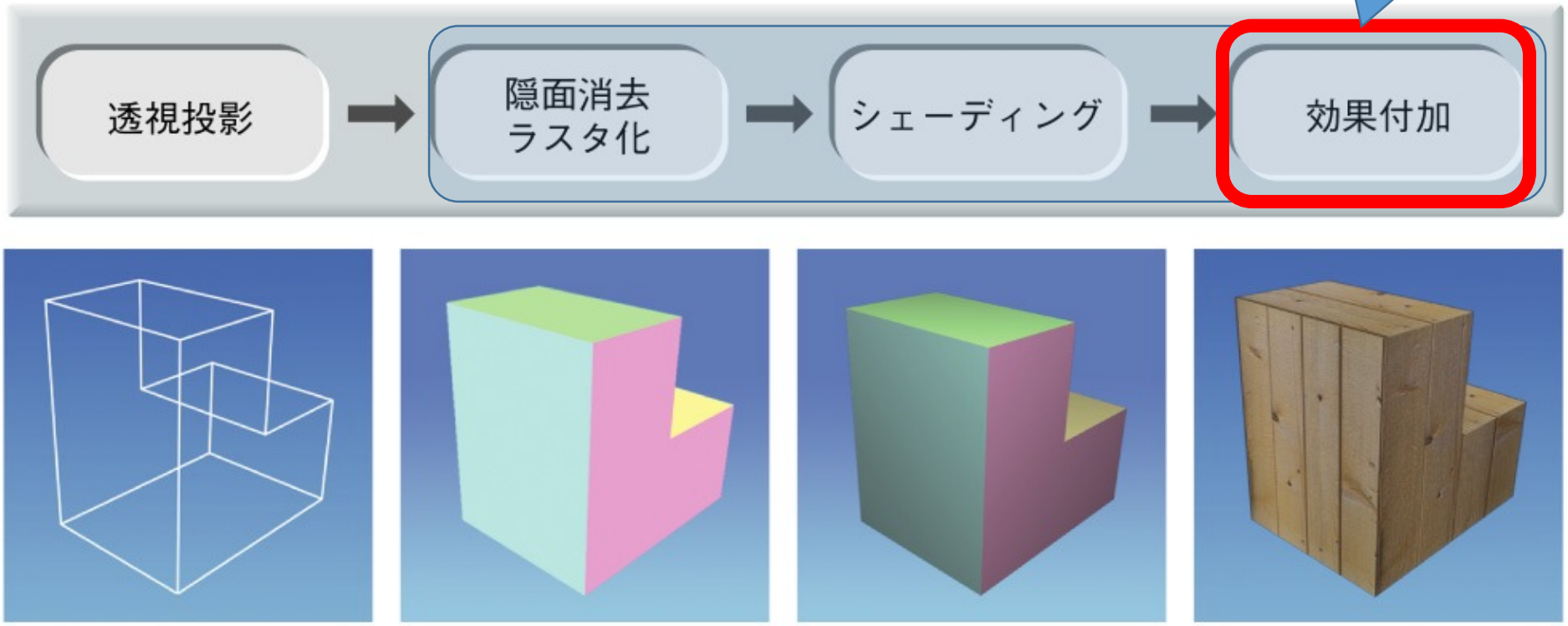
$$\mathbf{R} = \mathbf{L} + 2a\mathbf{N}$$

$$a = -\mathbf{L} \cdot \mathbf{N}$$

ラスタライズベースの処理手順

■図4.2——レンダリングを構成する処理過程

次回！



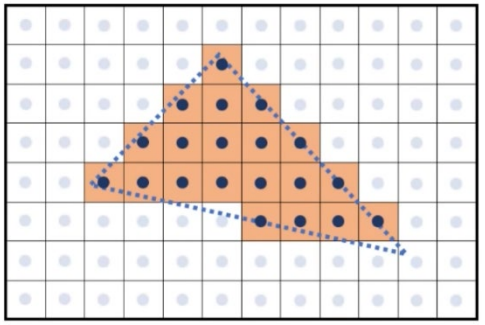
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

レンダリングの 2 つの枠組み

- ラスタライズベース
  - 三角形などの基本図形を 1 つずつ画面に塗りつぶす (rasterize)

  ○ハードウェア (GPU) で高速処理可能
  → ゲームなどのリアルタイム処理

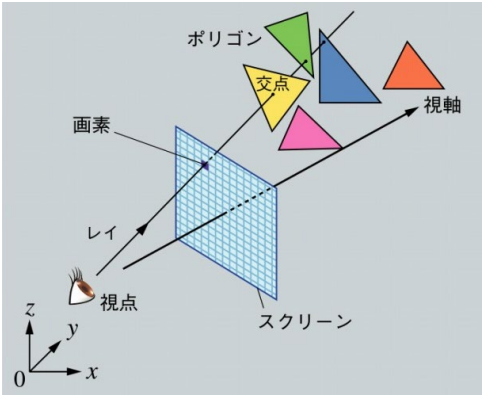
  ×複雑な光学現象を扱うには近似が必要



- レイトレーシング
  - 光の経路を 1 本ずつ追跡

  ○複雑な光学現象を自然に扱え、高品質
  → 映画などのオフライン処理

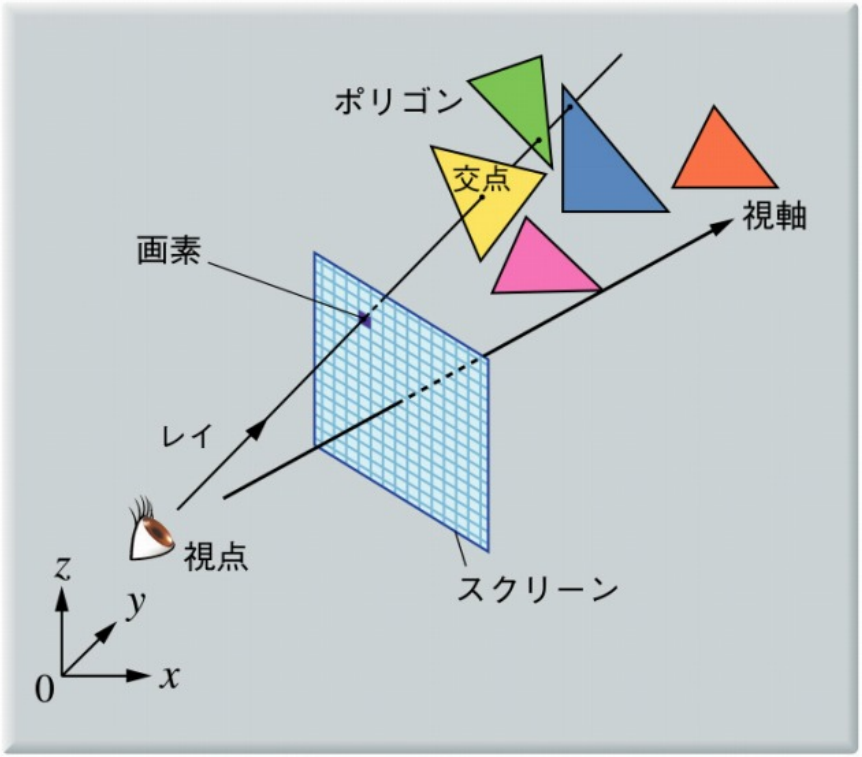
  ×計算コストが高い
  → GPU サポート (NVIDIA RTX シリーズ)



レイ トレーシング（光線追跡法）

- レイ(視線)と物体の交差判定による隠面消去
- 反射・透過・屈折を扱える

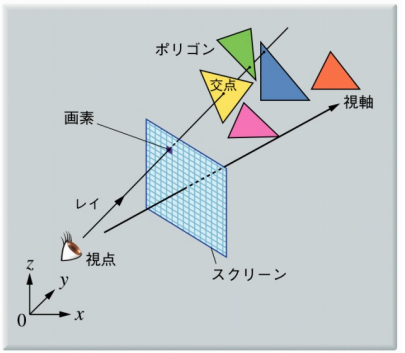
■図4.16——レイトレーシング法



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

レイトレーシング法のアルゴリズム

■図4.16——レイトレーシング法



「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

■アルゴリズム4.3——レイトレーシング法

```
スクリーンのすべての画素を順番にとり出し,
それぞれの画素について {
    視点から画素に向かうレイを決定する;
    レイとすべての物体との交差判定を行う;
    レイと交差する物体が1つ以上存在するならば {
        すべての交点のなかで最も視点に近いもの(可視点)を求める;
        その画素に可視点での物体の色を塗る;
    }
    そのほか(レイと交差する物体が存在しない)の場合 {
        その画素に背景色を塗る;
    }
}
```

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

レイと物体との交差判定

レイトレーシングでは、レイと物体の交点を求める必要がある

例：球との交差判定

$$||P_E + t\hat{E} - S|| = r$$

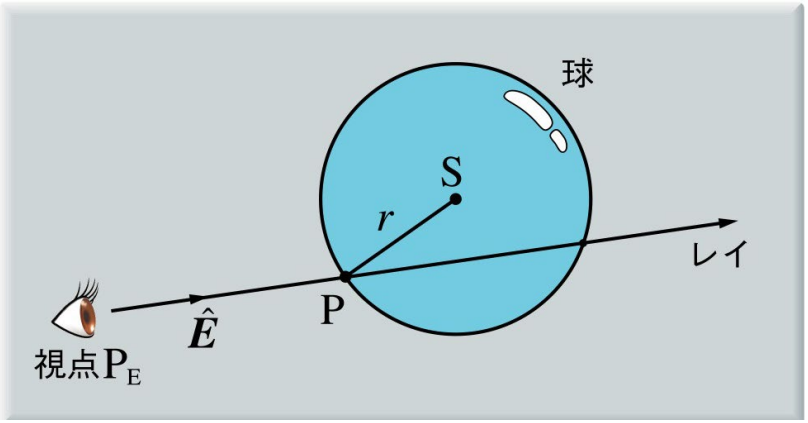
$$\left(P_E + t\hat{E} - S\right)_x^2 + \left(P_E + t\hat{E} - S\right)_y^2 + \left(P_E + t\hat{E} - S\right)_z^2 = r^2$$

⇩ 式変形

$At^2 + Bt + C = 0$の形

判別式 $D = B^2 - 4AC > 0$なら交点を持つ

$$t = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$



レイトレーシングによる表示例

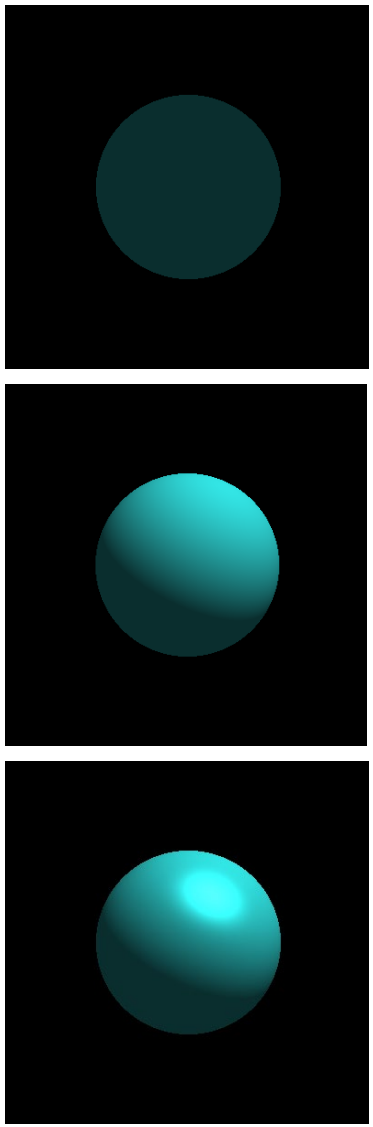
[a] 屈折率の違いによる画像の変化

■図4.43——屈折を考慮したレイトレーシング法による画像

「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

課題について

課題：球体をレンダリングする



サンプルコード実行結果

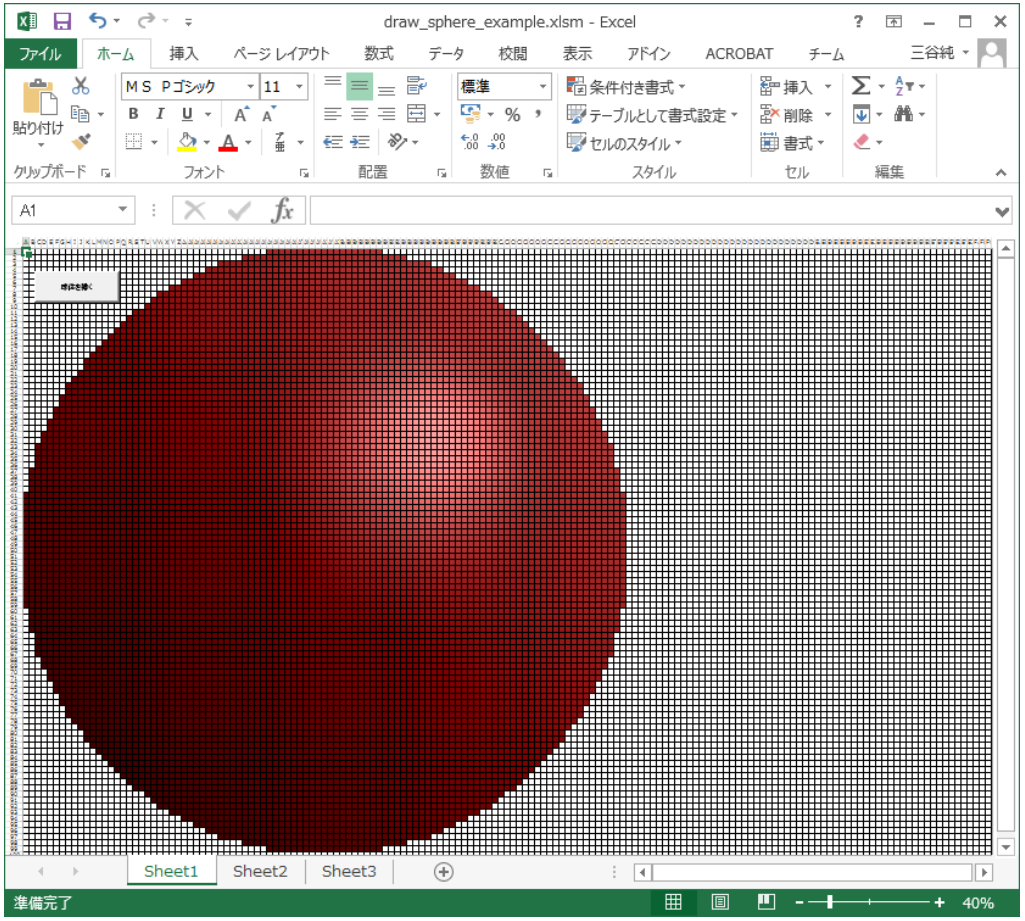
＋拡散(diffuse)反射モデル
（Lambert反射モデル）

＋鏡面(specular)反射モデル
（Phong反射モデル）

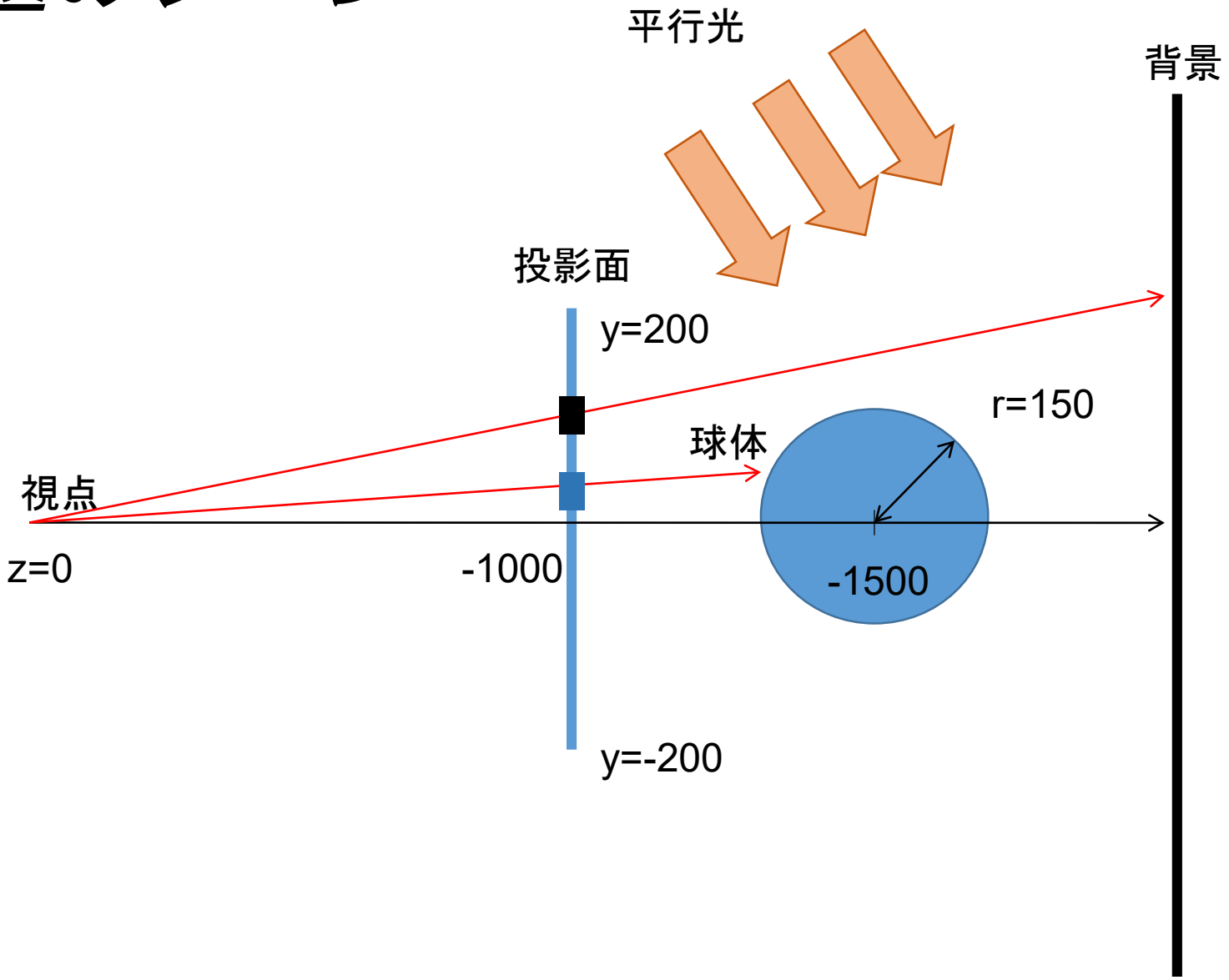
（雑談）

各画素（ピクセル）の色を計算で求めることで、陰影を含む立体を画面に描画する

Excelは、各セルの色を数値で指定できる
Excelでレンダリング！

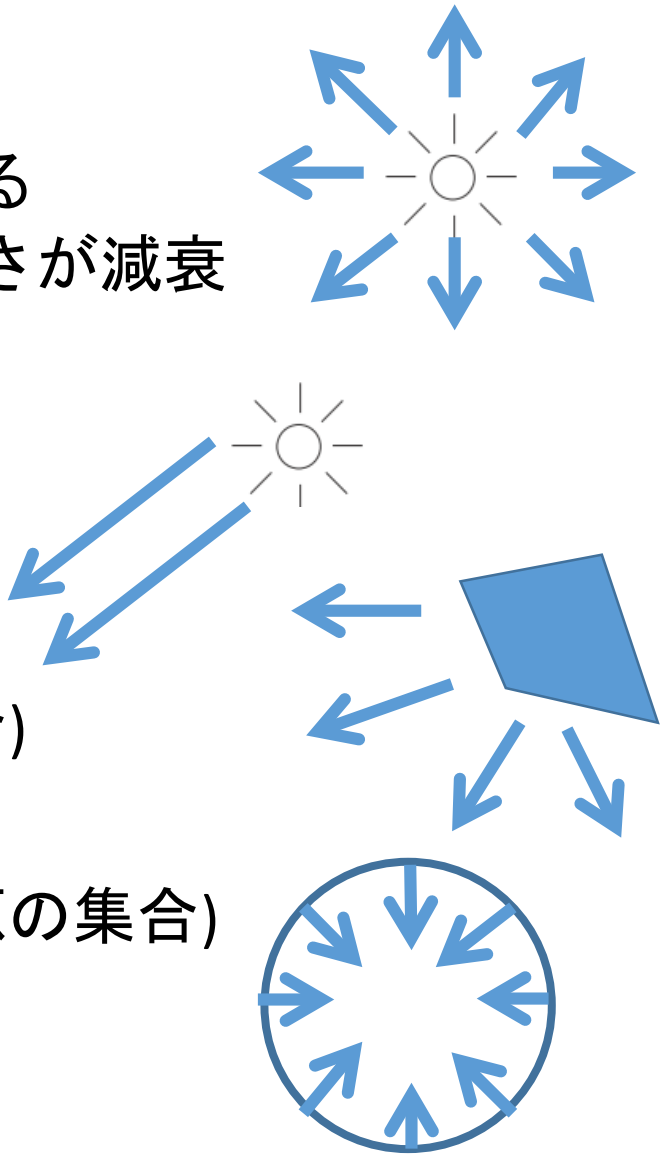


課題のシーン



再掲: CG でよく使われる光源の例

- 点光源
  - 仮想的な 1 点から光が放射される
  - 距離の 2 乗に反比例して光の強さが減衰
- 平行光源
  - 無限遠から光が届く (例: 太陽光)
  - 光の減衰なし
- 面光源
  - 面積を持った光源 (点光源の集合)
- 環境光源
  - 物体の周囲から届く光 (平行光源の集合)



レイトレーシング実装の流れ

- 全てのピクセルに対して以下を実行
  - 視点（原点）から投影面の各ピクセルを通るレイを飛ばす
  - レイと球との交点Pを求める
    - 交点がない（交差しない）場合は背景色を描画して終了
  - 点Pでの球面の法線ベクトルN（球の中心から点Pに向かうベクトル）を求める

  【拡散反射光の計算】
    - Nと光の方向との内積から、拡散反射光を計算し、反射光に加算

  【鏡面反射光の計算】
    - Nと光の方向から反射光の方向ベクトルを求め、視線ベクトルの内積から、鏡面反射光を計算し、反射光に加算