

コンピュータグラフィックス

第9回

レンダリング技法2

～シェーディング, マッピング～

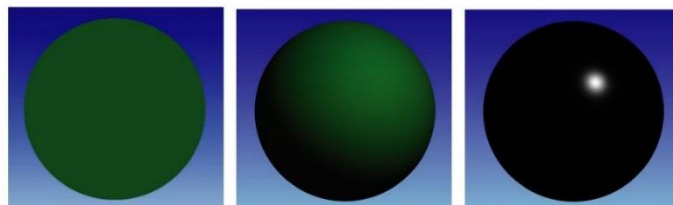
理工学部 兼任講師
藤堂 英樹

本日の講義内容

■ レンダリング技法 2

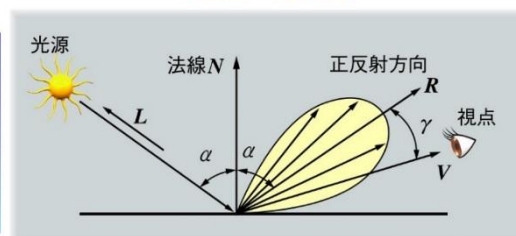
- シェーディング
- マッピング

■ 図4.29—環境光による反射、拡散反射、鏡面反射の各成分



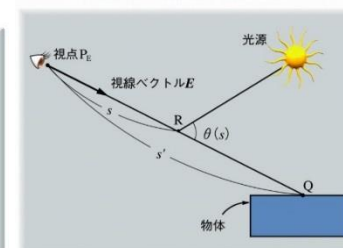
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■ 図4.38—鏡面反射光



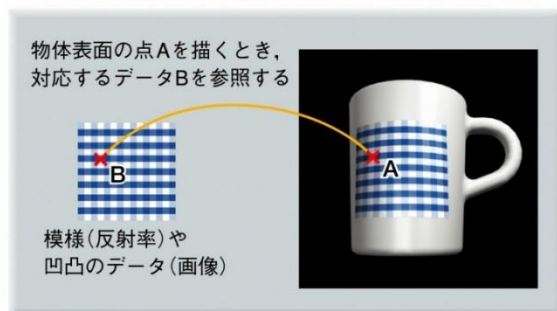
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■ 図4.44—散乱粒子による光の散乱



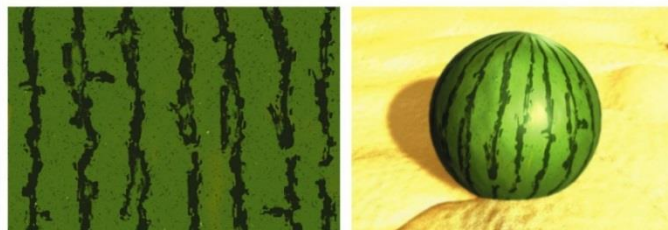
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■ 図4.64—(2次元) マッピングの考え方



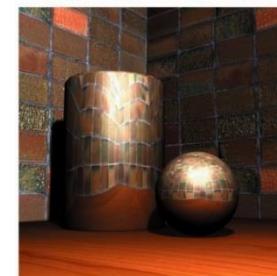
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

■ 図4.67—極座標変換を用いる方法によるテクスチャマッピングの例



「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)

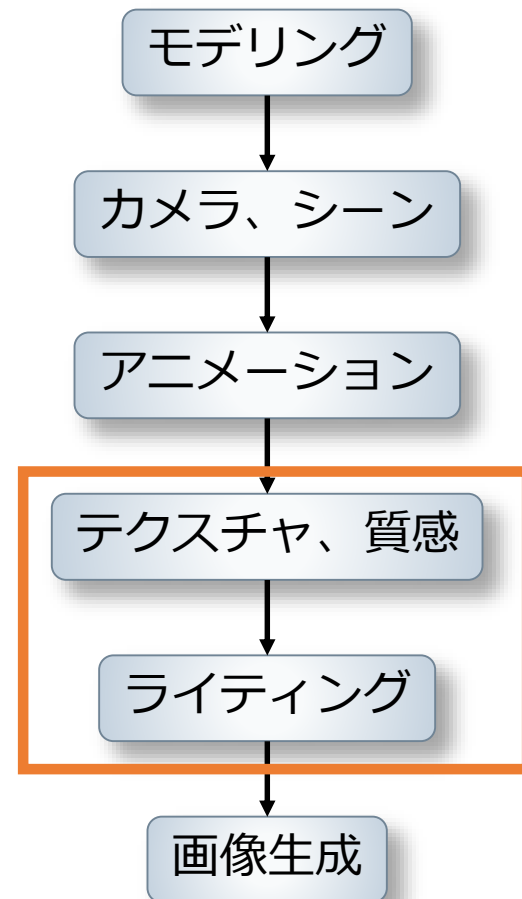
■ 図4.79—環境マッピングによる画像



「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)

CG制作の主なワークフロー

■3DCGソフトウェアの場合



実写との比較

■実写

- 質感: 人の肌質 + **化粧**で見栄えを調整
- ライティング
 - 太陽光に加えて**レフ板**で被写体の照明を調整

被写体

カメラ

レフ板

実写との比較

■CG

- 質感：**質感設定** + **テクスチャ**で見栄えの調整
- ライティング：
 - 各種**光源**を3次元空間上にデザイン



テクスチャ



レンダリングを構成する処理

■ 投影変換 (CG のための数学的基礎 2 参照)

- 透視投影, 平行投影等

■ 隠面消去 (前回の講義で説明)

- 見えない面を消す処理

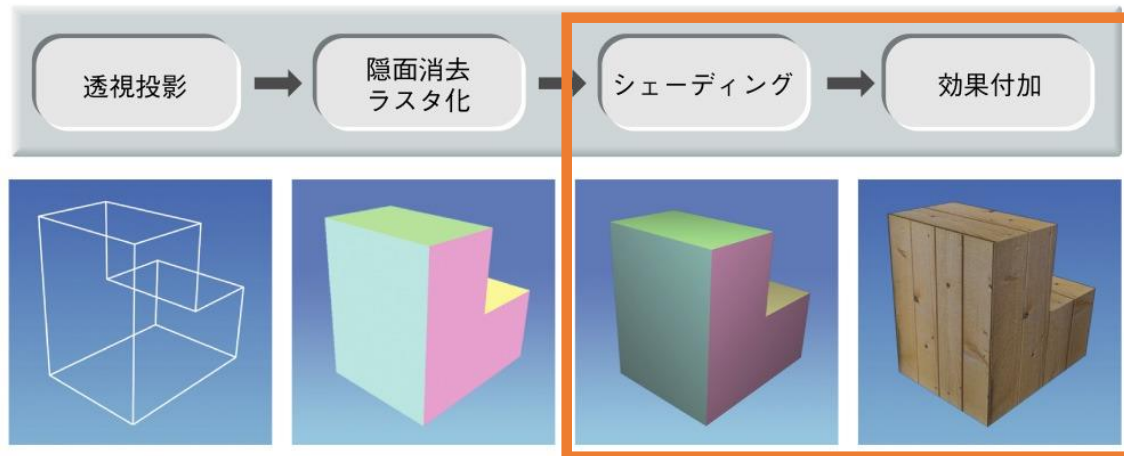
■ シェーディング

- 質感の表現

■ 効果付加

- テクスチャ

■ 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)

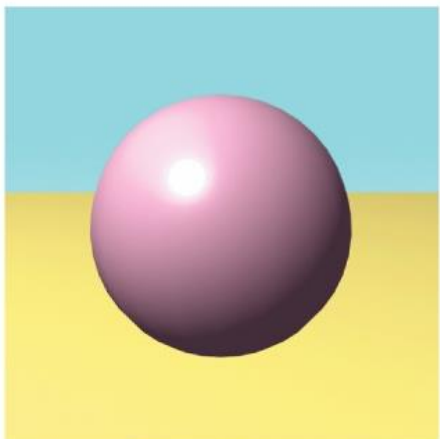
シェーディングの基礎と概要

■シェーディング

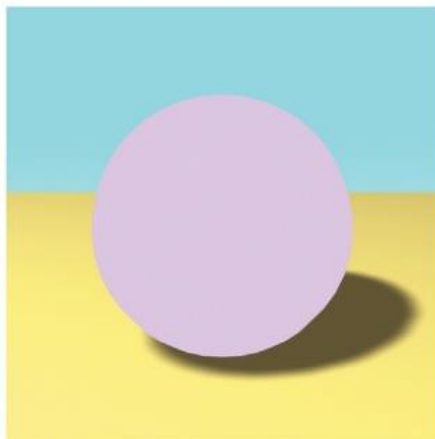
- 拡散反射：つやがない質感の反射
- 鏡面反射：ハイライト

■シャドウ

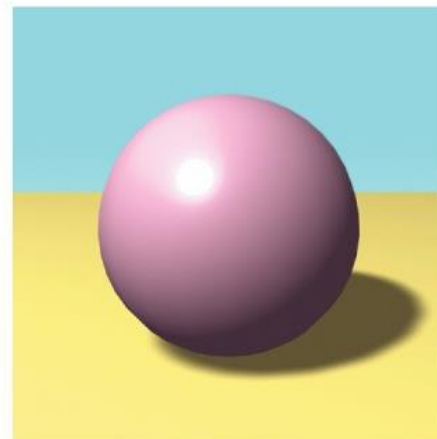
- 物体によって光が遮られた領域



[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

シェーディングモデル

■照明モデル, ライティングモデルとも呼ばれる

■局所照明モデル

- 直射光に対して物理的なモデルで照明計算
- 簡略化した計算で高速に表示が可能

■大域照明モデル

- 間接光までを含めた照明計算
- 写実的な画像の表現

光源の種類

■ 平行光源

- 太陽光線のように光線が同一方向に進む

■ 点光源

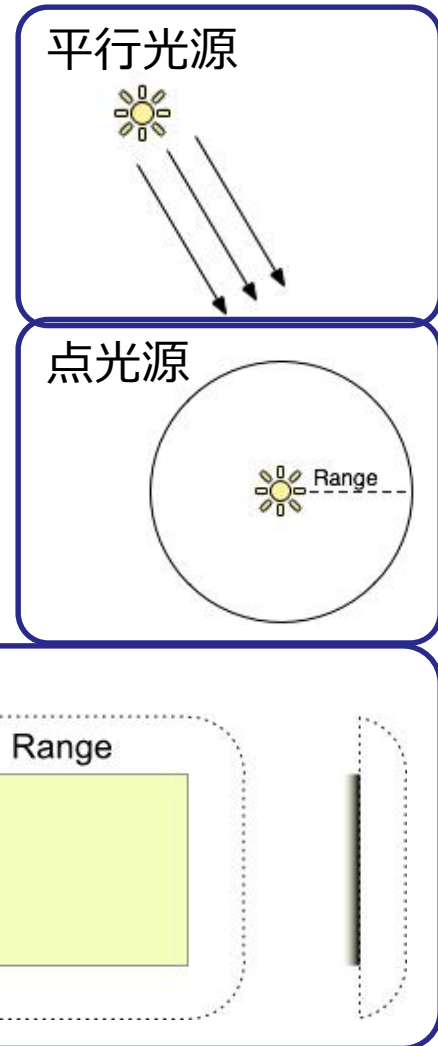
- ある点から全方向を等しく照らす

■ 線光源

- 長さを持つ光源

■ 面光源

- 領域を持つ光源



直射光と間接光

■ 直射光

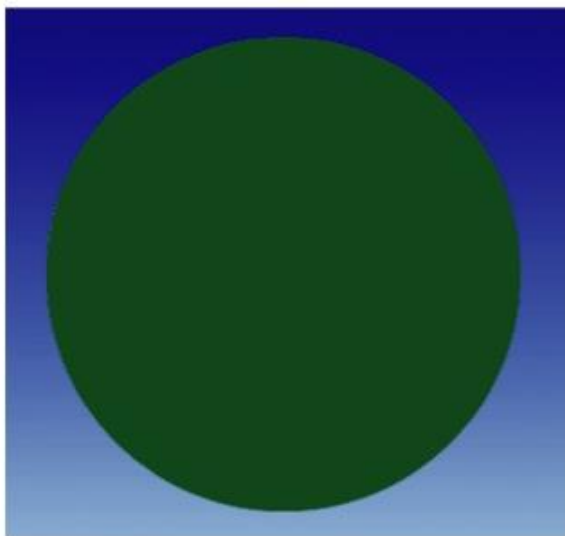
- 光源から出た**光が直接**物体に到達
- **局所照明モデル**は直射光のみの表現

■ 間接光

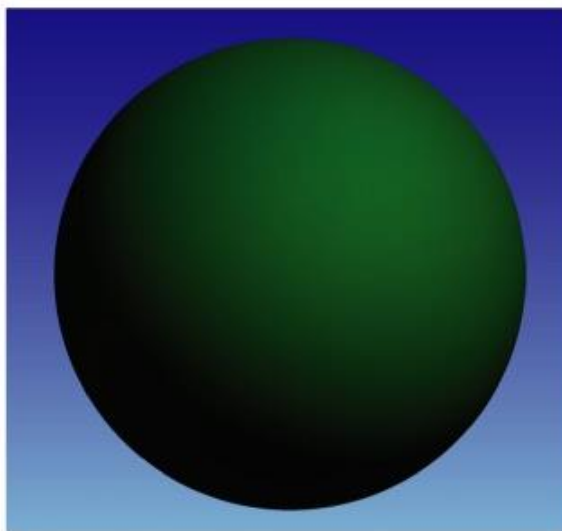
- いったん**他の物質との相互作用**をおこし，物体に到達
- **大域照明モデル**は間接光まで計算に含める

主な直射光の表現

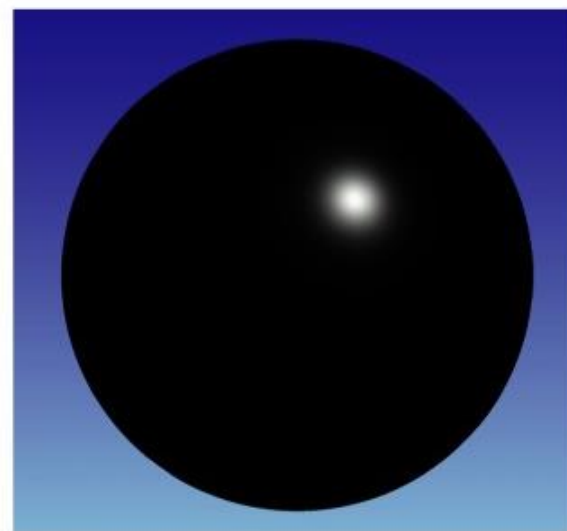
- 環境光: 周囲からくる**一様な光**をモデル化
- 拡散反射光: **視点に依存しない**反射光
- 鏡面反射光: 光が面で反射して見える**ハイライト**



[a] 環境光



[b] 拡散反射光

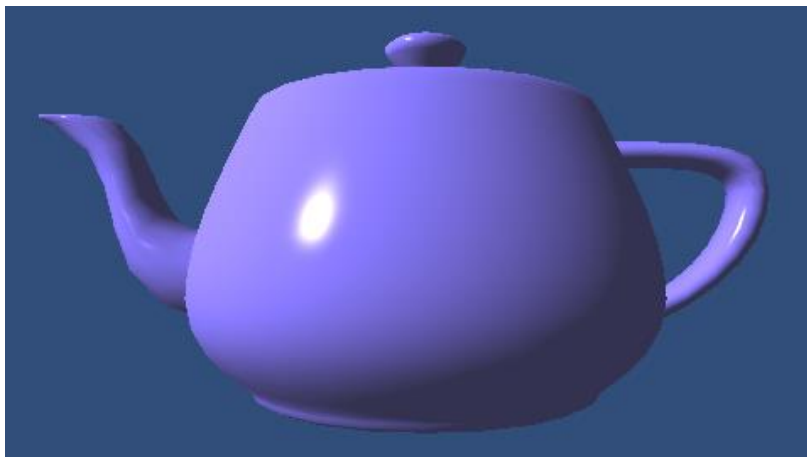


[c] 鏡面反射光

環境光

■ 周囲からくる一様な光

- 直射光が当たっていない間接光部分を近似
- $I = k_a I_a$
- k_a : 環境光に対する反射率
- I_a : 環境光に対する強さ



平行光源



環境光

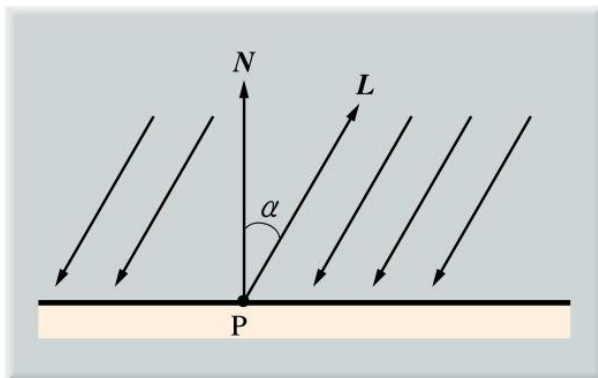
拡散反射

■ どの方向から見ても反射の仕方が一定

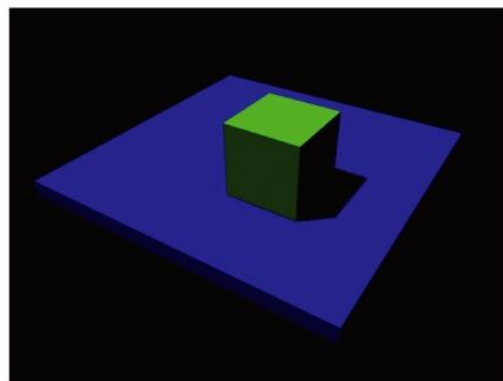
- つやの無い質感の反射

■ 平行光源

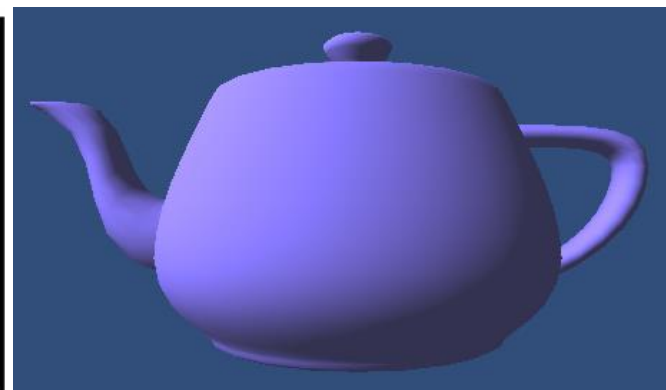
- $I = k_d I_i (\hat{N} \cdot \hat{L})$
- k_d : 拡散反射率, I_i : 入射光の強さ
- $\hat{N} \cdot \hat{L}$: **入射光と面の向きとなす角**



[a] 平行光線による拡散反射成分の計算



[b] 平行光線により照射された物体



拡散反射光

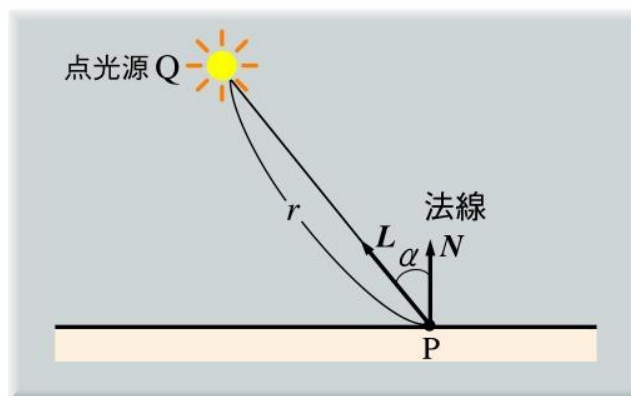
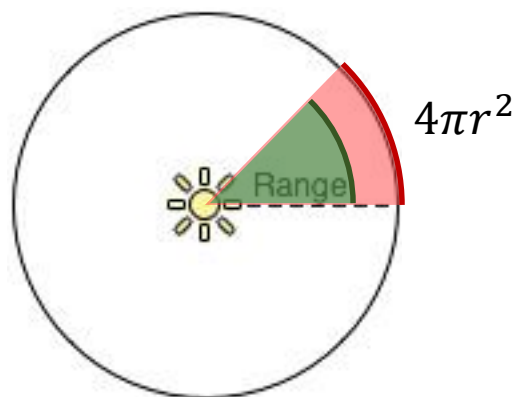
■点光源

- 光が光源から放射状に広がる
⇒ 光の強度は光源からの距離の 2 乗に反比例

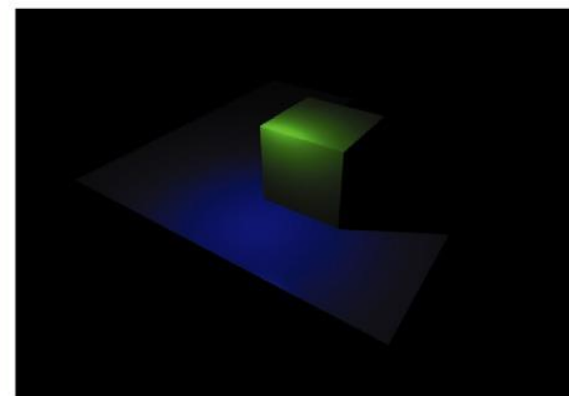
- $$I = \frac{k_d I_q}{r^2} (\hat{N} \cdot \hat{L})$$

- I_q : 点光源の強度, r : 光源から点までの距離

■ 図4.34——点光源による拡散反射



[a] 点光源による拡散反射成分の計算



[b] 点光源により照射された物体

拡散反射光

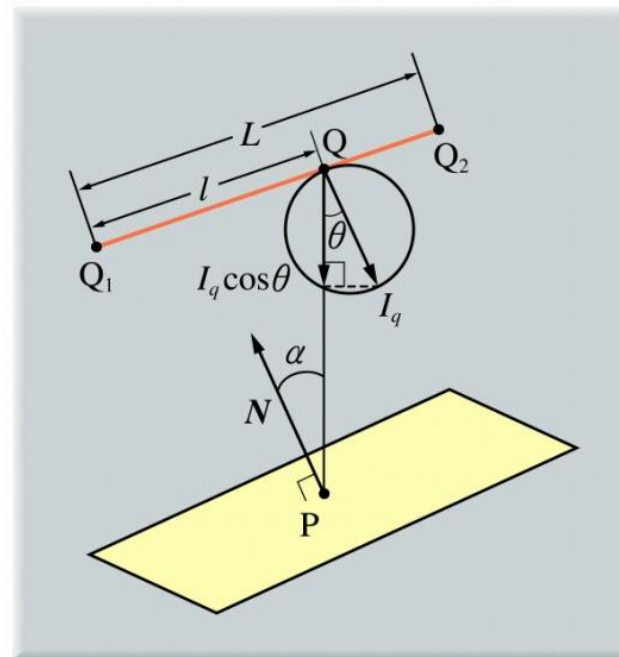
■線光源

- 蛍光灯のように細長い光源
- **点光源が直線上に並んでいるものとして計算**

- $$I = k_d I_q \int_0^L \frac{\cos \theta}{r^2} \cos \alpha \, dl$$

- I_q : 単位長さあたりの光度
- L : 線光源の長さ

■図4.36—線光源による拡散反射成分の計算



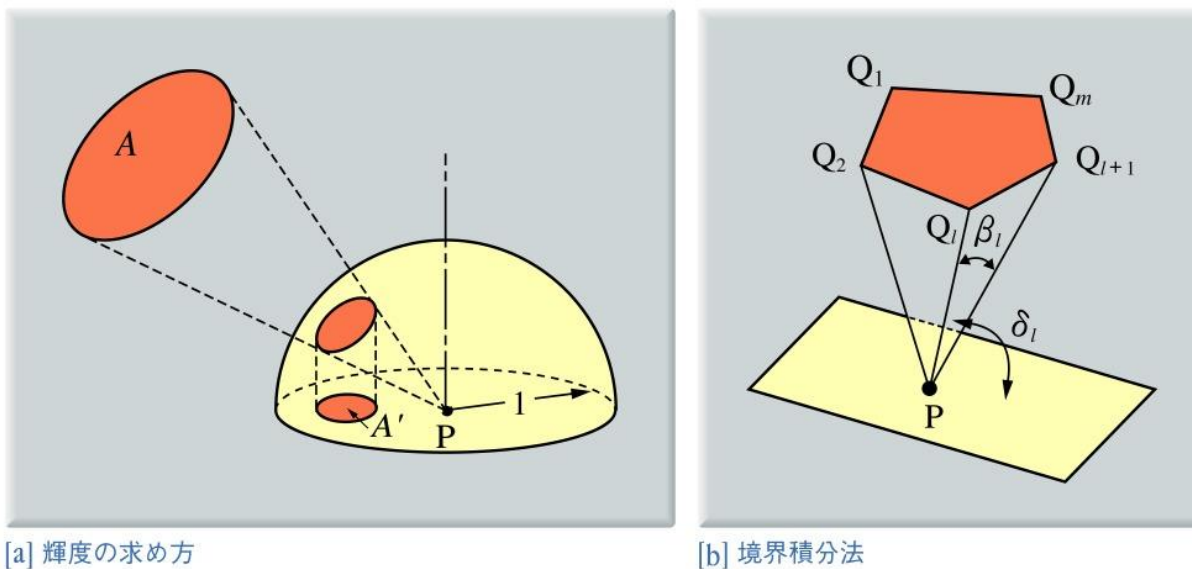
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

拡散反射光

■面光源

- 現実の光源の面積を考慮
- 微小面積を点光源として考え，積分計算

■図4.37——面光源による拡散反射成分の計算



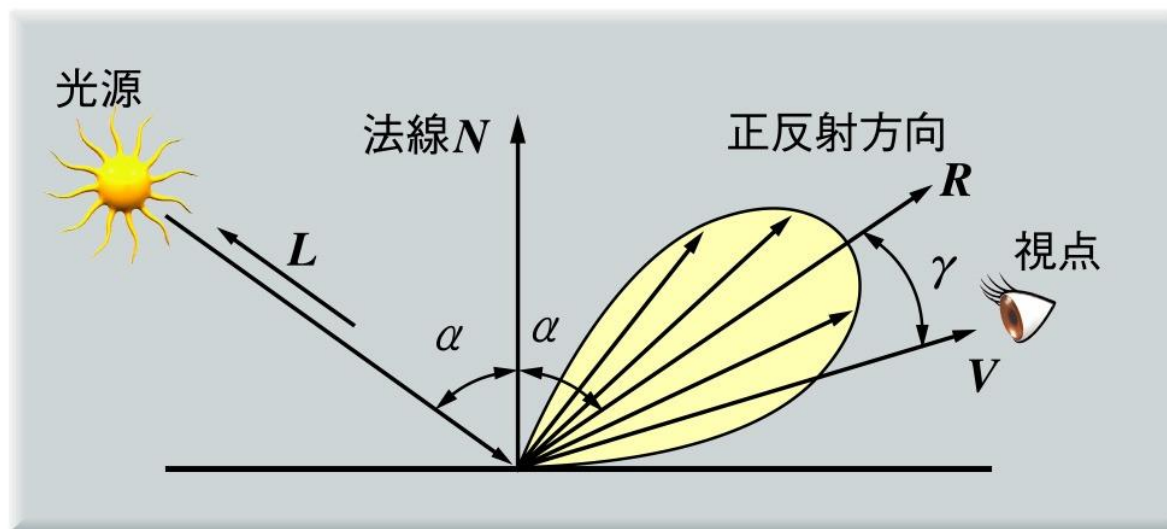
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

鏡面反射

■物質表面での直接反射

- ハイライト部分
- 光沢のあるプラスチックや金属表面

■図4.38——鏡面反射光



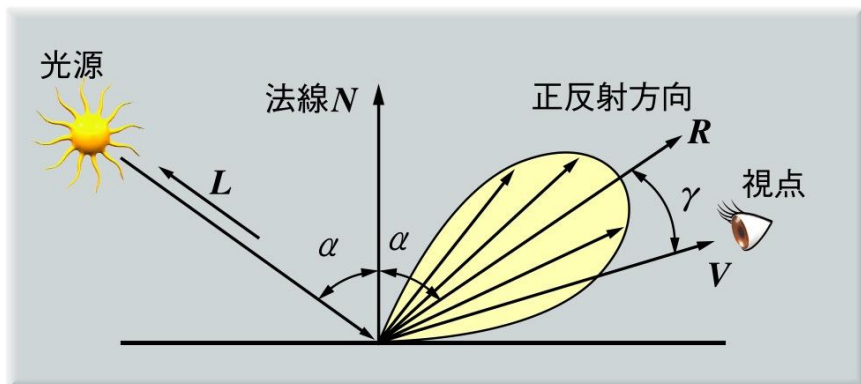
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

鏡面反射

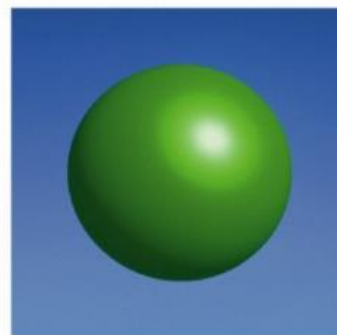
■ フォンのモデル

- 視点方向と正反射方向のなす角 γ により定義
- $I = k_s I_i (\hat{R} \cdot \hat{V})^n$
- k_s : 鏡面反射率
- n : ハイライトの鋭さ

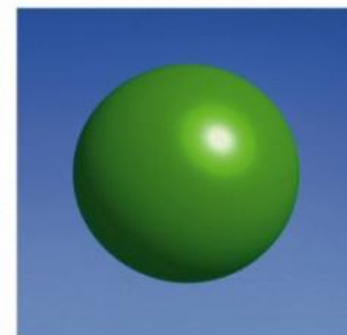
■ 図4.38——鏡面反射光



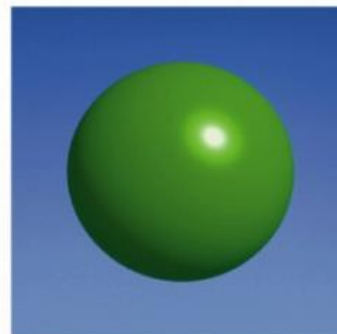
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)



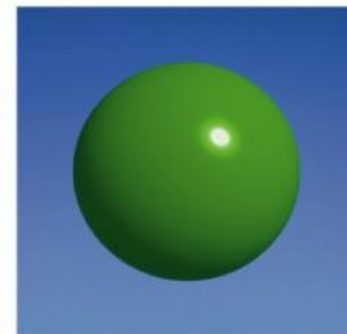
[a] $n = 2$



[b] $n = 5$



[c] $n = 10$

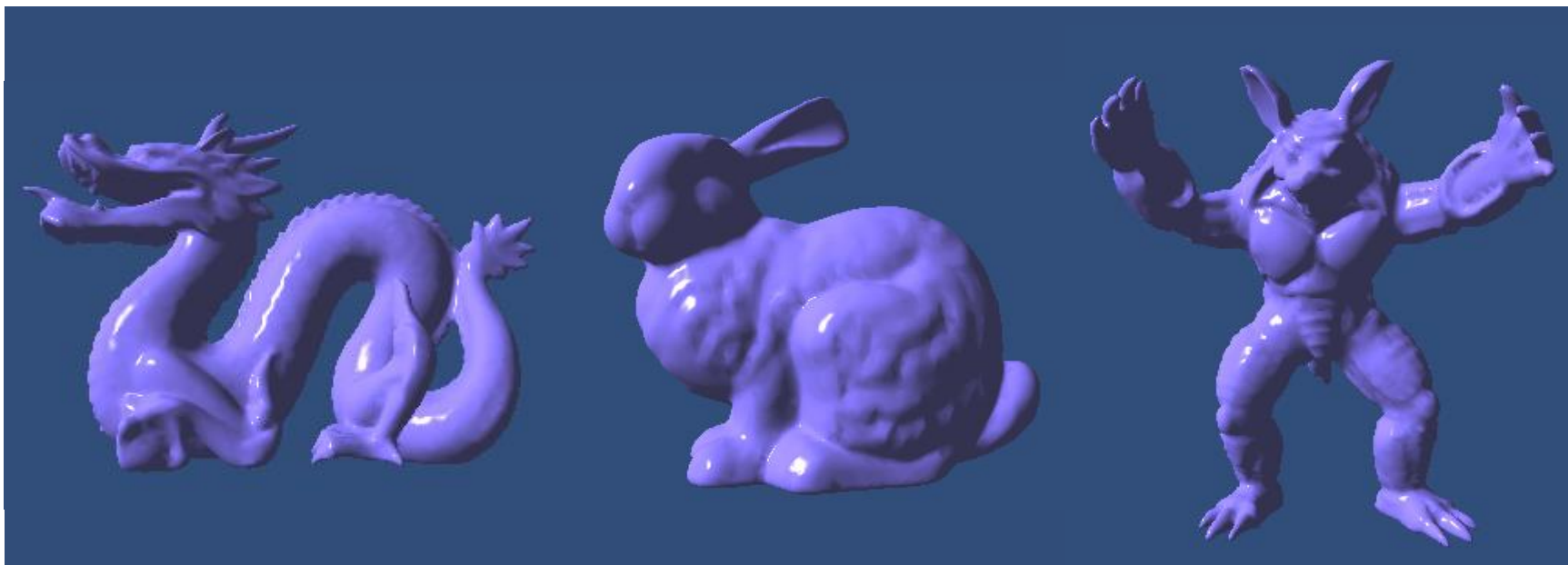


[d] $n = 20$

鏡面反射

■ 色々なモデルでの反射の様子

- Unityでのデモ

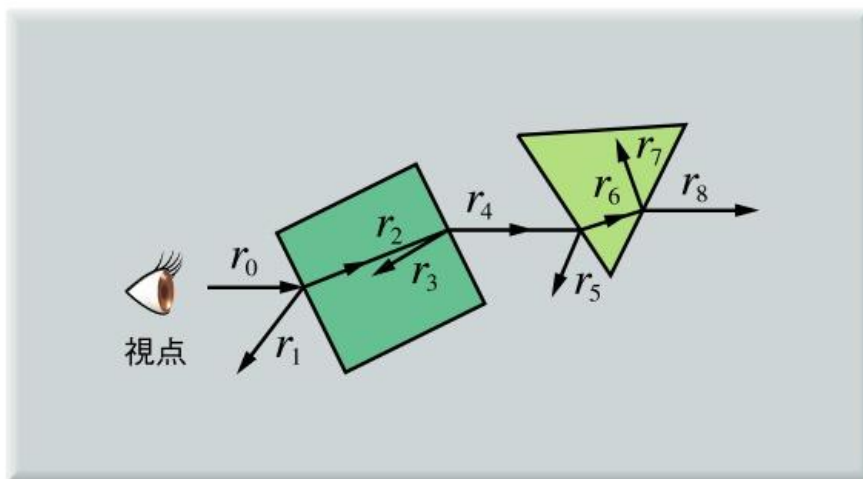


完全鏡面反射・透過・屈折

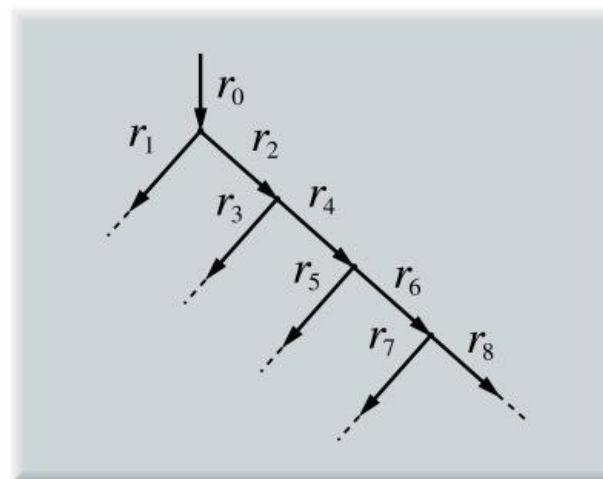
■ レイトレーシング法が用いられる

- 物体とレイが交点を持つ
⇒ 反射・屈折用のレイが新たに生成される

■ 図4.40——反射・屈折の表現と二分木表現との関係



[a] 光の反射と屈折



[b] 光線の追跡過程の二分木表現

「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)

完全鏡面反射・透過・屈折

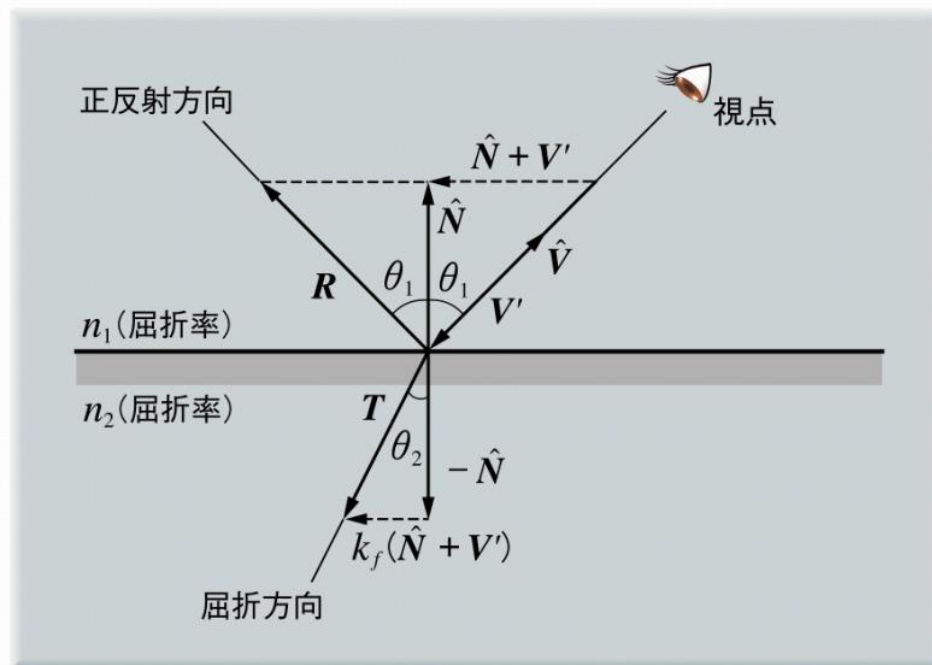
■ 正反射方向ベクトル

- 面の法線に対して視線が反射

■ 屈折方向ベクトル

- 透過した光の屈折方向
- **屈折率の異なる境界**で光の方向が曲がる
- スネルの法則
 - $n_1 \sin \theta_1 = n_2 \sin \theta_2$

■ 図4.41——反射・屈折方向の計算

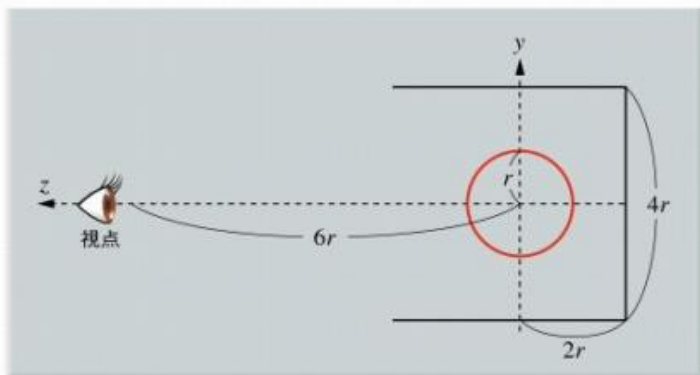


「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会 (CG-ARTS協会)

完全鏡面反射・透過・屈折

■ 反射率と透過率

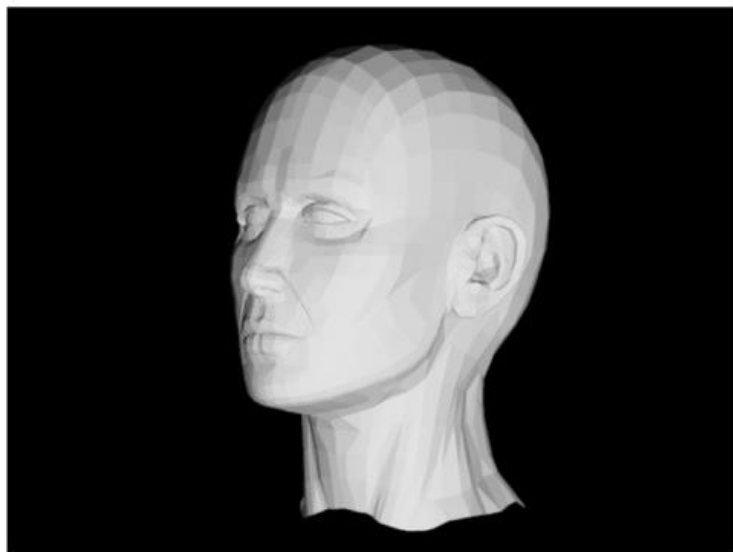
- 異なる屈折率をもつ媒質の境界では**反射**も起こる
- 反射率: $k_r \Rightarrow$ 透過率: $k_t = 1 - k_r$
- $I_v = k_r I_r + k_t I_t$
- I_v : 入射方向への光, I_r : 反射光, I_t : 透過光



スムーズシェーディング

■ コンスタントシェーディング

- ポリゴン面毎に明るさを計算して表示
- ポリゴンがなめらかでない場合
⇒ 陰影がかくかくして見える



[a] コンスタントシェーディングによる画像

スムーズシェーディング

■スムーズシェーディング

- ポリゴン内の明るさを補間
⇒滑らかな陰影表示



[a] コンスタントシェーディングによる画像

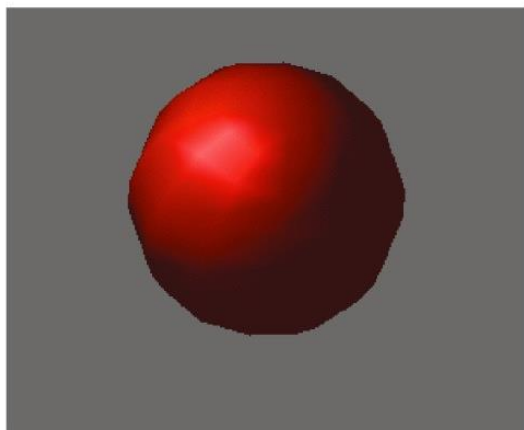


[b] スムーズシェーディングによる画像

スムーズシェーディング

■ グローシェーディング

- **輝度の補間**によるスムーズシェーディング
- 1. **頂点毎に明るさ**を計算
- 2. 頂点の明るさを**ポリゴン上で補間**
- 問題点: ハイライトがスムーズにならない

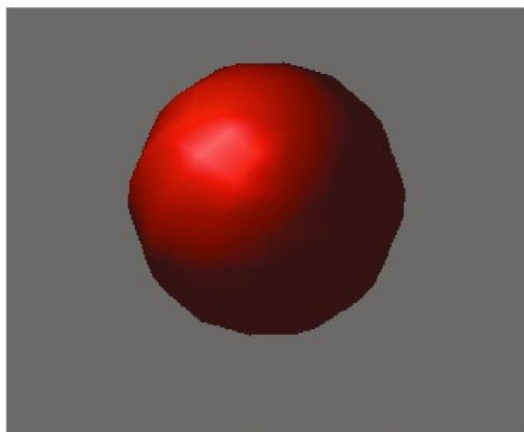


[a] グローシェーディング(輝度を内挿する)

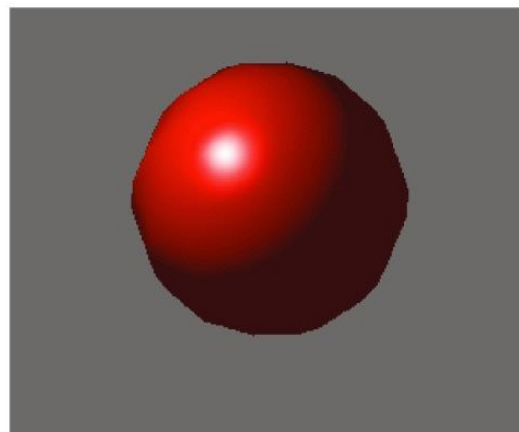
スムーズシェーディング

■ フォンシェーディング

- **法線の補間**によるスムーズシェーディング
- 1. **頂点毎に法線ベクトル**を計算
- 2. 法線ベクトルを**ポリゴン上で補間**
- 3. 明るさを補間した法線で計算
- グローシェーディングよりも綺麗な陰影表示



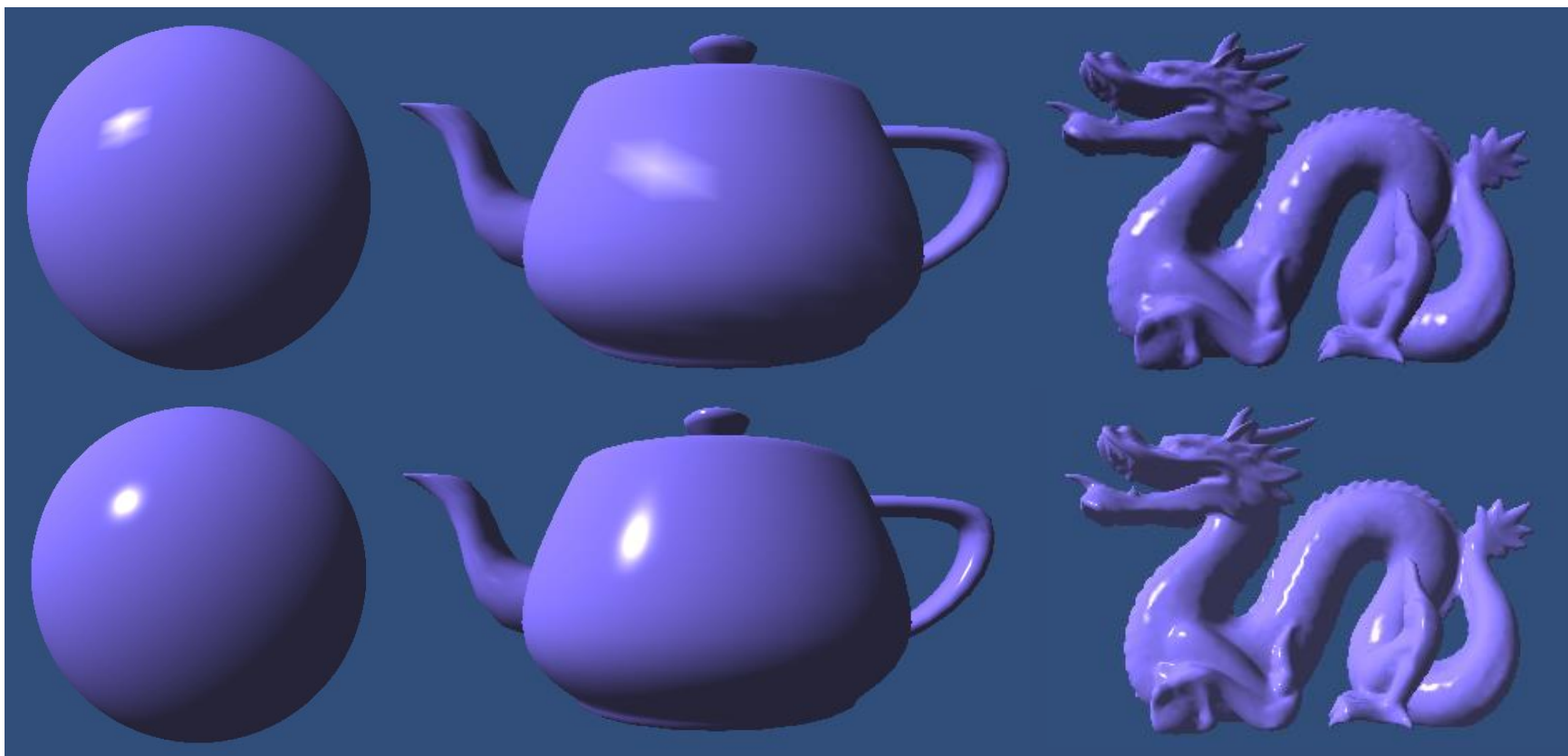
[a] グローシェーディング (輝度を内挿する)



[b] フォンシェーディング (法線ベクトルを内挿する)

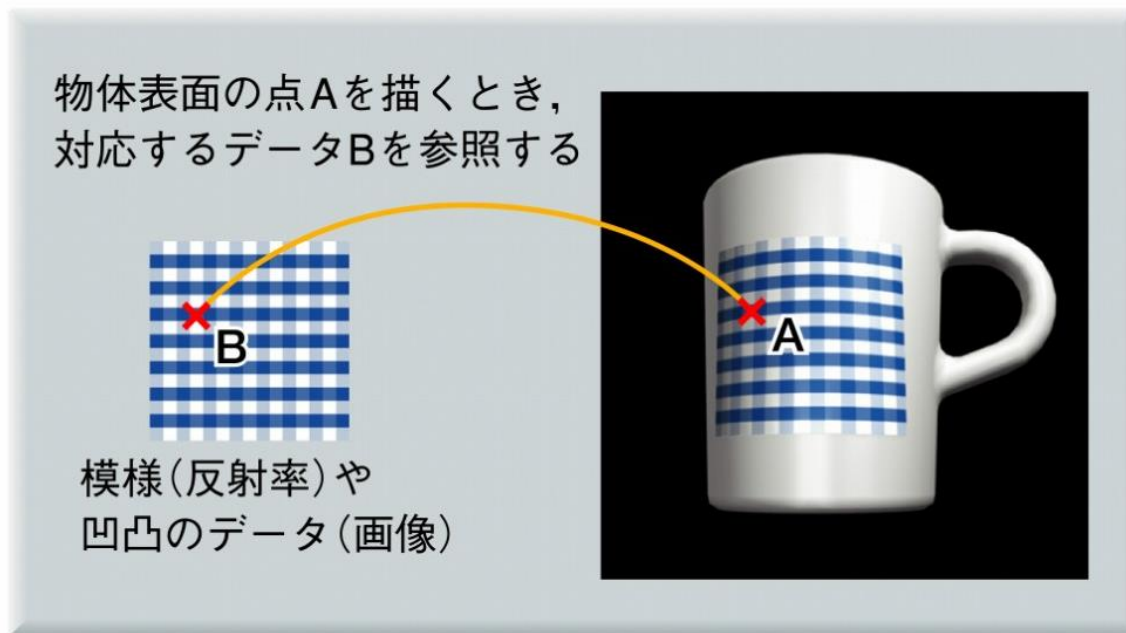
スムーズシェーディング

■ 色々なモデルでのスムーズシェーディングの違い



マッピング

- **模様や細かい凹凸**は3次元形状で表現しにくい
⇒ 模様や凹凸を付加する**マッピング**が考案
 - **3次元物体と画像**を対応付け，効果を付加する
 - リアルな画像を簡単に作成できる



テクスチャマッピング

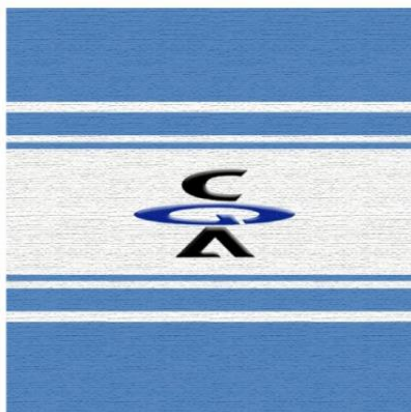
■ テクスチャ:

- マッピングで貼り付ける**模様や画像**

■ マッピング:

- テクスチャを3次元物体に貼り付ける処理

■ 図4.65——テクスチャマッピングの例



[a] テクスチャ

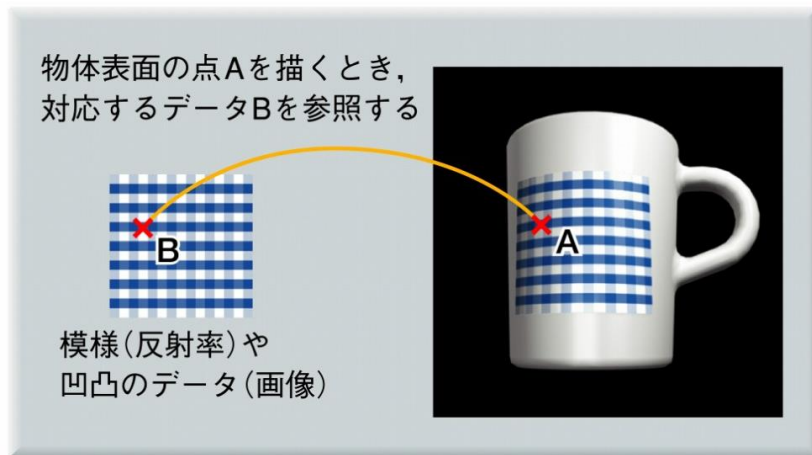


[b] 曲面上へのマッピング

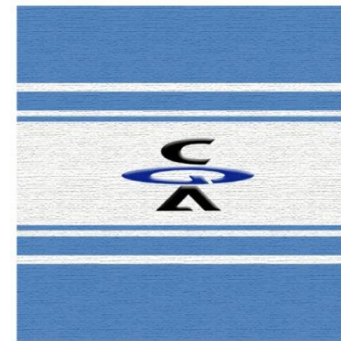
テクスチャマッピングの種類

■パラメータ座標によるマッピング

- 通常の2次元画像を貼り付けるマッピング
- 曲面の uv パラメータとも相性が良い



■図4.65——テクスチャマッピングの例



[a] テクスチャ



[b] 曲面上へのマッピング

テクスチャマッピングの種類

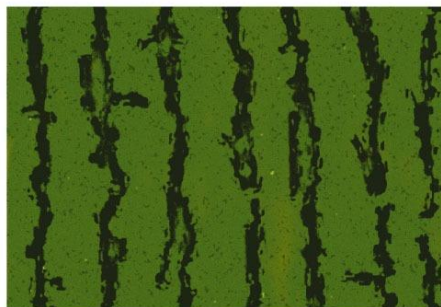
■ 投影によるマッピング

- プロジェクタで画像をスクリーンに投影
- マッピングをアニメーション可能

■ 極座標によるマッピング

- 球面や円柱へのマッピング方法

■ 図4.67——極座標変換を用いる方法によるテクスチャマッピングの例



[a] すいかの模様の画像

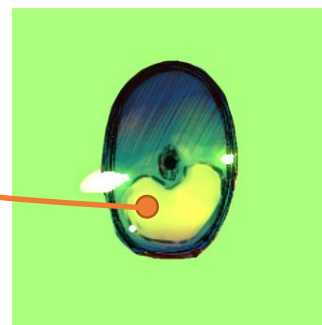
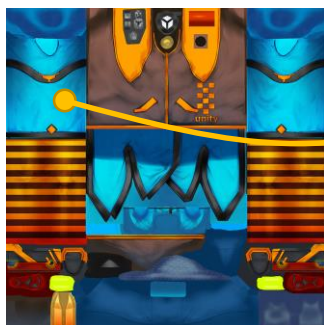
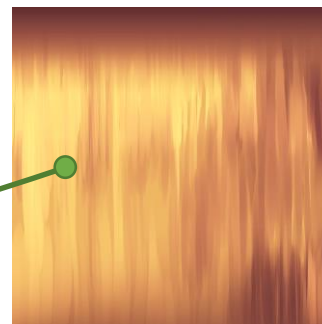
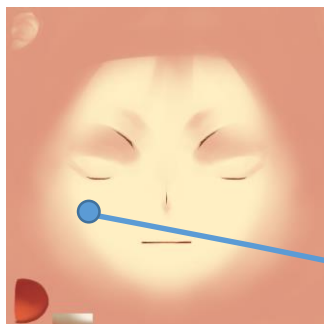


[b] 球面へのマッピング

テクスチャマッピングの種類

■ 実際のモデル: Unityちゃん

- **アーティスト**がパラメータ座標をデザイン
- 綺麗に画像を貼るためには**技術と手間**が必要



実制作でのテクスチャ座標のデザイン

- 綺麗な画像を貼り付けるために必要な処理
 - 座標が重ならない
 - 歪みになるべく少なく
 - 不連続部分になるべく少なく
 - ツールの自動的なサポートもあるが...
⇒細かい手作業を要求される

テクスチャデザインを効率化する工夫

■ Ptex (© Walt Disney Animation Studios)

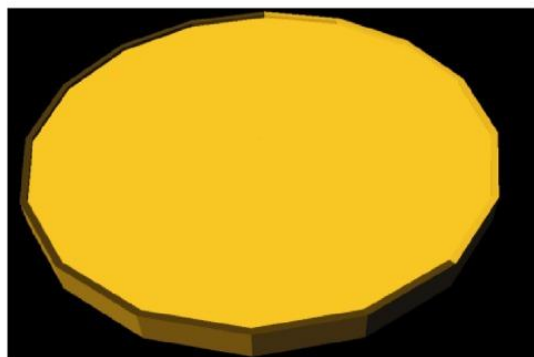
- ポリゴン上に**小さな四辺形テクスチャ領域**を多数割り当ててタイリング
- パラメータ座標のデザインを必要としない
- メッシュの解像度に依存しない

バンプマッピング

■ 模様の代わりに凹凸を物体表面上にマッピング

- テクスチャに応じて**面の法線**を変化させる
- P_u : u 方向の接線ベクトル, P_v : v 方向の接線ベクトル
- 法線ベクトル: $N = P_u \times P_v$
- $N' = N + F_i N \times P_v - F_j N \times P_u$
- F_i, F_j : テクスチャから得られた法線ベクトルの揺らぎ

■ 図4.76——バンプマッピングによる画像



[a] 表面がフラットなコイン形状



[b] バンプマッピングを施した画像



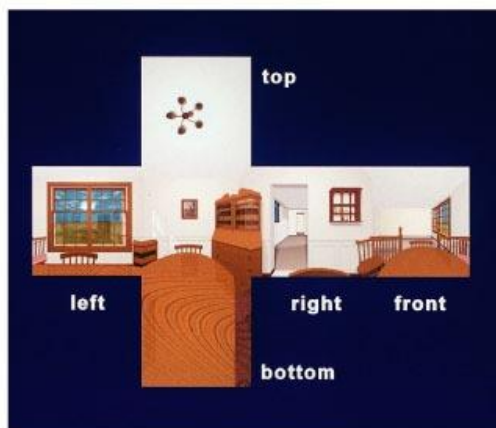
[a] バンプマッピング

環境マッピング

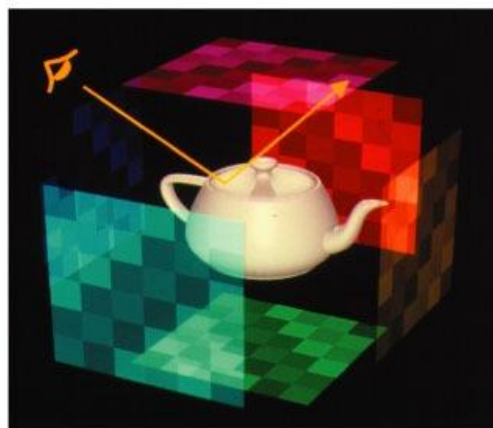
■周囲の映り込みを疑似的にマッピングで表現

- 1. 表示する物体を取り囲む球や立方体を仮想的に配置
- 2. 仮想球や立方体の内面に周囲の環境を貼り付ける
- 3. 物体表面で反射したレイと仮想球との交点を求める
- 4. 交点でのテクスチャの色を求める

■図4.78——環境マッピングの処理方法



[a] 6枚のテクスチャ



[b] テクスチャの対応点の算出

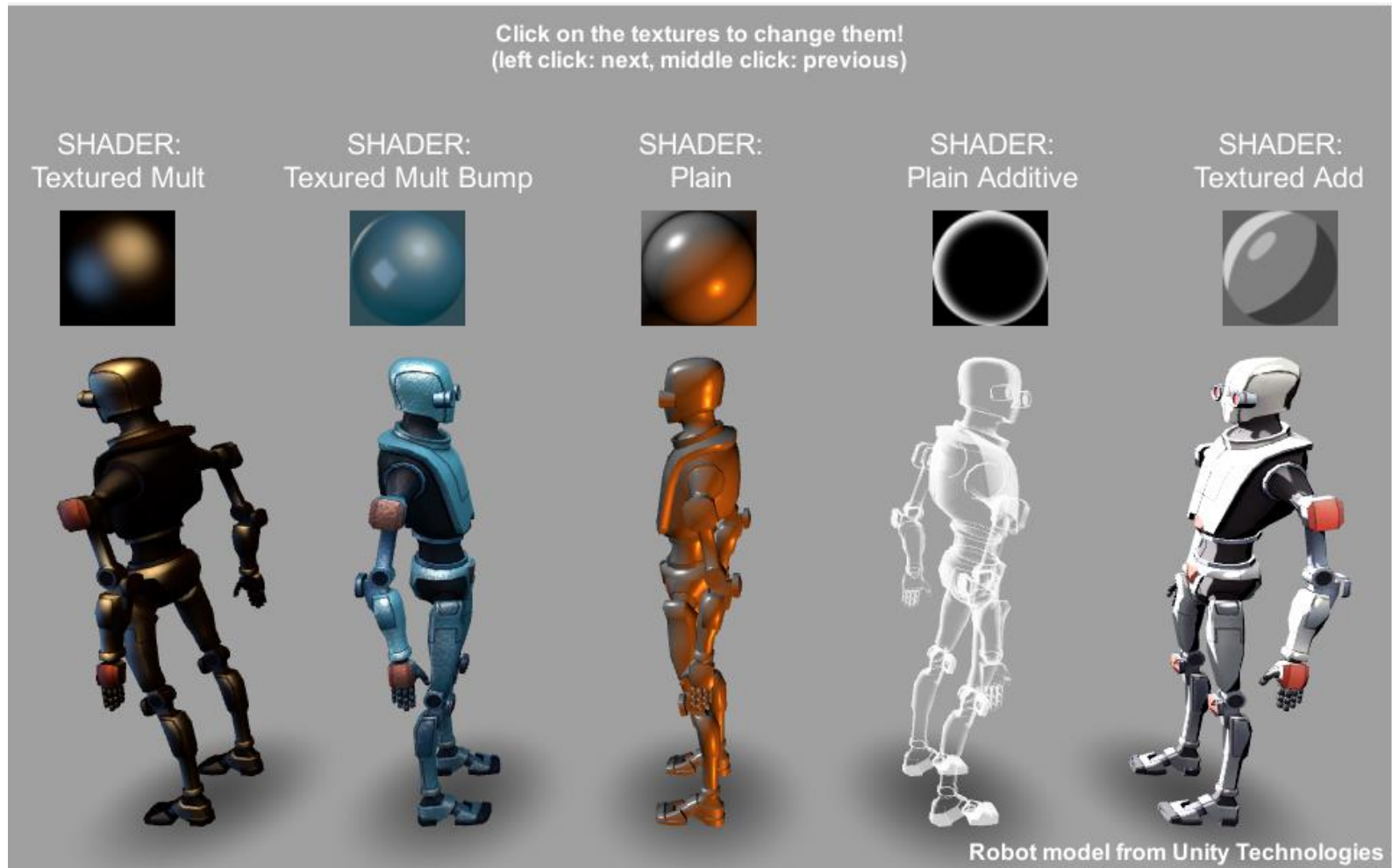


[c] 環境マッピングを施したティーポット

LightCap (MatCap) © Pixologic, Inc.

- 投影によるマッピングや環境マップを利用して複雑な陰影を画像ベースで行う手法

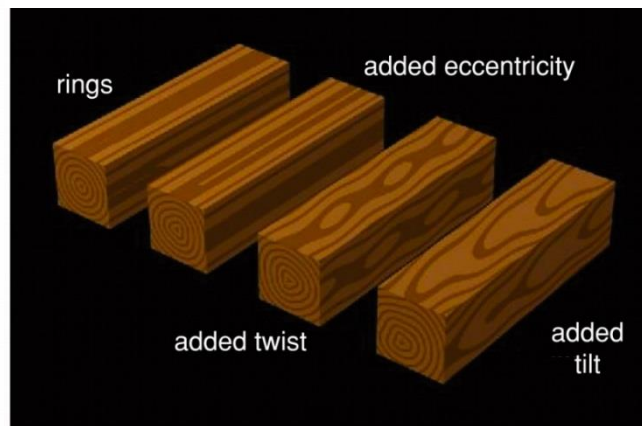
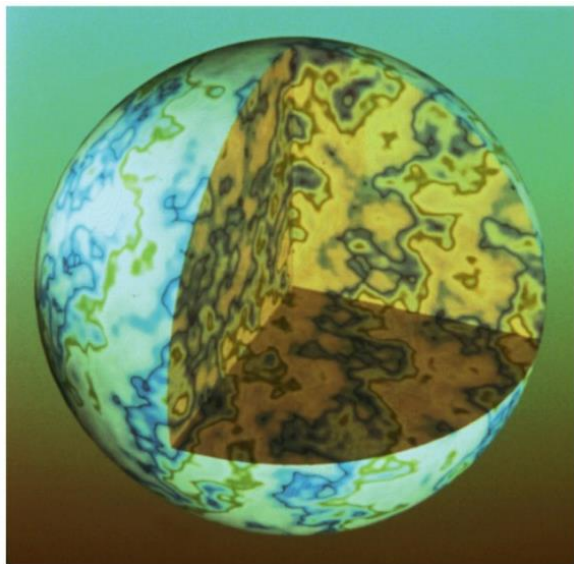
MatCap demo (Unity) © Jean Moreno



ソリッドテクスチャリング

■ 3次元空間でテクスチャを定義

- 任意の切断した面でもリアルな画像を表現
- 画像として準備するのは**データ量が膨大**
⇒ **3次元空間の関数**として定義するのが一般的



ソリッドテクスチャリングの実用例

- レミーのおいしいレストラン ©Disney, Pixar
 - パンの中身の見た目を表現するのに利用
 - 他にも料理をおいしく見せるための工夫がたくさん行われている

次回

■ レンダリング技法3

～写実的な陰影表現，大域照明モデル～

