コンピュータグラフィックス基礎

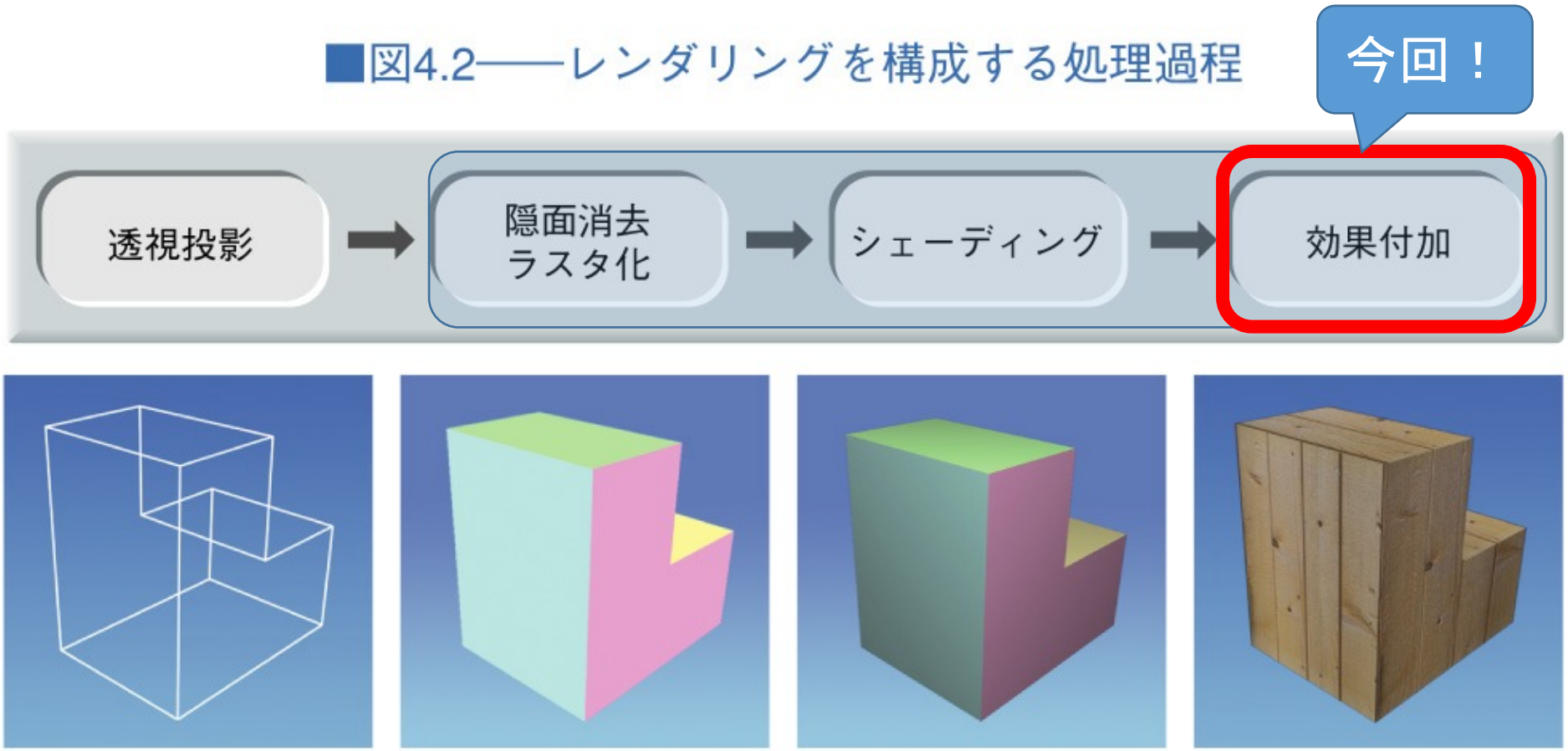
第9回 レンダリング (2)

学習の目標

- 各種マッピングの手法について学習する
- 大域照明の概念を理解する
- レイトレーシング法を用いた光の反射や屈折の表現を理解する

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）

マッピング

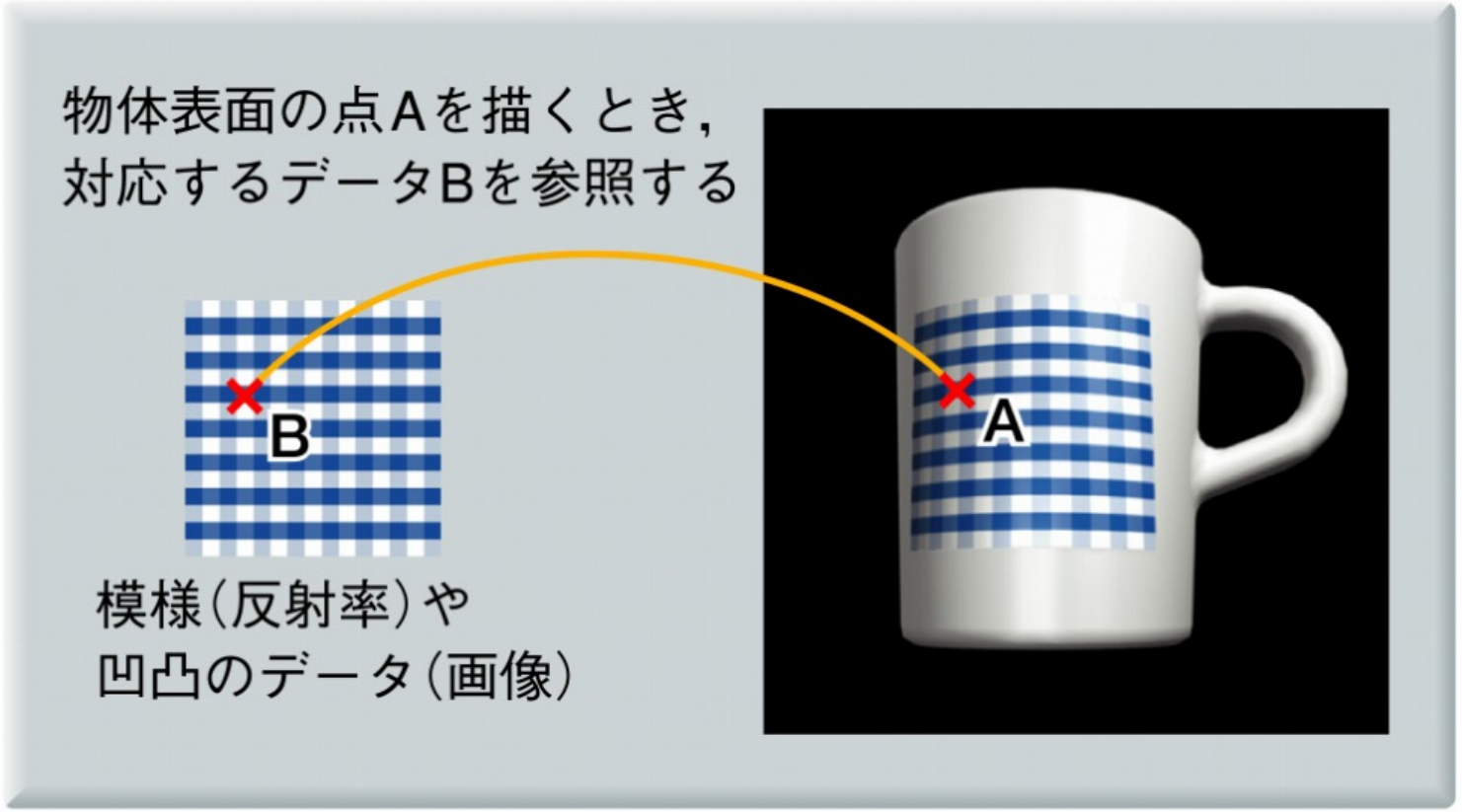
マッピング

- 3次元物体の表面に様々なパターンや模様をマッピング（写像）し、表示を行う手法
- 模様や凹凸などを効率的に表現する
- マッピング手法
  - ２Ｄ平面上で定義される画像や関数を3次元物体の表面にマッピングする
    - テクスチャマッピング、バンプマッピング等
  - 3次元空間で定義されるパターンや関数を3次元物体内部にマッピングする
    - ソリッドテクスチャリング

テクスチャマッピング

- ２Ｄテクスチャ画像を3次元物体の表面に貼り付けて表示

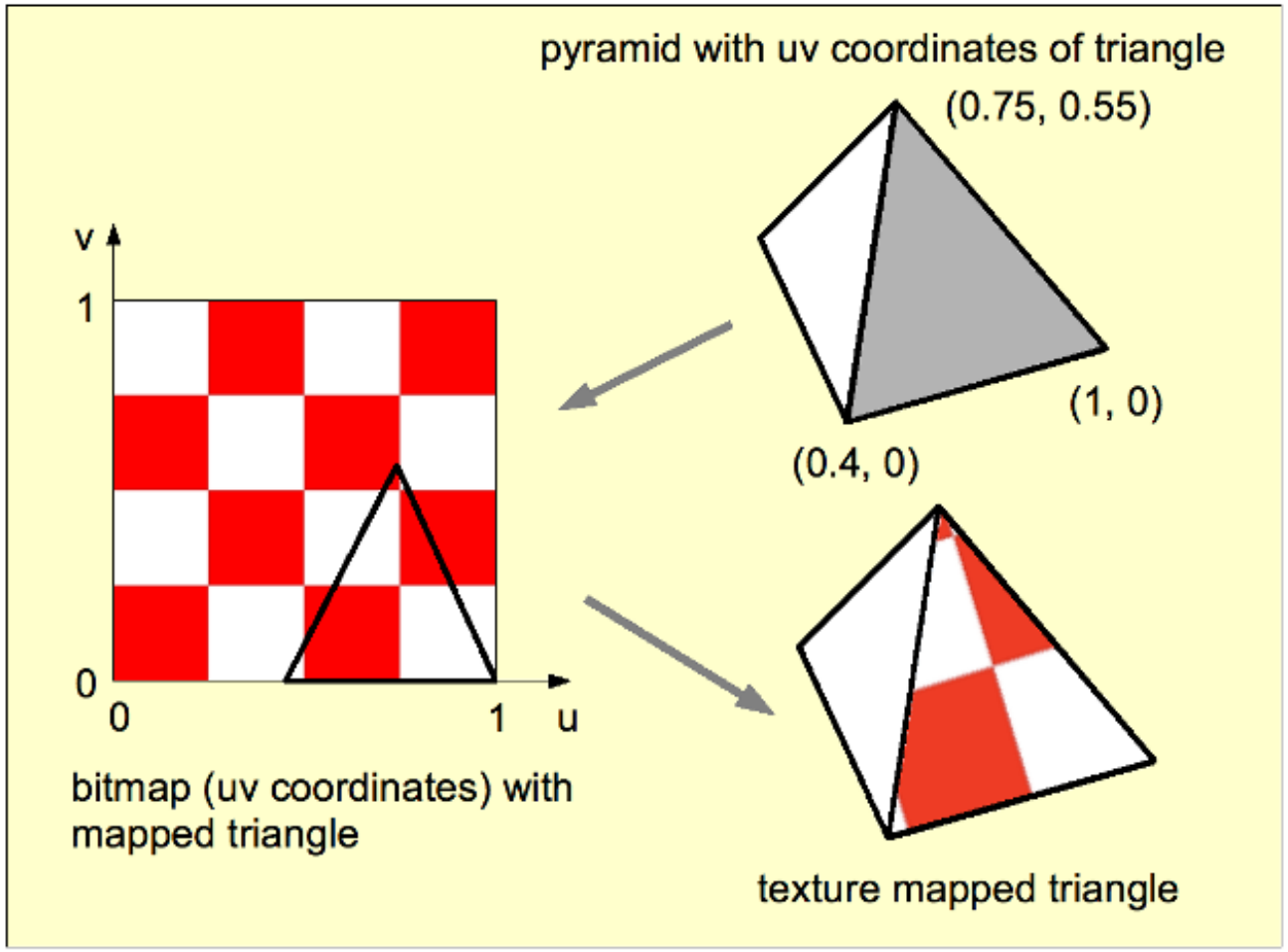
■図4.64――(2次元)マッピングの考え方



「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）

テクスチャ座標系 (uv 座標系)

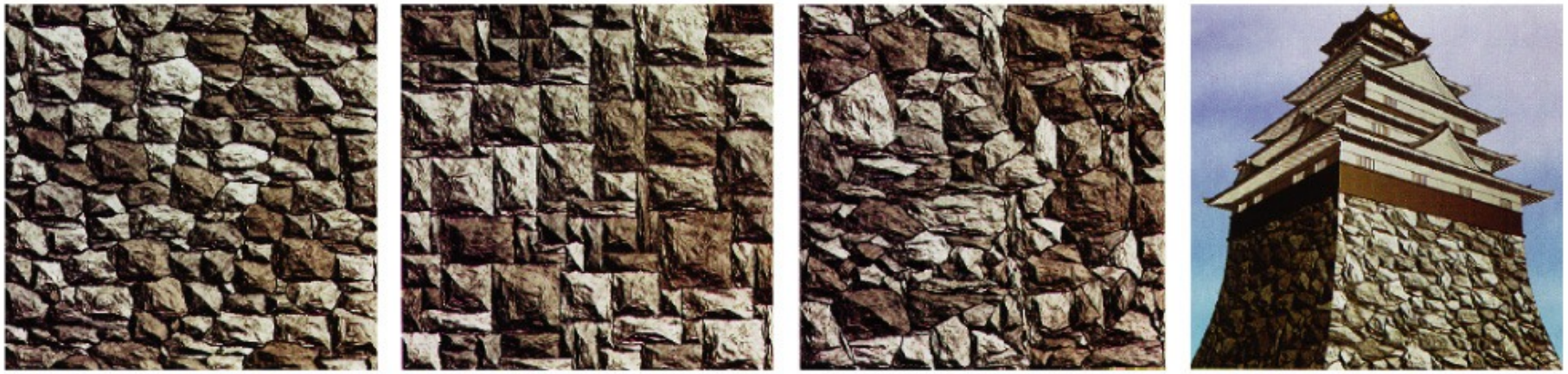
- 画像 (テクスチャ) の空間が $[0, 1]^2$ に正規化される
- ポリゴンはその空間の座標 (uv 座標) を保持



テクスチャマッピングの活用事例

- 表面上の細かな凹凸変化
  - 直接ポリゴンでモデル化するのは手間・効率悪い

■図4.66——手続き的モデルによる石垣のテクスチャ



[a] パラメータを変更して作成した石垣のテクスチャ

[b] 石垣のテクスチャをマッピングして作成された画像

（提供：北陸先端科学技術大学大学院 宮田一乗）

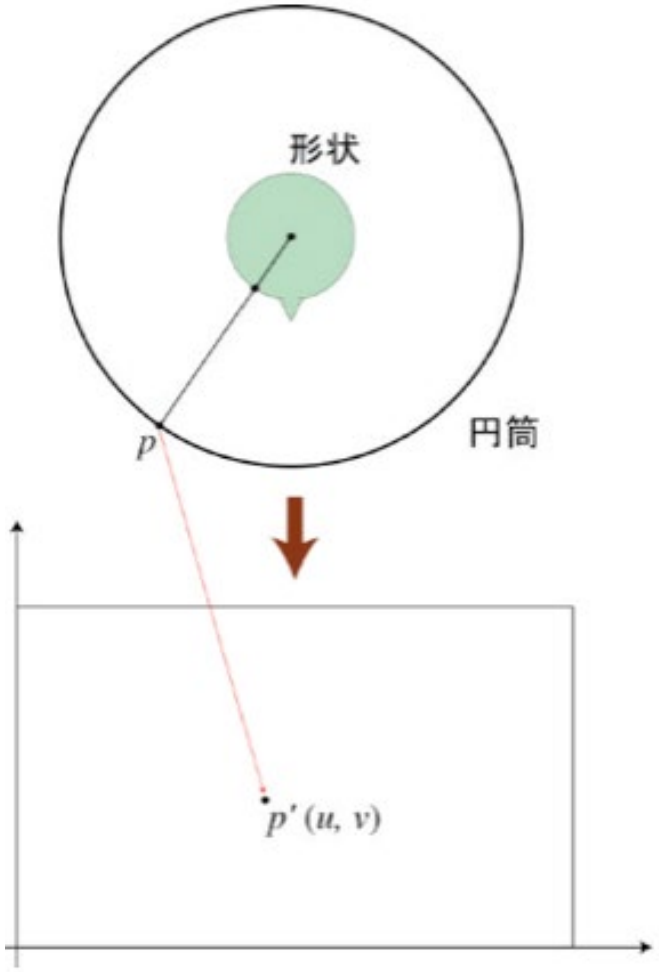
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

テクスチャ座標の計算手法

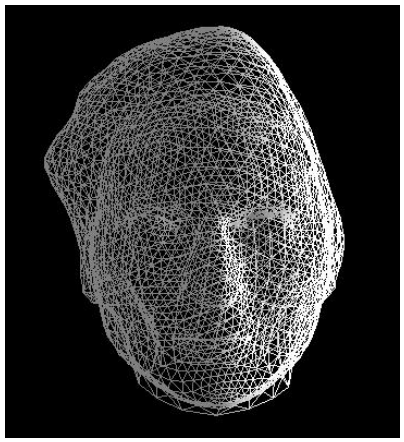
- 3次元 → 2次元のマッピング
  - 円筒マッピング
  - 球面マッピング
  - 平面マッピング
    - 各種パラメタライゼーション、手作業

円筒・球面マッピング

- 形状を覆う円筒(球)を定義
- 円筒の表面上の点が
  テクスチャ座標に相当
- 形状の各頂点から円筒へ
  光線を延ばし、円筒の
  表面との交点をその点に
  おけるテクスチャ座標とする
- 適用できる形状に制限がある



円筒マッピングの例



顔3D形状

+

顔全周テクスチャ画像

マッピング後の画像

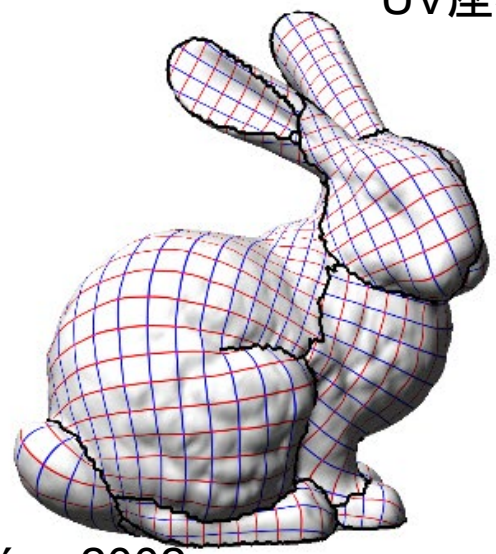
http://www.eurecom.fr/~image/Clonage/geometric2.html

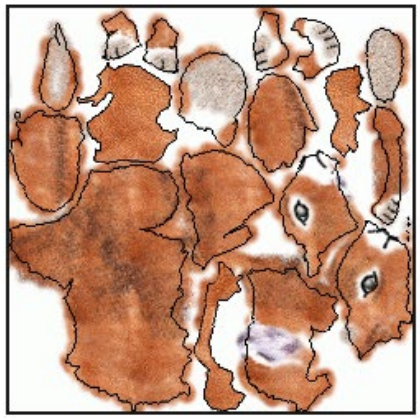
平面マッピング

- テクスチャ座標を計算により定める
  (パラメータ化)
- 可展面でない場合は必ず歪みが生じる
  - 特定の評価関数の値を最小にするように計算

例) $\sum_{j=1}^{m} \|\mathbf{U}_j - \mathcal{U}(\mathbf{M}_j)\|^2 \longrightarrow \min$

UV座標　マッピング関数　頂点座標



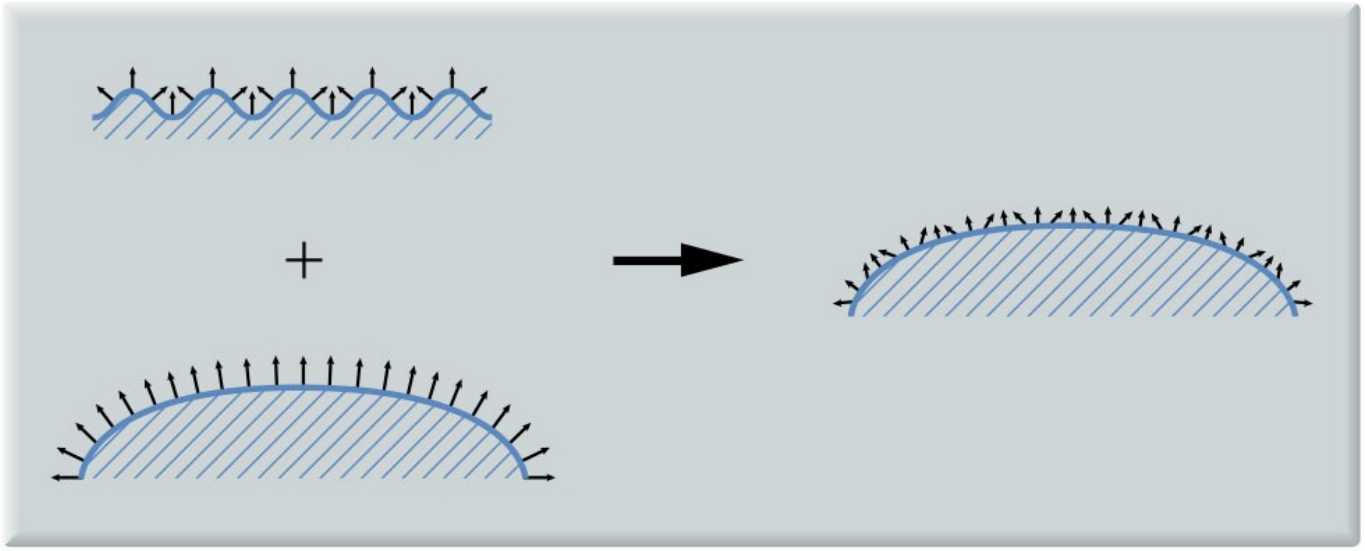


Bruno Lévy 2002

バンプマッピング

- 物体の凹凸を擬似的に表現する手法
- 法線ベクトルを場所に応じて変化させる（形は変えない）

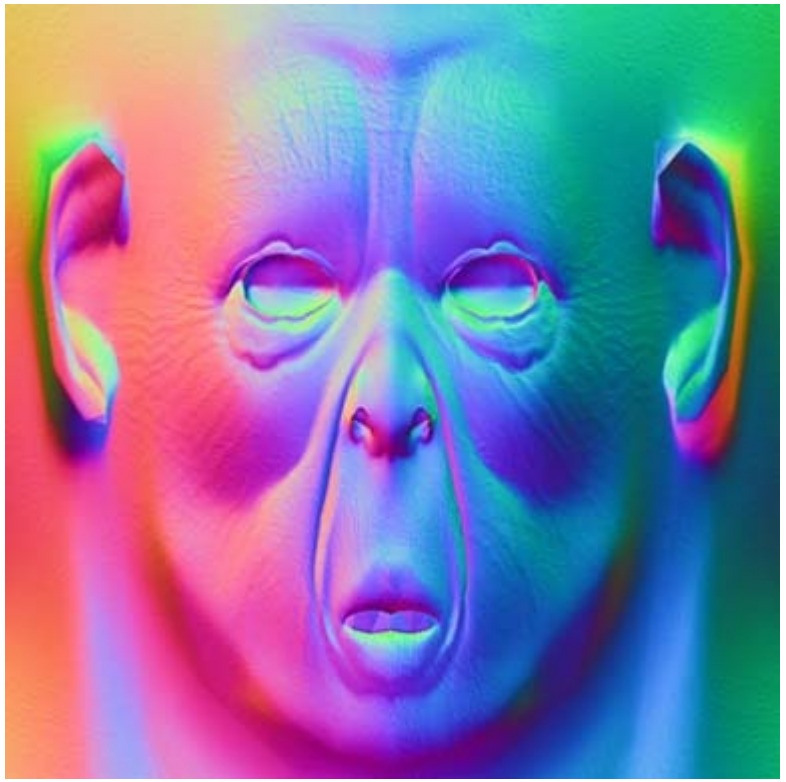
■図4.75——バンプマッピングの考え方



「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）

法線マップ（Normal Maps）

- 画像の各画素ごとに
  法線ベクトルを定義
- テクスチャ画像と
  同じように3D形状に
  マッピング
- 少ない情報で
  複雑な凹凸を表現できる

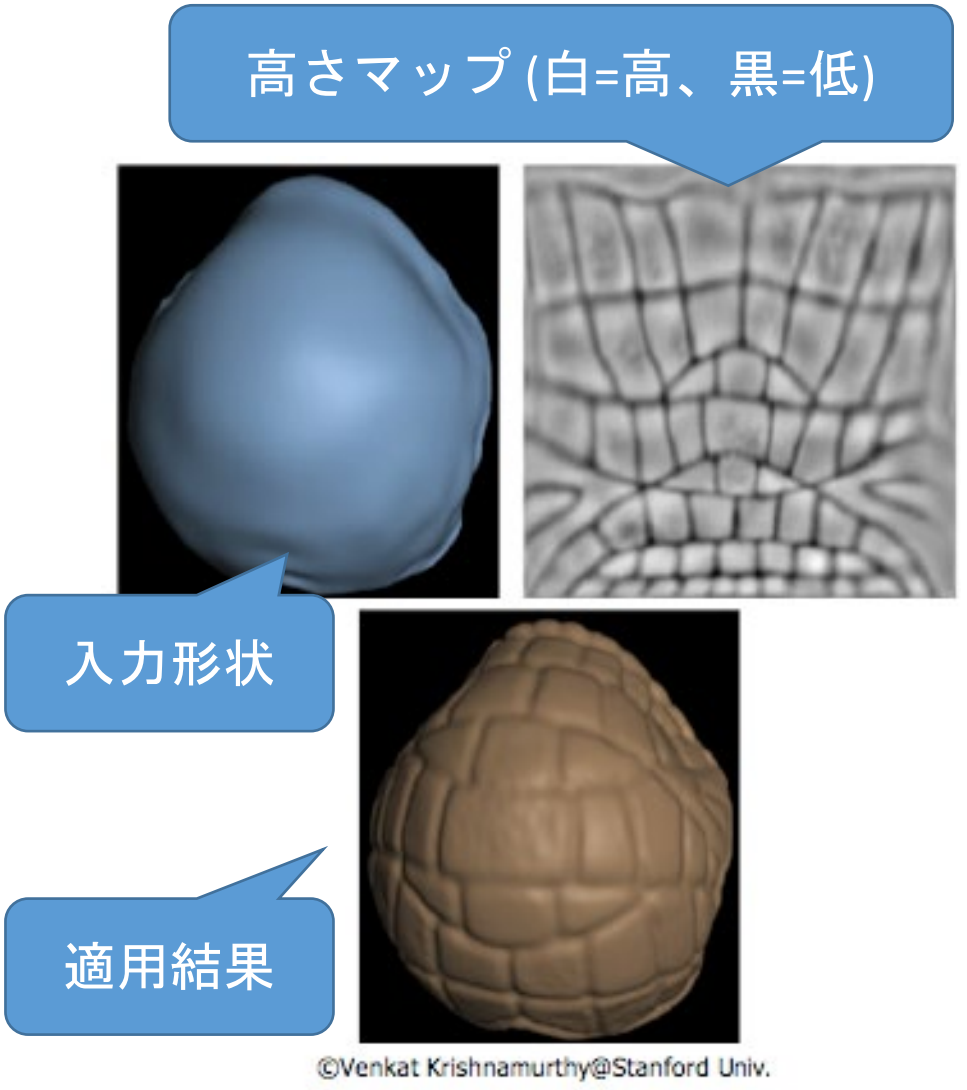


※ RGBの値に、法線ベクトルの x, y, z の
各成分を格納した画像を使用

ディスプレイスメントマッピング

- 高さマップ
  (ハイトフィールド) を
  使ったマッピング

- バンプマッピング
  とは異なり、
  実際に形を変形
  - 高さマップで
    与えられた高さに
    なるように
    頂点の位置をずらす



バンプマッピングと ディスプレイスメントマッピング

- 輪郭の形状に注目
  - バンプマッピング … 輪郭の形状はなめらかなまま
  - ディスプレイスメントマッピング … 輪郭もガタガタ

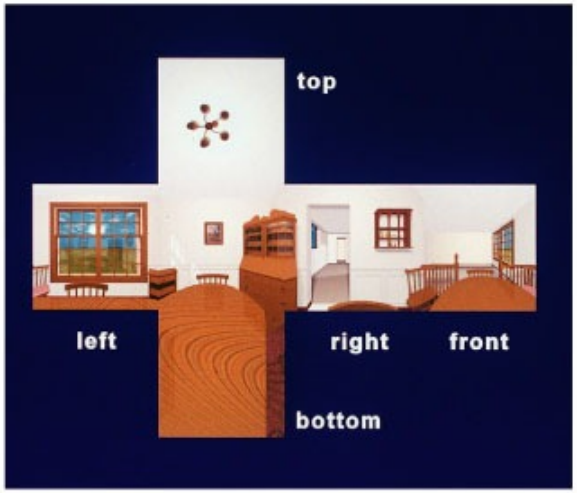
[a] バンプマッピング

[b] ディスプレイスメントマッピング

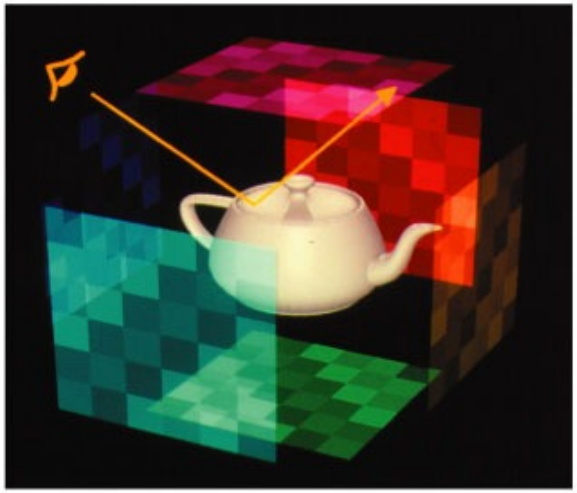
環境マッピング

- 反射を扱うには周囲の情報も必要
  - そのためだけに周囲のモデルを作りこむのは大変
- 反射による周囲の映り込みをマッピングにより擬似的に表現

■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

（Dr. R. Wolfe, Depaul University, Education Slide Set SIGGRAPH 1997 No.20, 21, 60 ©1997 ACM, Inc, Reprinted by permission.）

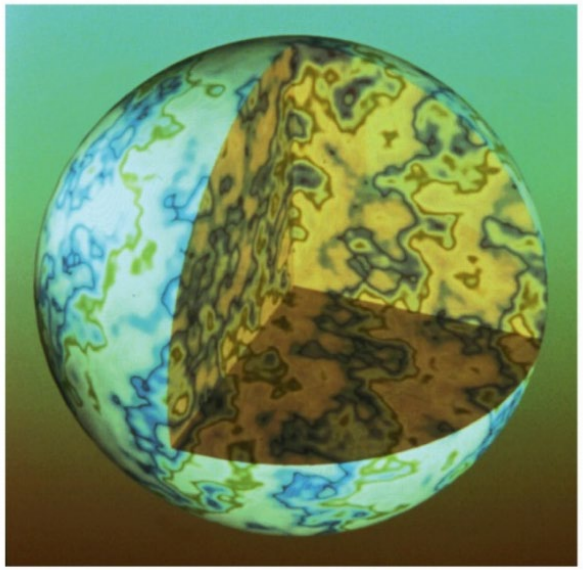
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

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し、様々な形状に切り出して表示
- 任意の切断した面でもリアルな画像を表現
- ３Ｄテクスチャ→３次元空間で定義される関数

■図4.80——各面ごとにテクスチャマッピングを施して表示した例

つなぎ目の模様が不連続



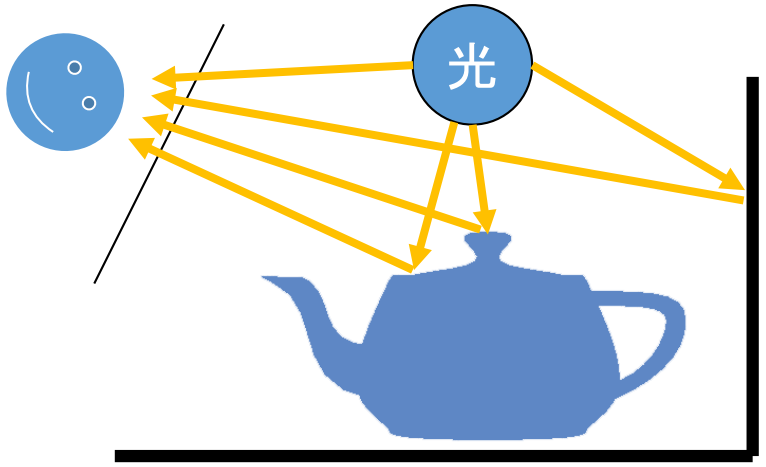
■図4.82——ソリッドテクスチャリングの考え方

Dr. R.Wolfe, Education Slide Set SIGGRAPH 1997 No.37 ©1997 ACM, Inc.Reprinted by permission.)
「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）

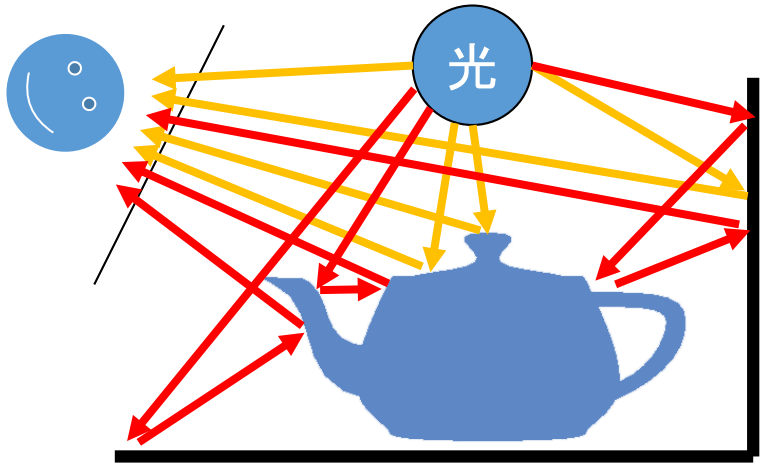
大域照明

大域照明 (Global Illumination)

- 局所照明 (local illumination)
  - これまでの授業では光源から出た光が直接 or 1 回反射して目に届くと仮定
  - 反射・屈折した光が他の物体を照らすことは考えていない

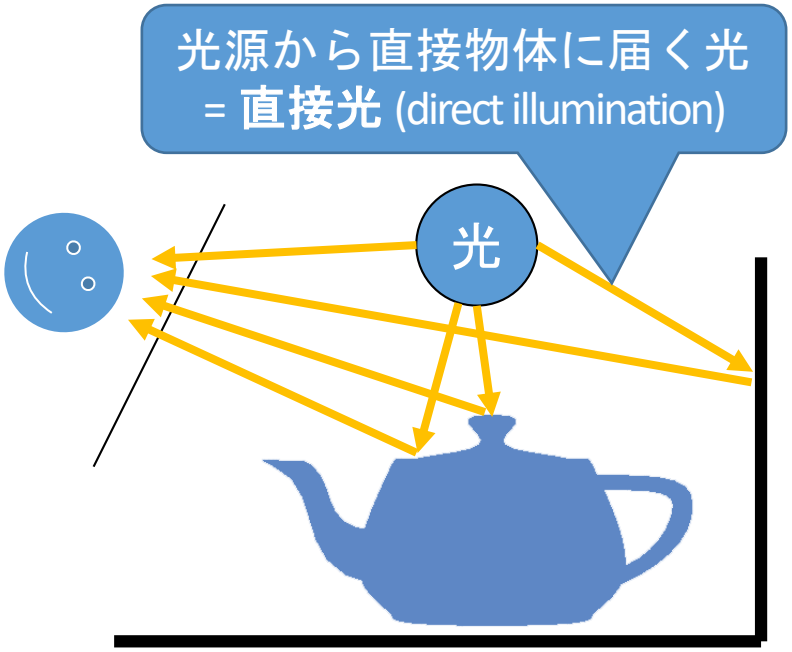


- 大域照明 (global illumination)
  - 実際には、反射・屈折した光が他の物体を照らす (相互反射)
  - 写実的な CG 生成には必須

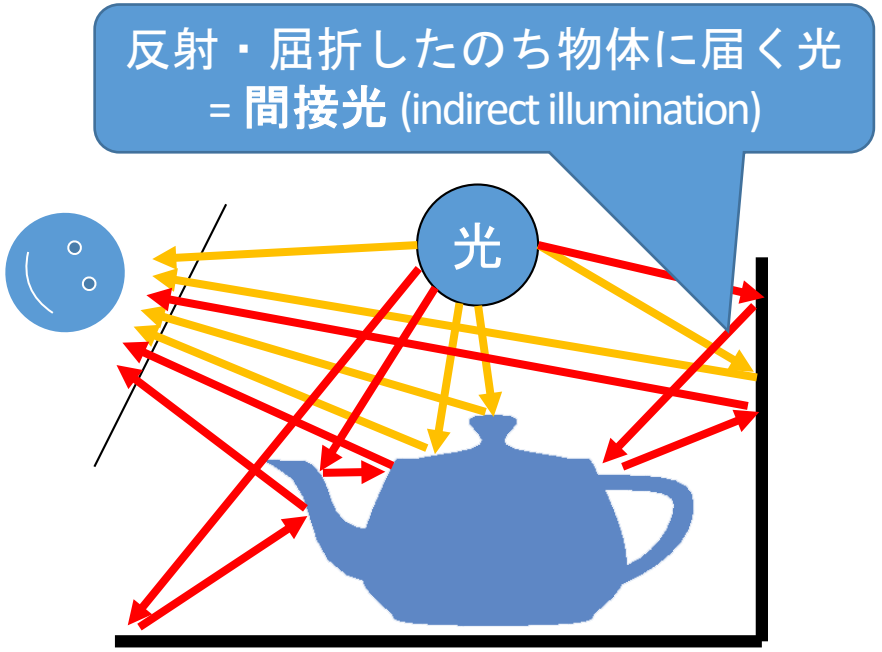


大域照明 (Global Illumination)

- 光路長 (光が目に届くまでの線分の数) による区別
  - 1 or 2 … 局所照明
  - 1 ～ ∞ … 大域照明 (局所照明も包含)

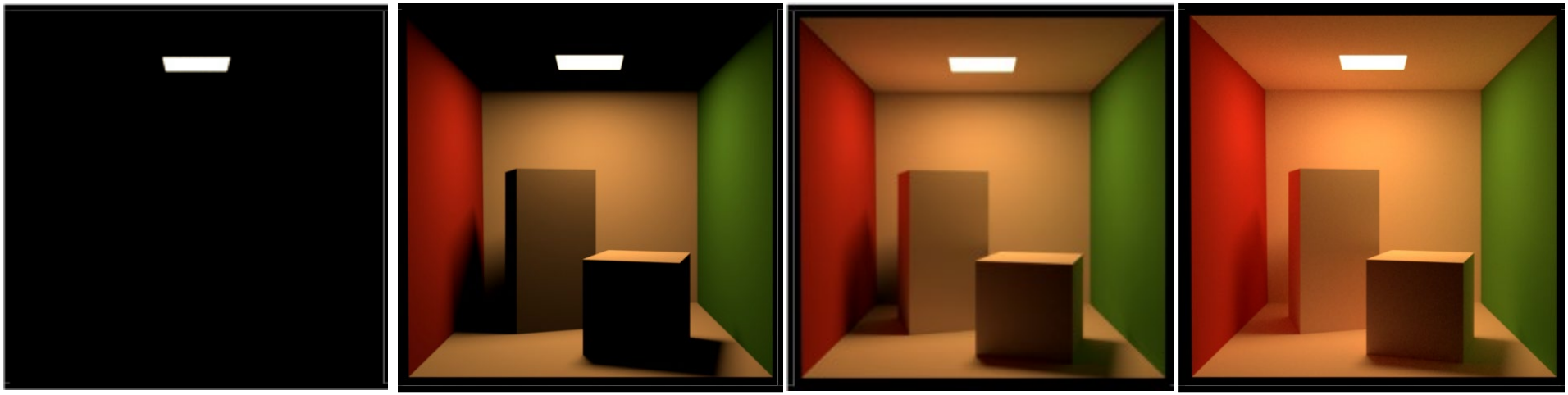


局所照明



大域照明

光路長の違いによる比較



光路長最大 1

光路長最大 2

光路長最大 3

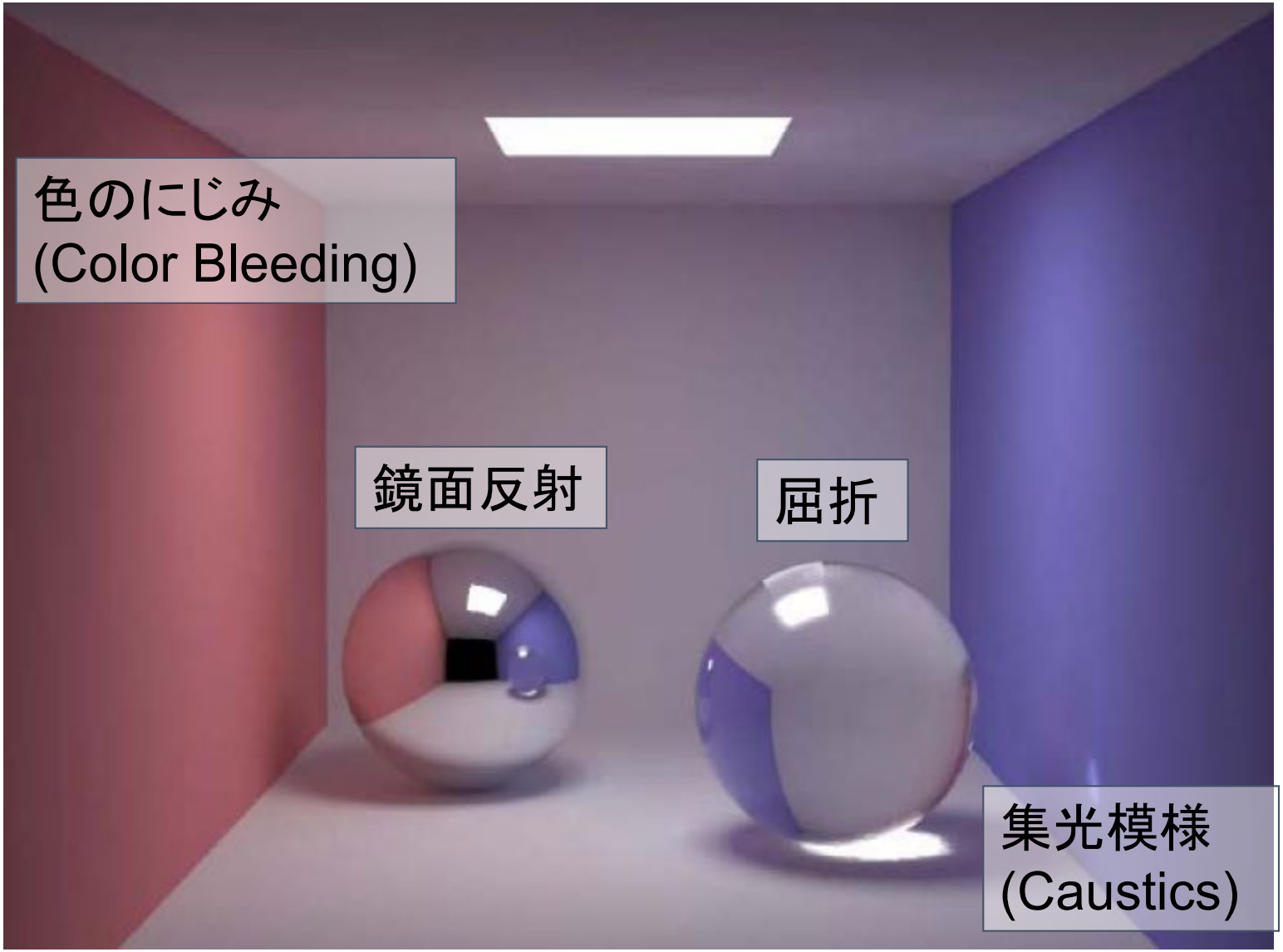
光路長最大 ∞

(物体を照らしていない)

局所照明

大域照明

大域照明で得られる光学効果



大域照明のアルゴリズム

- 目に届くまでの光路(パス)により様々なアルゴリズム
  - 光がどういう順番でどういう材質の表面で反射・屈折するか

- 古典的レイトレーシング (Classical ray tracing)
- ラジオシティ (Radiosity; 相互反射法)
- パストレーシング (Path tracing)
- 双方向パストレーシング (Bi-directional Path Tracing)
- フォトンマッピング (Photon mapping)
- メトロポリス光伝達 (Metropolis light transport, MLT)
- 漸進的フォトンマッピング (Progressive photon mapping)

鏡面反射、屈折

主に拡散面のみ

任意の種類の光路、レイトレーシングに基づく

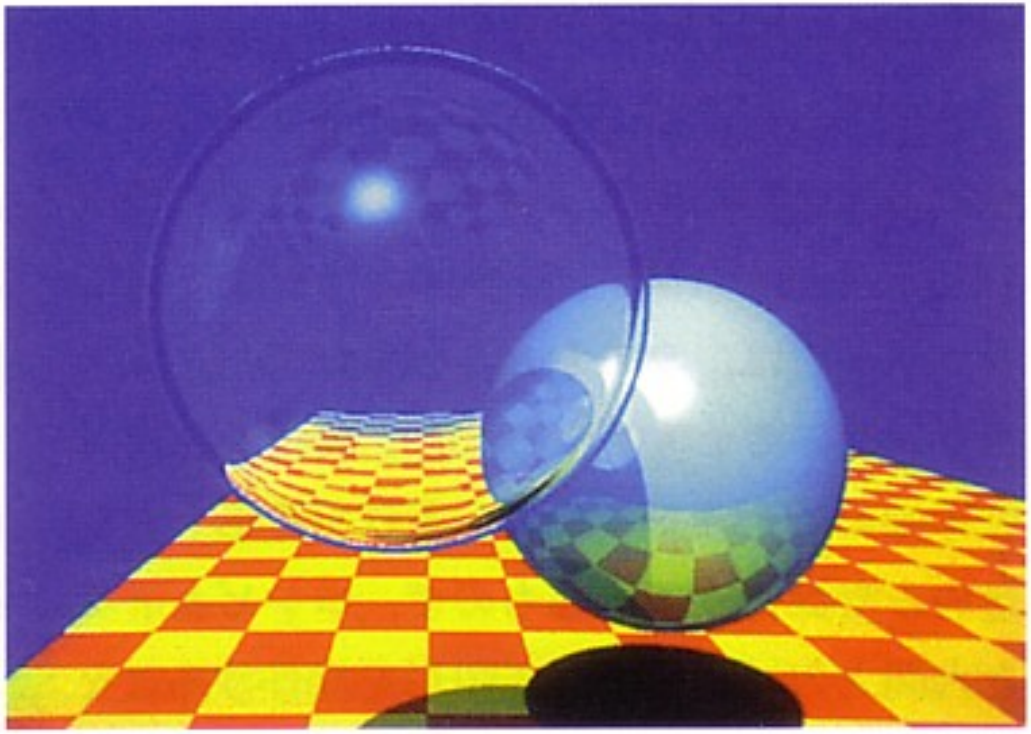
大域照明のアルゴリズム

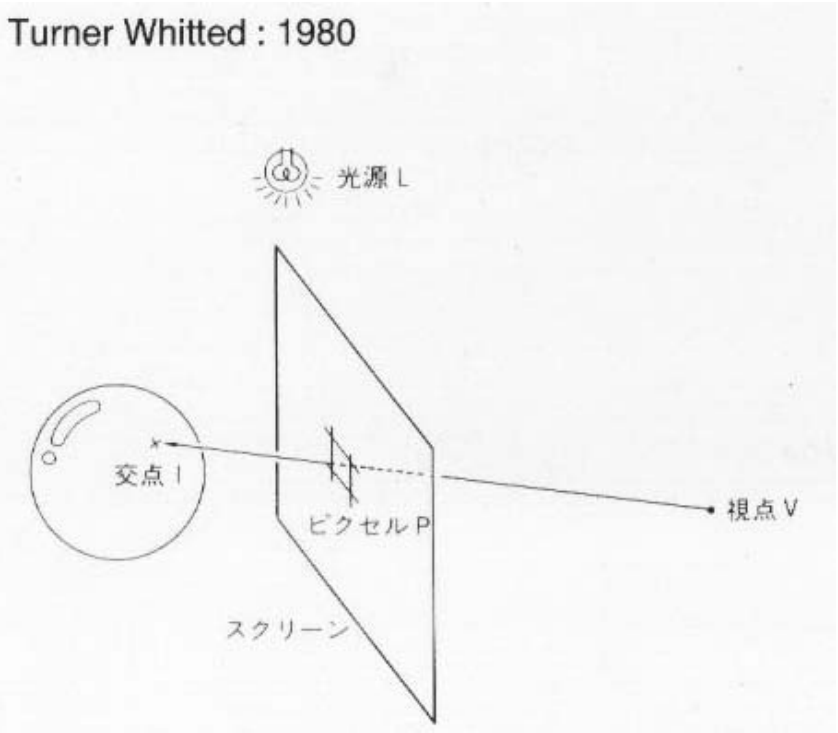
- 目に届くまでの光路(パス)により様々なアルゴリズム
  - 光がどういう順番でどういう材質の表面で反射・屈折するか

- **古典的レイトレーシング (Classical ray tracing)**
- ラジオシティ (Radiosity; 相互反射法)
- パストレーシング (Path tracing)
- 双方向パストレーシング (Bi-directional Path Tracing)
- フォトンマッピング (Photon mapping)
- メトロポリス光伝達 (Metropolis light transport, MLT)
- 漸進的フォトンマッピング (Progressive photon mapping)

古典的レイトレーシング

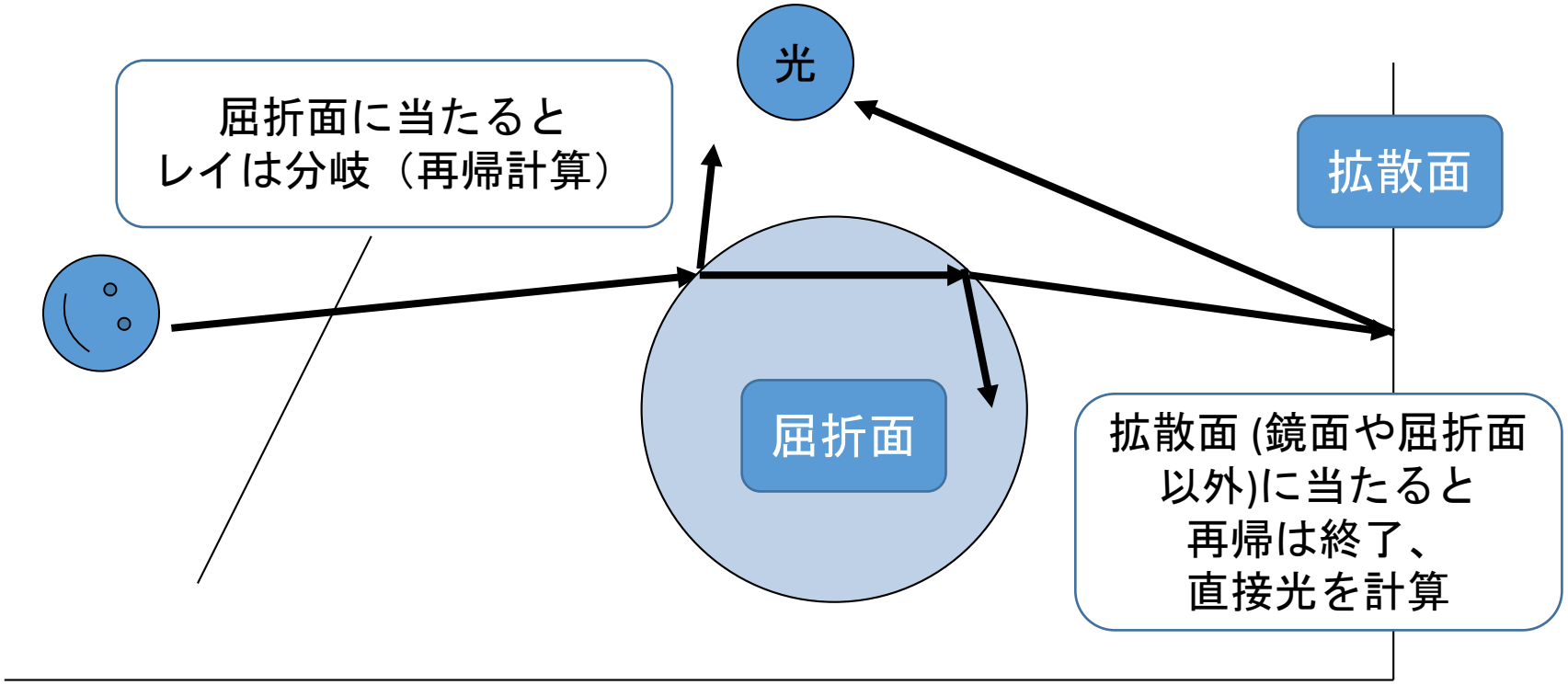
- 1979 Whitted: Ray tracing (光線追跡法)
  - 鏡の映り込みやガラスなどでの屈折を初めて実現





古典的レイトレーシングの仕組み

- 拡散面（鏡面や屈折面以外）では局所照明を採用



レイトレーシングの適用例

■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）

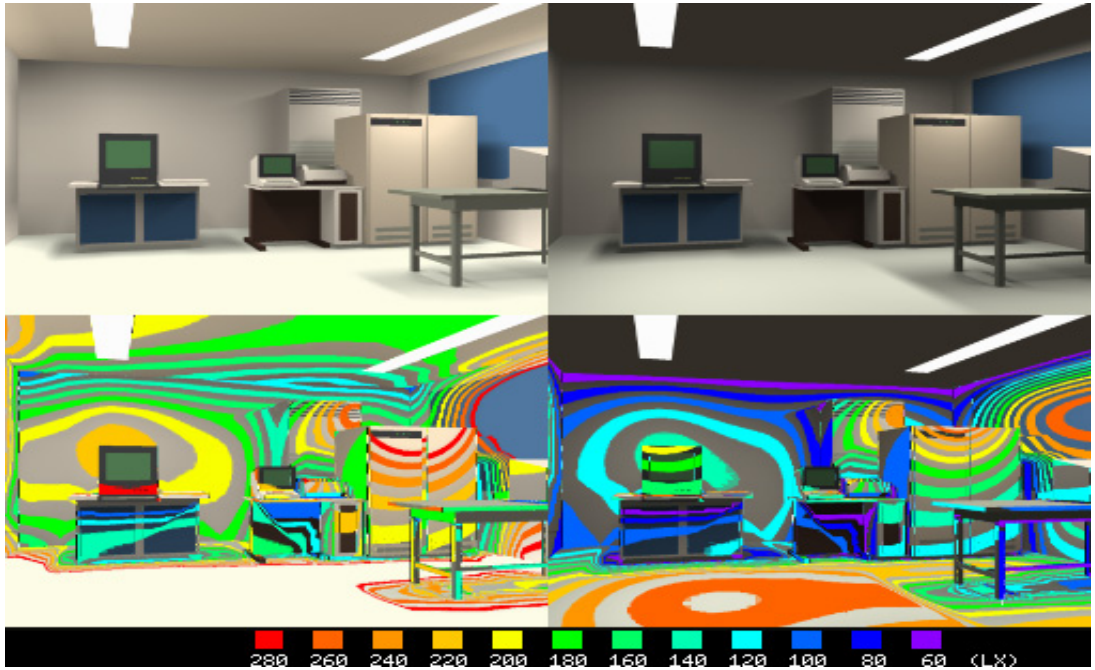
大域照明のアルゴリズム

- 目に届くまでの光路(パス)により様々なアルゴリズム
  - 光がどういう順番でどういう材質の表面で反射・屈折するか

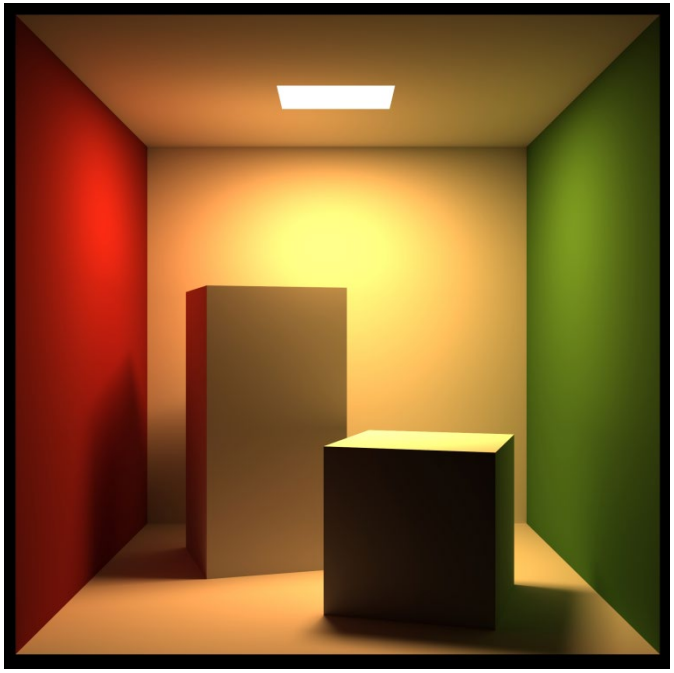
- 古典的レイトレーシング (Classical ray tracing)
- **ラジオシティ (Radiosity; 相互反射法)**
- パストレーシング (Path tracing)
- 双方向パストレーシング (Bi-directional Path Tracing)
- フォトンマッピング (Photon mapping)
- メトロポリス光伝達 (Metropolis light transport, MLT)
- 漸進的フォトンマッピング (Progressive photon mapping)

ラジオシティ (Radiosity; 相互反射法)

- 元々は拡散面のみからなるシーンを対象
  - 日米で同時に発表 … 西田ら (1985, 照明光学に基づく)、Cohen ら (1985, 熱工学に基づく)



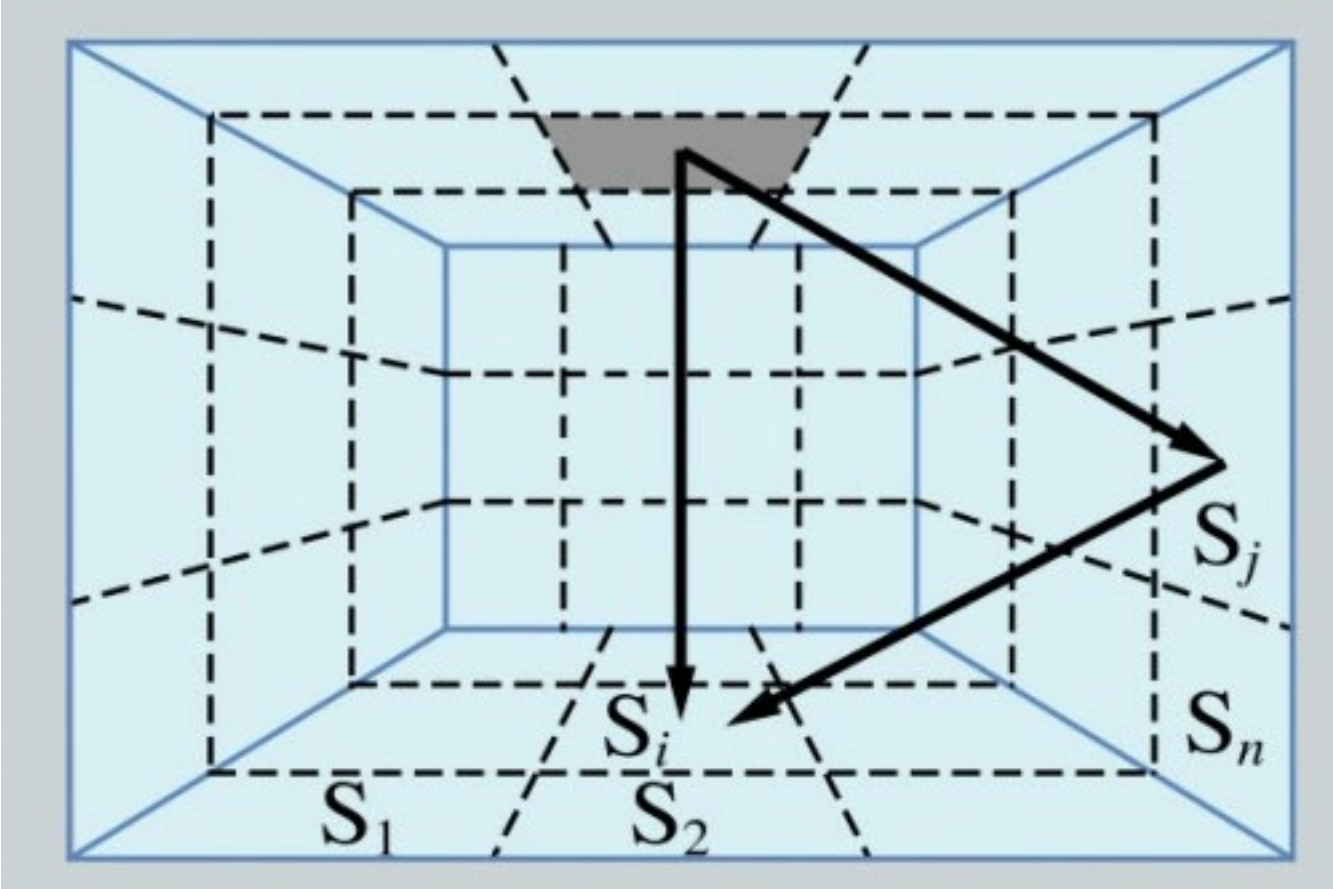
1985 相互反射法



1985 ラジオシティ法(“Cornel Box”)

ラジオシティ法 (Radiosity)

- シーンを細かい要素 (パッチ) に分割し、パッチ同士の光のやり取りを計算
  - 有限要素法 (Finite Element Method) の一種



ラジオシティ法 (Radiosity)

- 拡散面のみなら、各パッチの明るさ (放射照度 E) が線形方程式を解くことで求まる

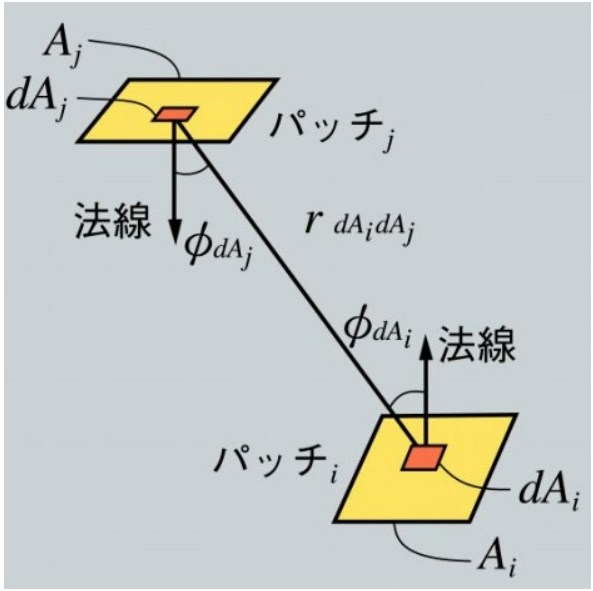
拡散反射率

パッチ j のパッチ i に対する寄与率
(form factor; 形状因子)

$$E_i = E_{0i} + \rho_i \sum_j F_{ji} E_j$$

$$\sum_i F_{ij} = 1 \qquad F_{ij} A_i = F_{ji} A_j$$

$$\begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix} = \begin{pmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & & -\rho_2 F_{2n} \\ \vdots & & \ddots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1-\rho_n F_{nn} \end{pmatrix}^{-1} \begin{pmatrix} E_{01} \\ E_{02} \\ \vdots \\ E_{0n} \end{pmatrix}$$



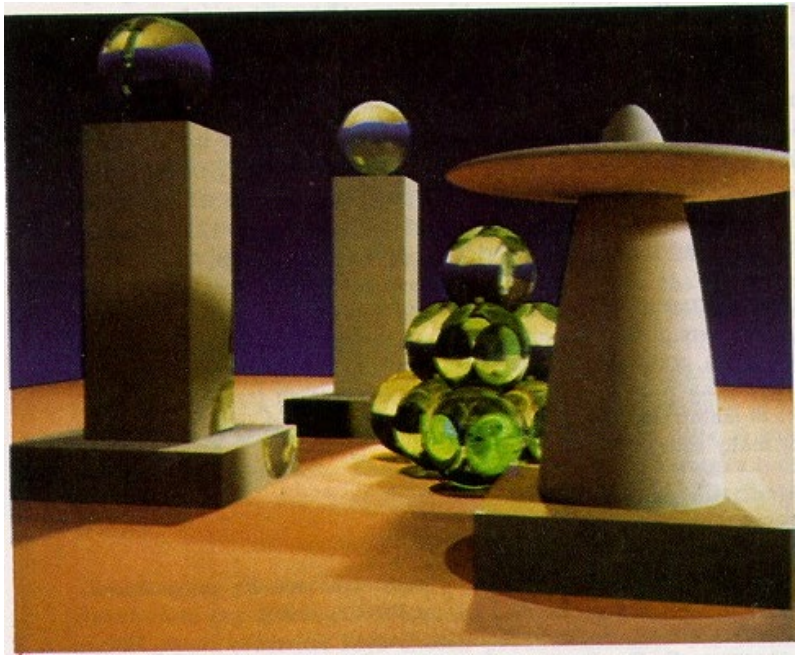
大域照明のアルゴリズム

- 目に届くまでの光路(パス)により様々なアルゴリズム
  - 光がどういう順番でどういう材質の表面で反射・屈折するか

- 古典的レイトレーシング (Classical ray tracing)
- ラジオシティ (Radiosity; 相互反射法)
- **パストレーシング (Path tracing)**
- 双方向パストレーシング (Bi-directional Path Tracing)
- フォトンマッピング (Photon mapping)
- メトロポリス光伝達 (Metropolis light transport, MLT)
- 漸進的フォトンマッピング (Progressive photon mapping)

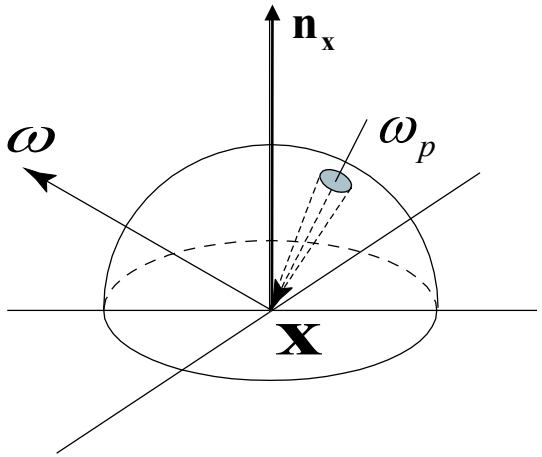
パストレーシング (経路追跡法)

- 任意の種類の光路を扱える
  最初の大域照明アルゴリズム
- 照明計算の理論式である
  「レンダリング方程式」
  とともにその解法として
  提案された
- モンテカルロ積分に基づく
  - これ以降のアルゴリズムは
    いずれもモンテカルロ積分



レンダリング方程式 [Kajiya 1986]

- 真空中での任意の光学現象の理論方程式
- 再帰的な積分方程式
  - 左辺の L_{out} が右辺の L_{in} として再登場



物体が自己発光する場合の放射輝度

反射率 (BRDF)

$$L_{out}(\mathbf{x},\boldsymbol{\omega}) = L_e(\mathbf{x},\boldsymbol{\omega}) + \int_{\Omega_\mathbf{x}} f_r(\mathbf{x},\boldsymbol{\omega}_\mathbf{p},\boldsymbol{\omega}) L_{in}(\mathbf{x},\boldsymbol{\omega}_\mathbf{p}) \max(0,\mathbf{n}_\mathbf{x}\cdot\boldsymbol{\omega}_\mathbf{p}) d\omega_p$$

出射光の放射輝度

自己発光しない場合は省略

ある点 x での単位半球面

入射光の放射輝度

微小立体角

モンテカルロ積分

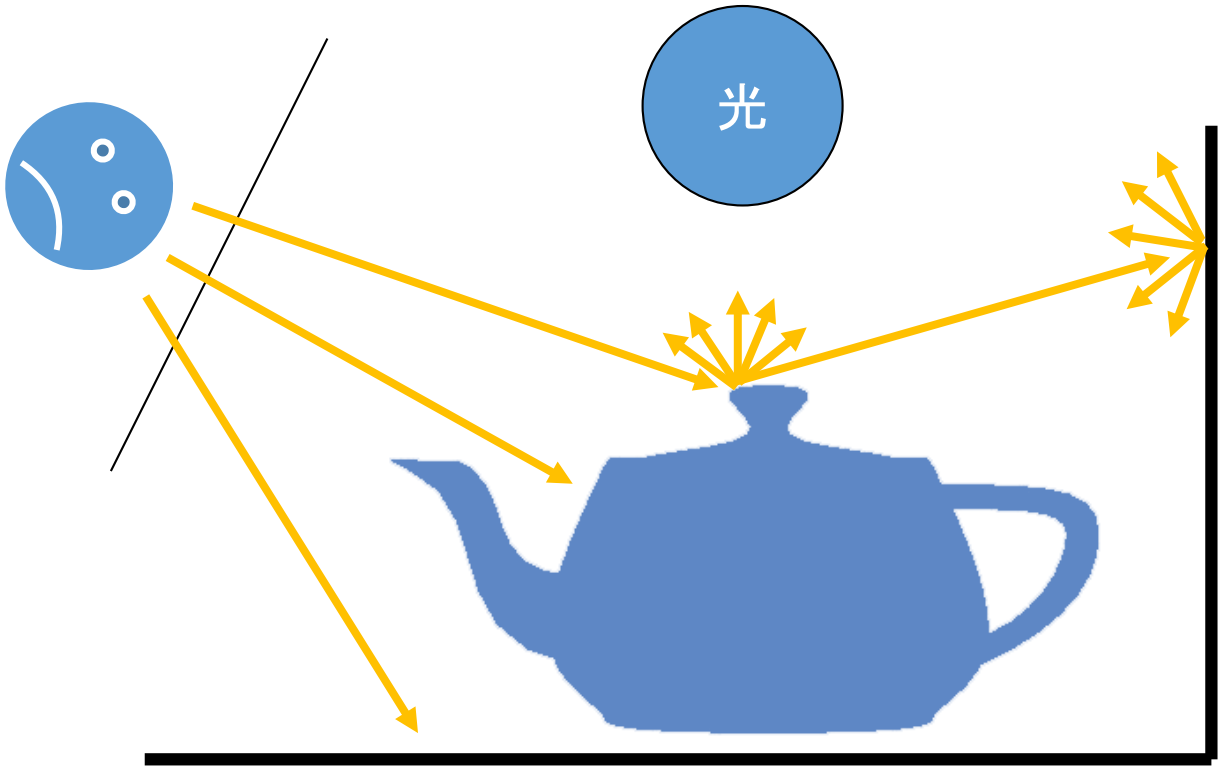
- 乱数を使った数値積分、方程式が複雑でも適用可能
  - ○高次元の積分でも次元数の影響なく収束
  - ×ノイズが現れる → サンプル数を増やす or 賢くサンプリング
- 簡単な積分の例　$I = \int_{\Omega} f(x)\ dx$
- モンテカルロ積分　$\hat{I} = \frac{1}{N}\sum_{i}^{N} f(X_i),\ \lim_{N\to\infty} \hat{I} = I$

Ω の中からランダムに選ばれたサンプル

  - ノイズ (標準偏差、誤差) はサンプル数 N に対し $O\left(\frac{1}{\sqrt{N}}\right)$ で減少

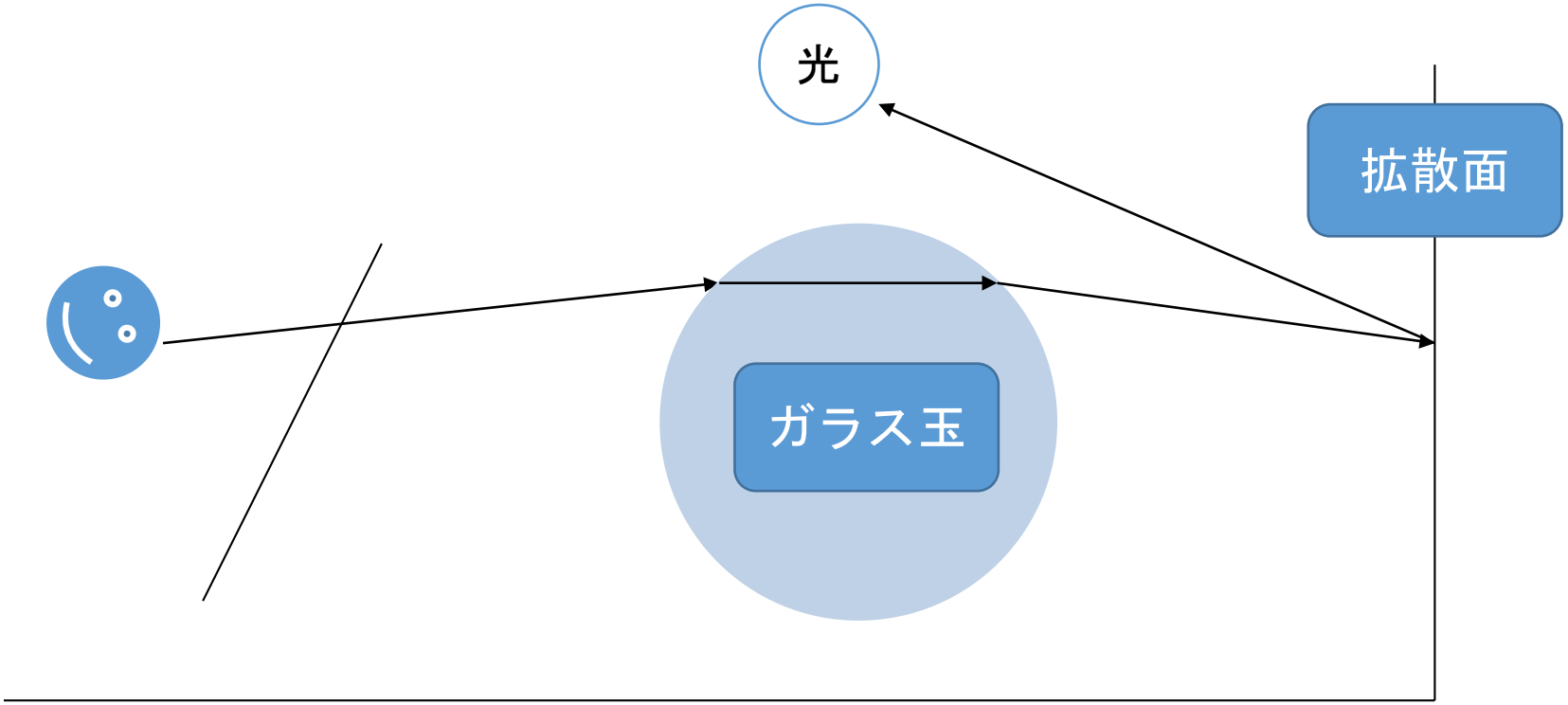
単純な発想でのモンテカルロ積分

- レイが物体表面に当たるたび N 個サンプリング
  → すぐ指数爆発、計算機では実現困難



パストレーシング

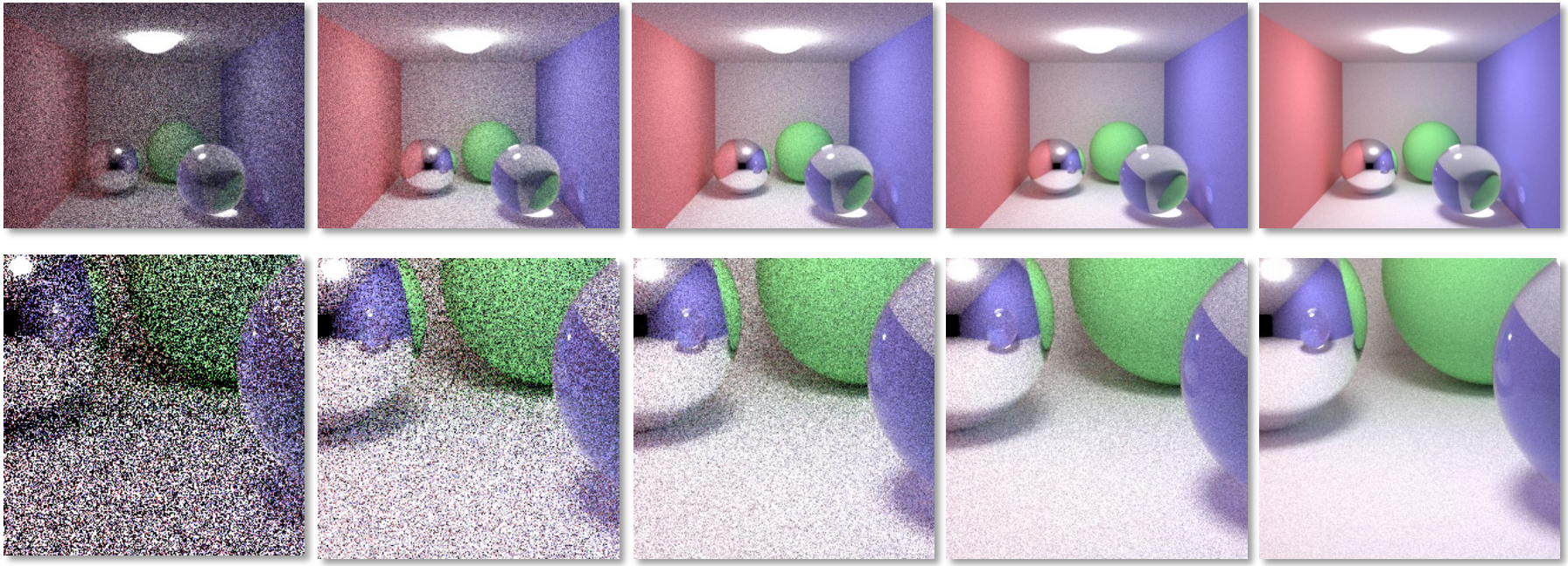
- レイを分岐させずに 1 本ずつ追跡
- 画素ごとに複数 (例: 1,000 本) レイを飛ばし、期待値を計算



収束の様子

出典:「物理ベースレンダラ edupt 解説」by hole

- Intel Core i7 980 (3.33GHz) 10スレッドで計算



16 sample/pixel
13秒

64 sample/pixel
44秒

256 sample/pixel
174秒

1024 sample/pixel
690秒

4096 sample/pixel
2764秒

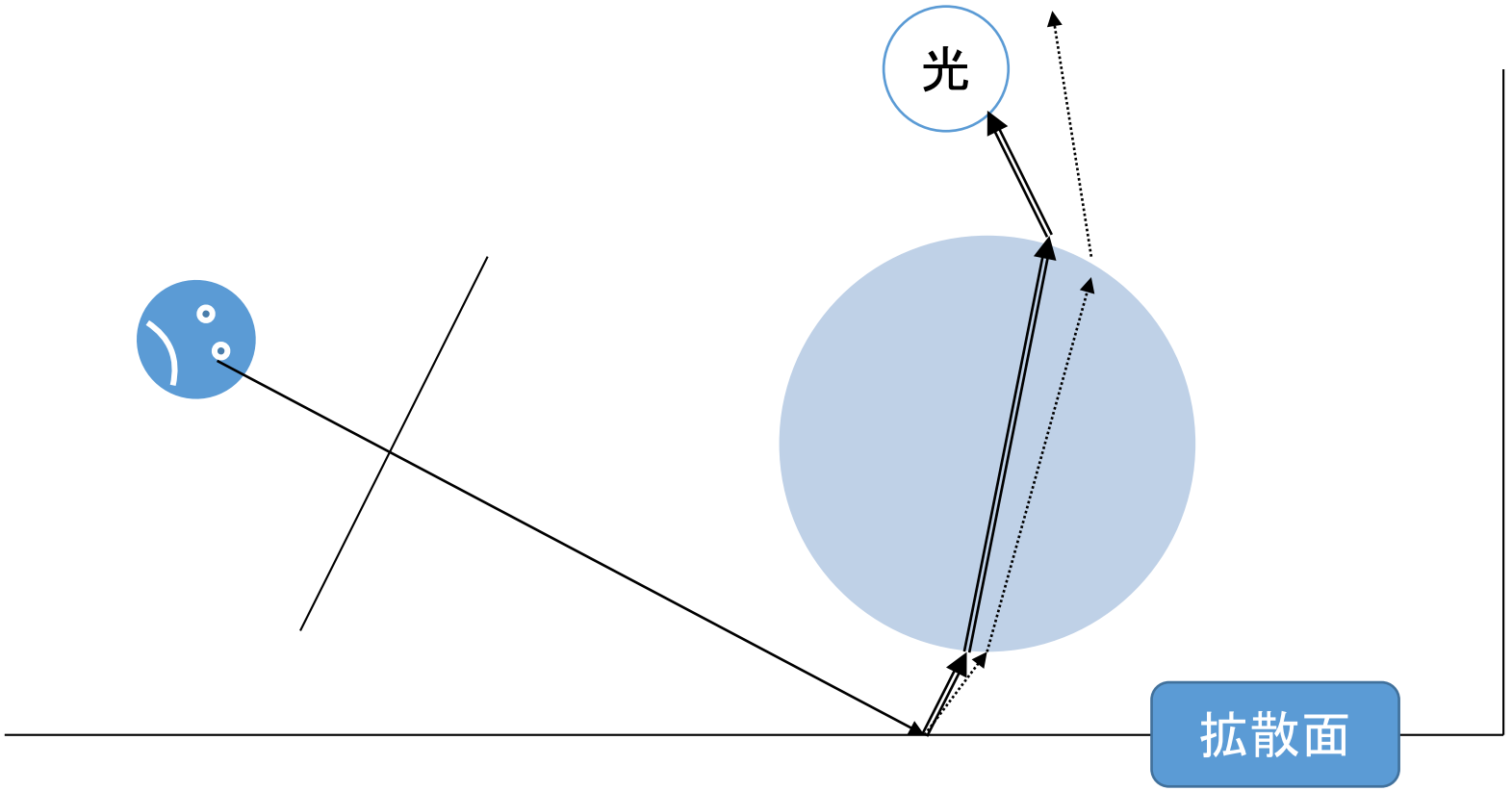
パストレーシングの特徴

- 間接光の変化が緩やかなら収束はそれほど遅くない
  - 明るさの変化が極端でない、遮蔽があまりない、など
- 工夫 (Next Event Estimation) すればかなり実用的
  - 有名な商用レンダラ Arnold もパストレーシングベース

- 苦手な種類の光路もある → ノイズが多い、収束が遅い
- パストレーシングを改良したアルゴリズムは
  「1 サンプル (= レイ) ごとのコストを高める代わりに
  賢くサンプリングして収束を速める」戦略
  → 単純なシーンならむしろパストレーシングの方が速い

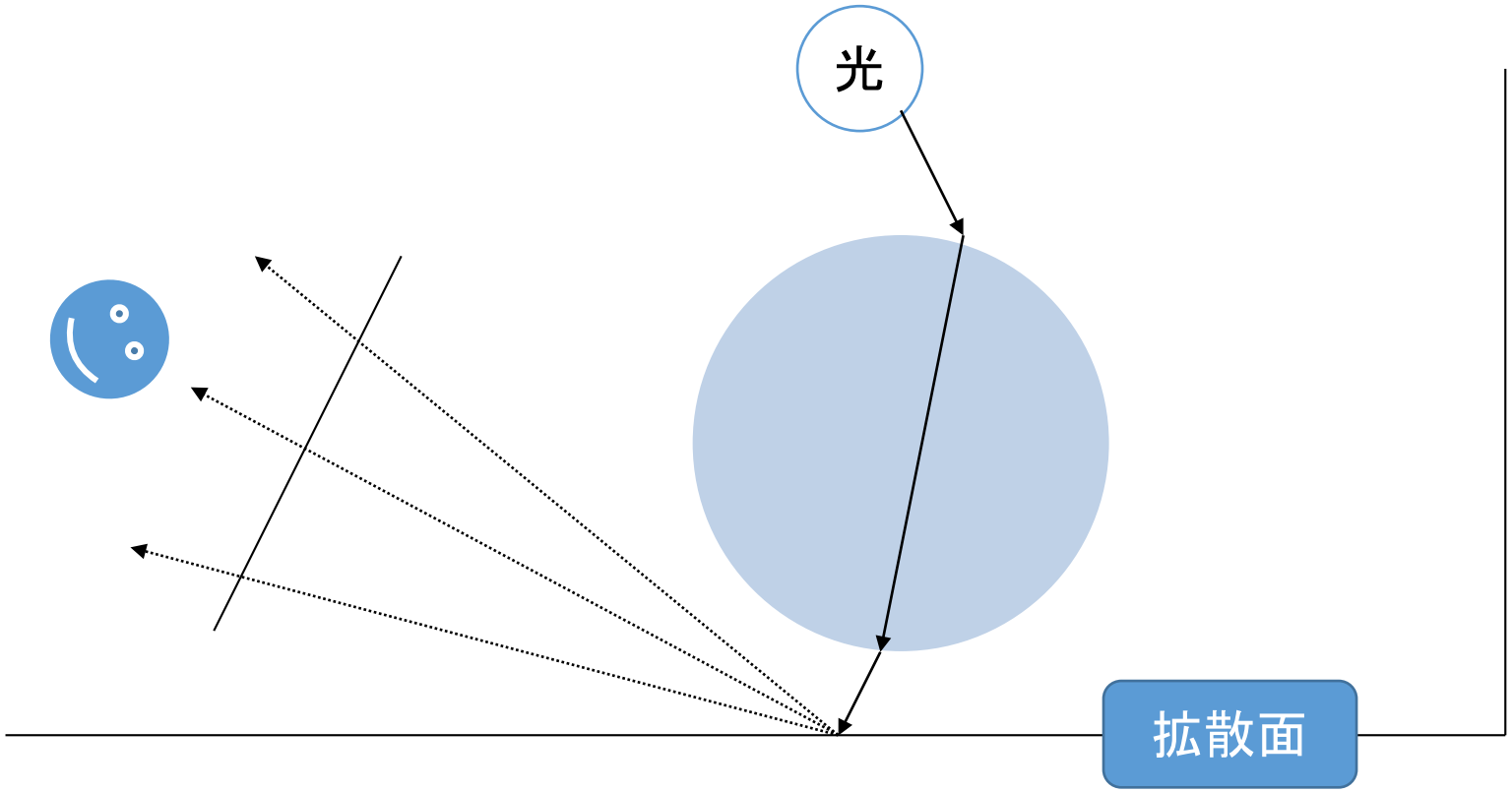
パストレーシングの短所

- 拡散面→反射・屈折→光源の光路は大変
  - ちょうど光源に届くような光路を見つけにくい



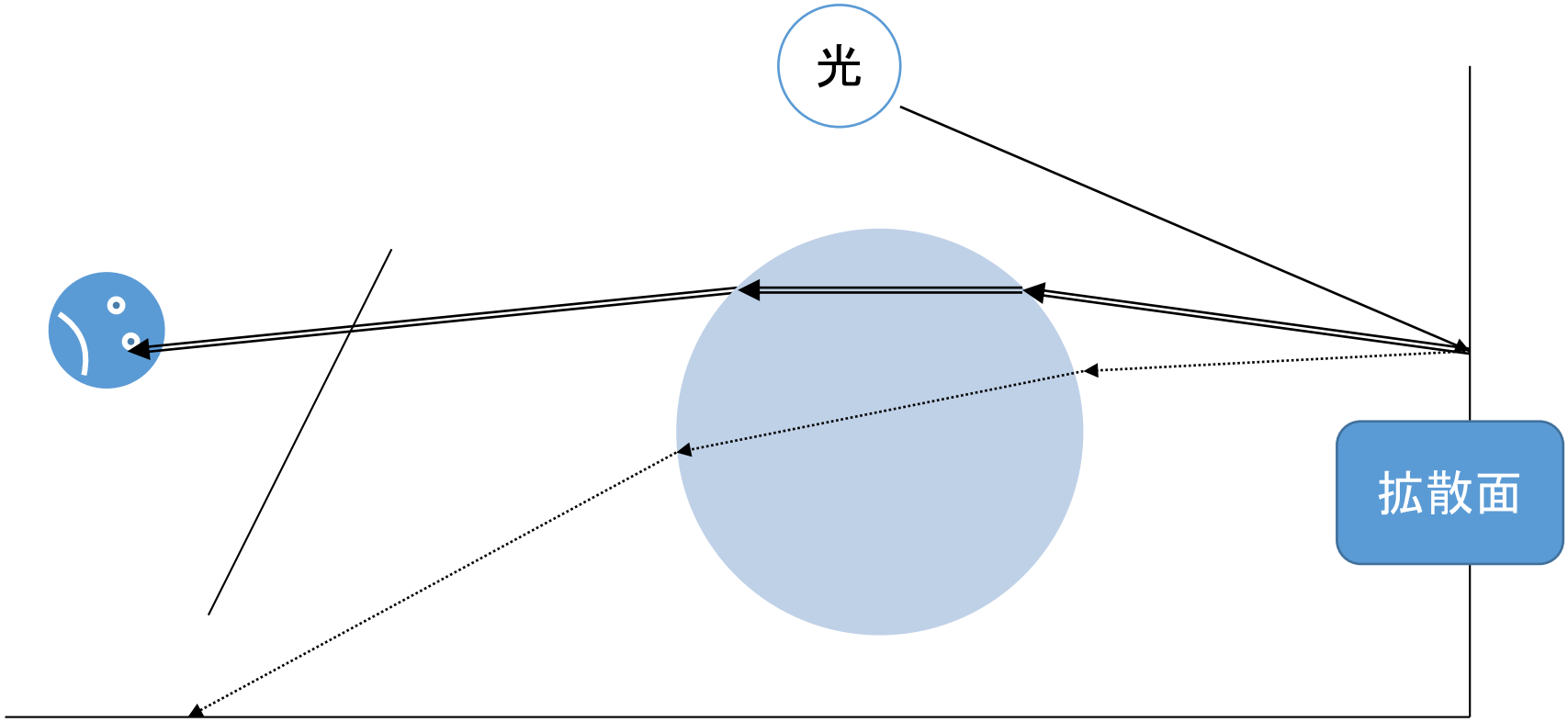
逆経路追跡法 (Particle Tracing)

- 一方、光源→反射・屈折→拡散面なら、光源から光を追跡した方が、目に届く光路を作りやすい



逆経路追跡法 (Particle Tracing)

- しかし光源から追跡しづらい種類の光路もある
  - ちょうど視点に届く光路を見つけづらい



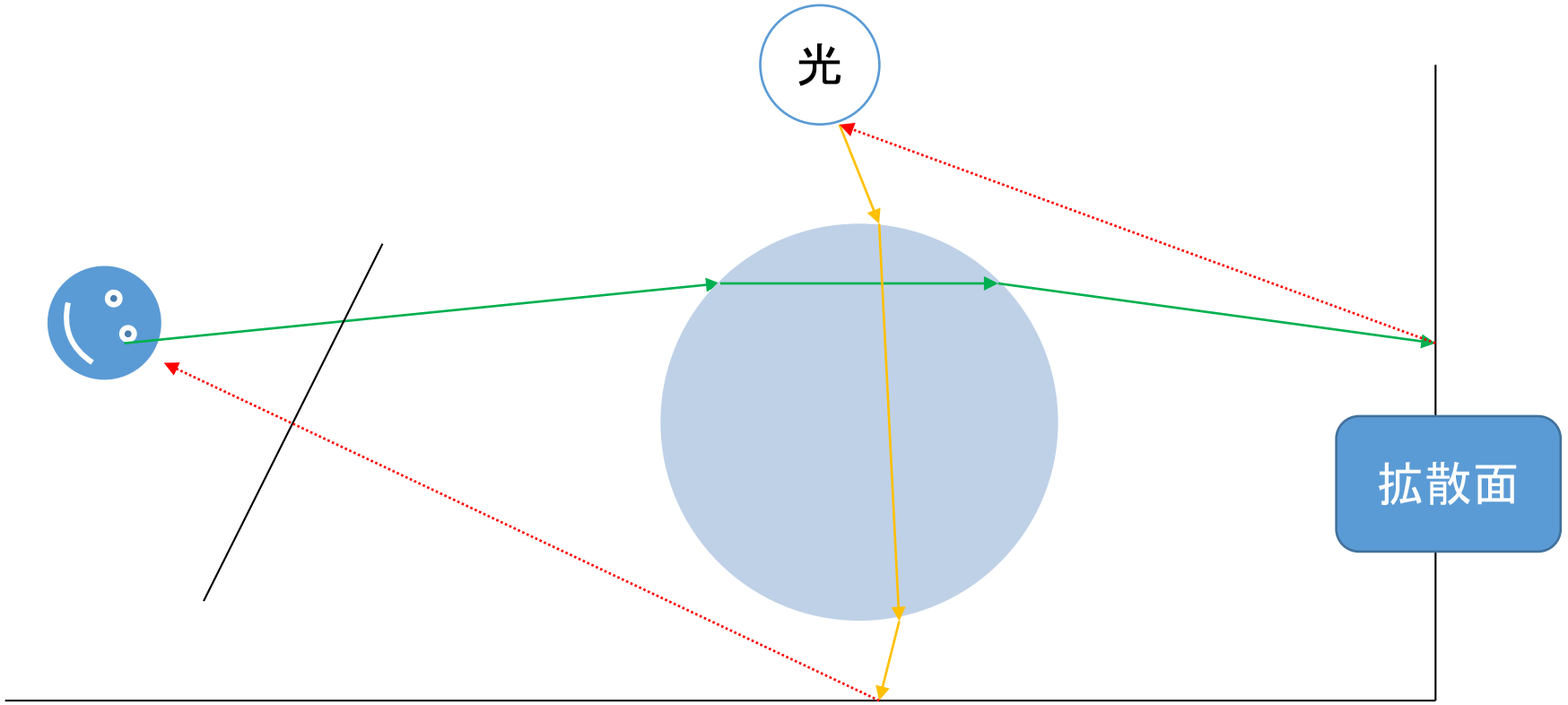
大域照明のアルゴリズム

- 目に届くまでの光路(パス)により様々なアルゴリズム
  - 光がどういう順番でどういう材質の表面で反射・屈折するか

- 古典的レイトレーシング (Classical ray tracing)
- ラジオシティ (Radiosity; 相互反射法)
- パストレーシング (Path tracing)
- **双方向パストレーシング (Bi-directional Path Tracing)**
- フォトンマッピング (Photon mapping)
- メトロポリス光伝達 (Metropolis light transport, MLT)
- 漸進的フォトンマッピング (Progressive photon mapping)

双方向パストレーシング

- 目からの追跡と光源からの追跡のいいとこ取り
  - それぞれ途中まで光路を作り、最後につなぎ合わせる



双方向パストレーシング

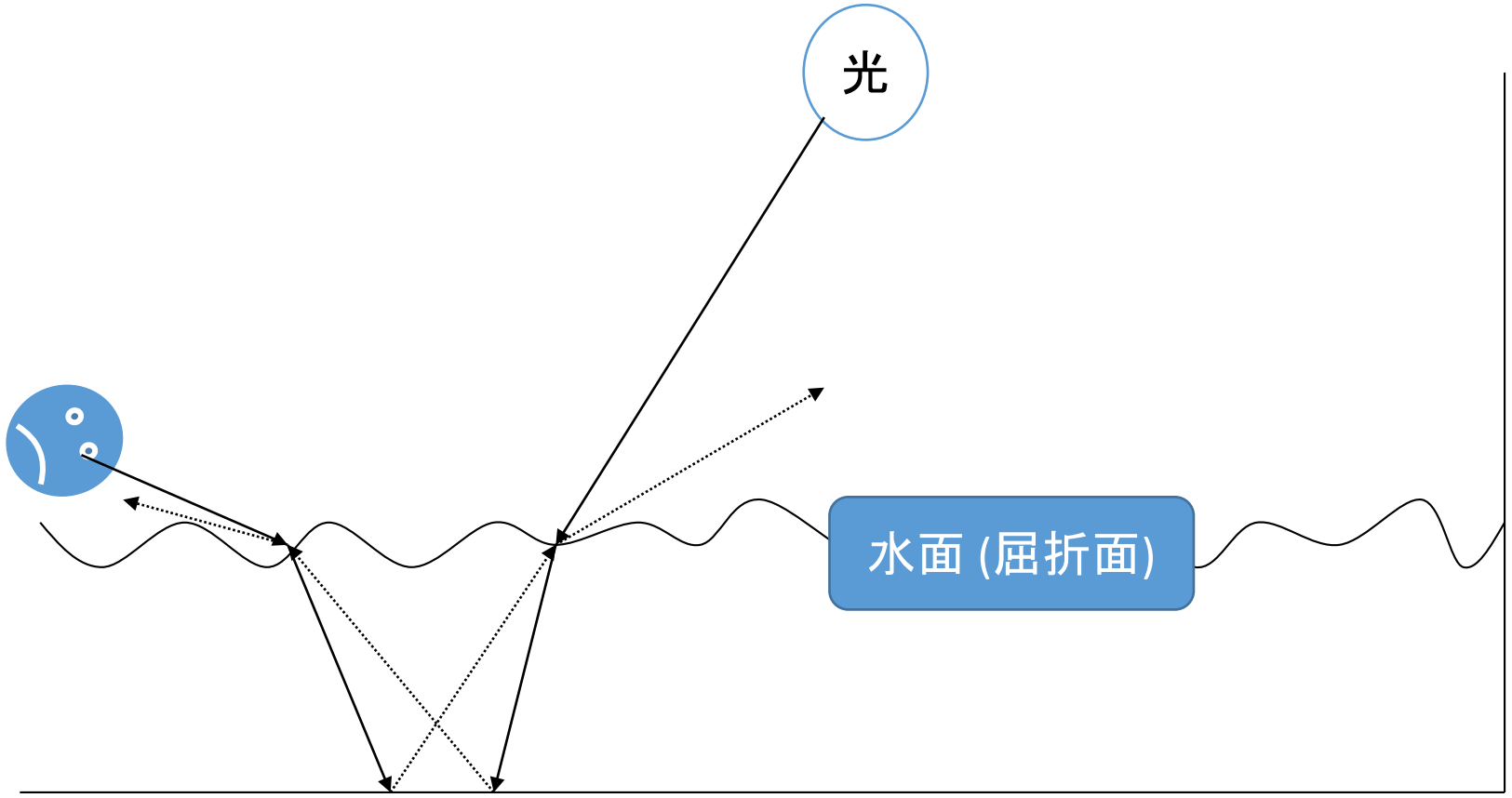
- パストレーシングとの比較 (32 サンプル / 画素)

パストレーシング

双方向パストレーシング

双方向パストレーシングの短所

- 双方向で追跡しても光路をつなげづらい場合がある
  - 水中の集光模様を水の上から見ている場合など



大域照明のアルゴリズム

- 目に届くまでの光路(パス)により様々なアルゴリズム
  - 光がどういう順番でどういう材質の表面で反射・屈折するか

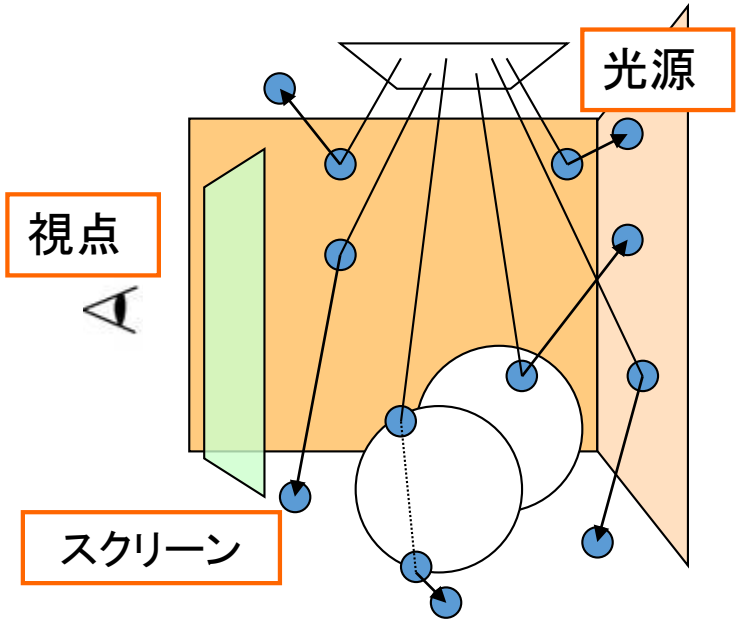
- 古典的レイトレーシング (Classical ray tracing)
- ラジオシティ (Radiosity; 相互反射法)
- パストレーシング (Path tracing)
- 双方向パストレーシング (Bi-directional Path Tracing)
- **フォトンマッピング (Photon mapping)**
- メトロポリス光伝達 (Metropolis light transport, MLT)
- 漸進的フォトンマッピング (Progressive photon mapping)

フォトンマッピング (Photon Mapping)

- 2 段階に分けて計算
  - 最初に光源から追跡、次に目から追跡

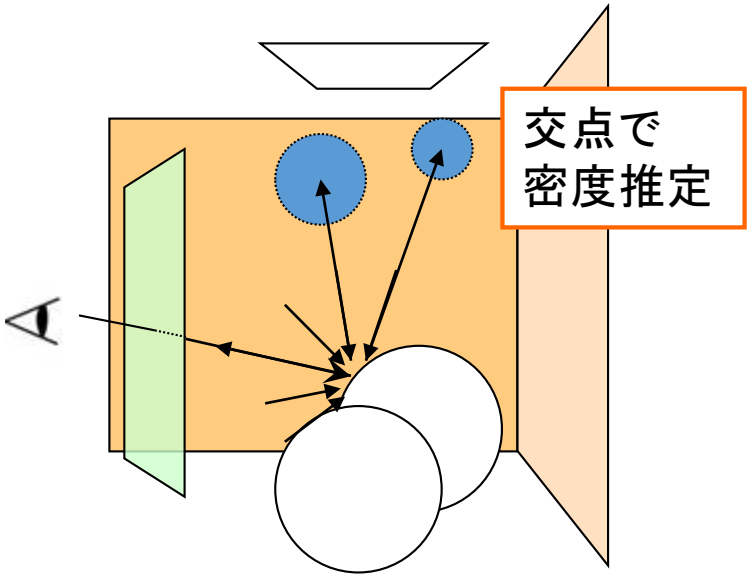
①フォトン (光子) トレース

- 光源からフォトンを放射
- 衝突面にフォトンを格納

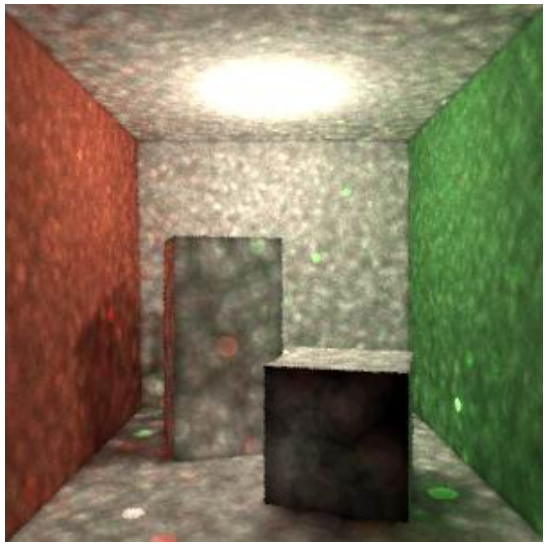


②ファイナルギャザリング(FG)

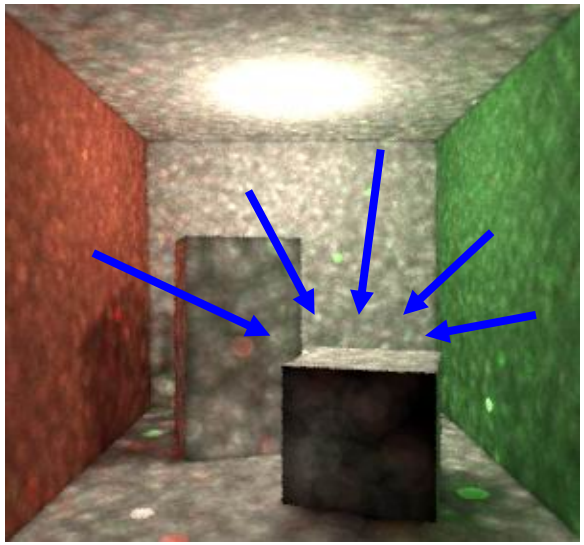
- 各視線との交点で輝度を積分



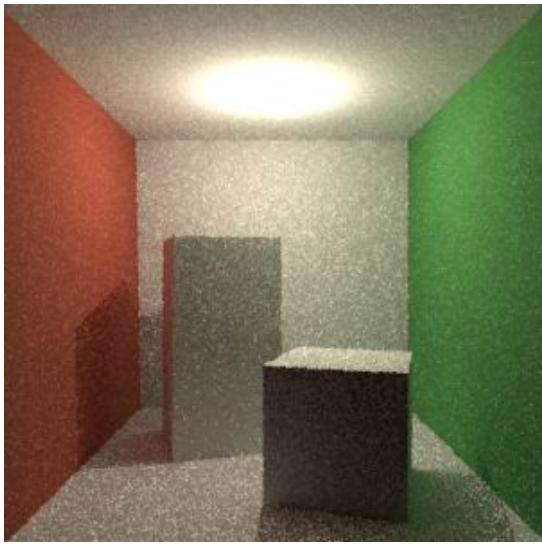
フォトンマッピング (Photon Mapping)



フォトントレースの結果
色ムラが目立つ



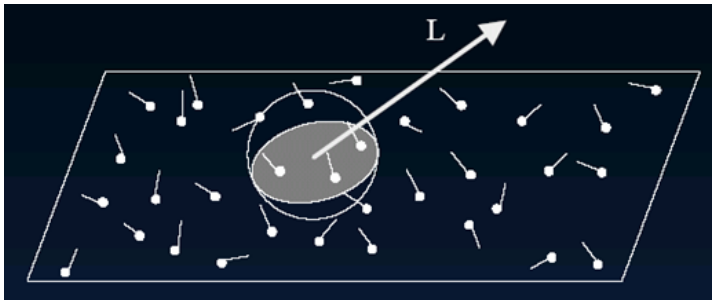
周囲の入射光を積分
(ファイナルギャザリング)



計算結果

フォトンマッピングの特徴

- 長所
  - 効率的
  - 集光模様専用のフォトンマップを作ることで、集光模様もうまく表現できる



- 短所
  - 計算途中の期待値が真の値からズレる
    - フォトン探索半径をユーザが指定
  - フォトン探索半径の調整が面倒
    - 半径が小さいと色ムラ、半径が大きいとボケる

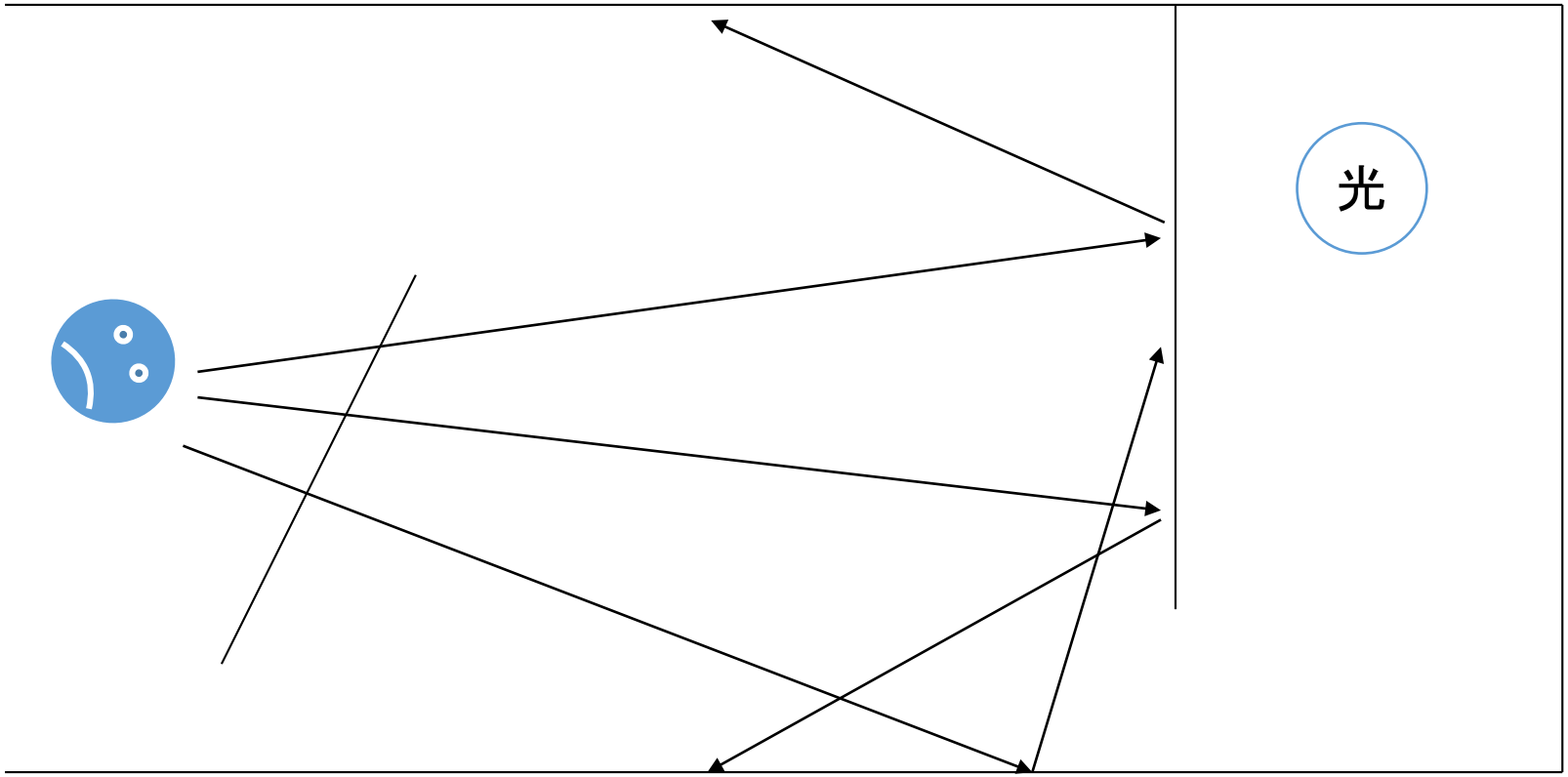
大域照明のアルゴリズム

- 目に届くまでの光路(パス)により様々なアルゴリズム
  - 光がどういう順番でどういう材質の表面で反射・屈折するか

- 古典的レイトレーシング (Classical ray tracing)
- ラジオシティ (Radiosity; 相互反射法)
- パストレーシング (Path tracing)
- 双方向パストレーシング (Bi-directional Path Tracing)
- フォトンマッピング (Photon mapping)
- **メトロポリス光伝達 (Metropolis light transport, MLT)**
- 漸進的フォトンマッピング (Progressive photon mapping)

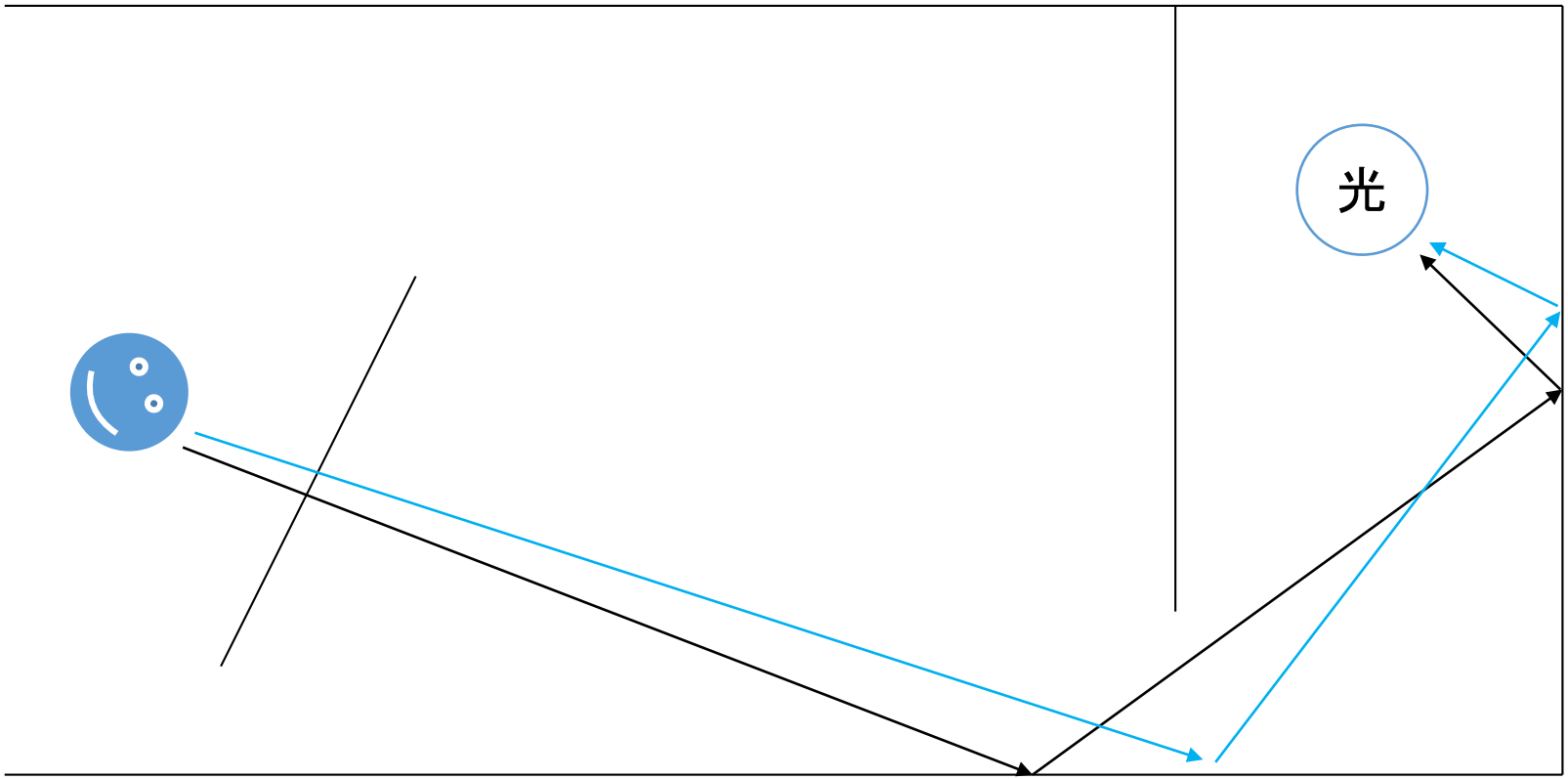
メトロポリス光伝達 (MLT)

- 光がシーン中の一部から漏れている場合、
  目まで届く光路を発見しづらい



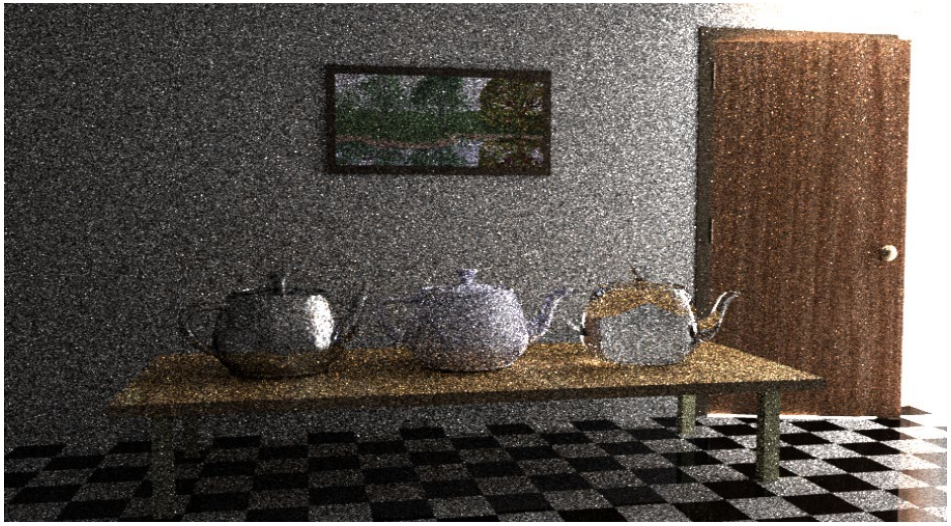
メトロポリス光伝達 (MLT)

- 光源から目に届く光路を発見したら、それを
  つなぎ替える or ズラして新しい光路を増やす



双方向パストレーシングとの比較

- ドアの隙間から光が漏れている (...イヤらしい...) 例



双方向パストレーシング
(40 サンプル / 画素)

メトロポリス光伝達
(同じ計算時間、
平均 250 摂動 / 画素)

メトロポリス光伝達 (MLT) の特徴

- 長所
  - より寄与の大きい (= 明るい) 光路を効率的に見つけられる
  - 特に前のページのような、ごく一部から光が届く場合に有効

- 短所
  - 穴だらけのシーンなどでは、どれかの穴に捕らわれてむしろ収束が遅くなってしまう場合がある
  - どのように光路をズラすのがよいかはシーン依存

大域照明のアルゴリズム

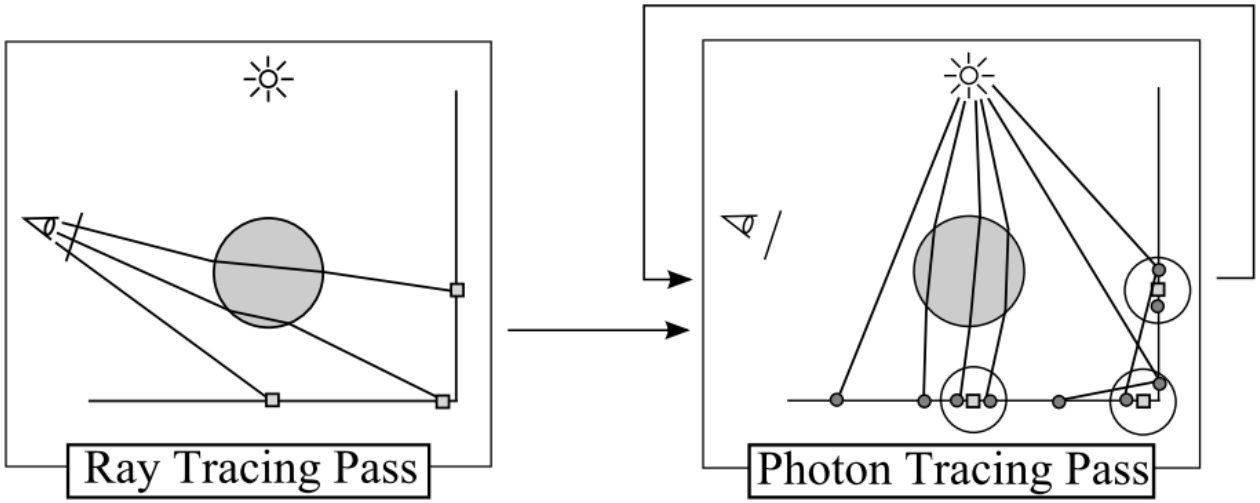
- 目に届くまでの光路(パス)により様々なアルゴリズム
  - 光がどういう順番でどういう材質の表面で反射・屈折するか

- 古典的レイトレーシング (Classical ray tracing)
- ラジオシティ (Radiosity; 相互反射法)
- パストレーシング (Path tracing)
- 双方向パストレーシング (Bi-directional Path Tracing)
- フォトンマッピング (Photon mapping)
- メトロポリス光伝達 (Metropolis light transport, MLT)
- **漸進的フォトンマッピング (Progressive photon mapping)**

漸進的フォトンマッピング

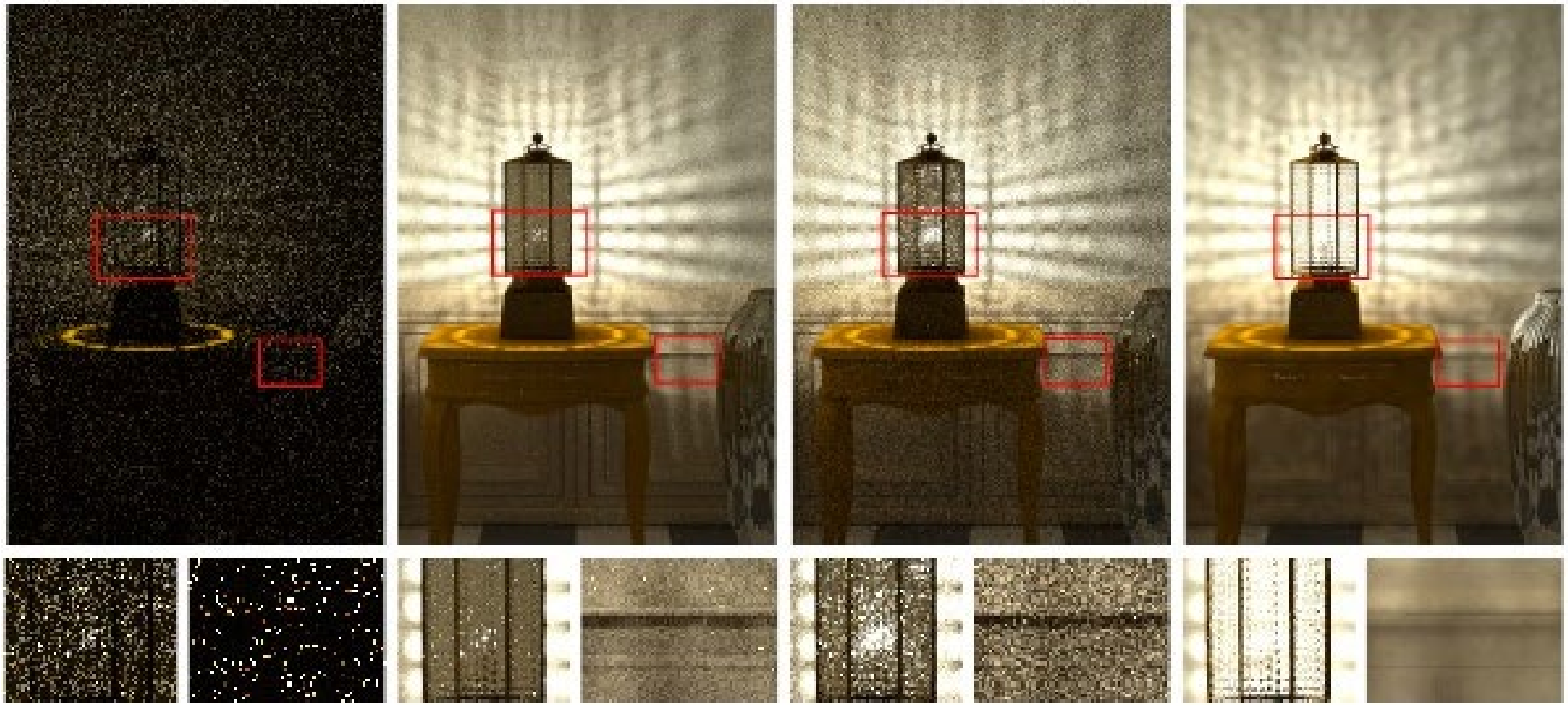
- 通常のフォトンマッピングと計算順を入れ替え
  - 目からのレイトレーシングで 3D 計算点を見つける
  - フォトントレーシングを繰り返す
- フォトンの寄与は計算点に蓄積して廃棄

  → 無限個のフォトンを扱える、
  途中の様子を確認できる (漸進的)
- フォトン探索半径を自動的かつ徐々に収縮



漸進的フォトンマッピング

- 集光模様が支配的なシーン … 22 時間での比較



Path tracing

Bi-directional path tracing

Metropolis light transport

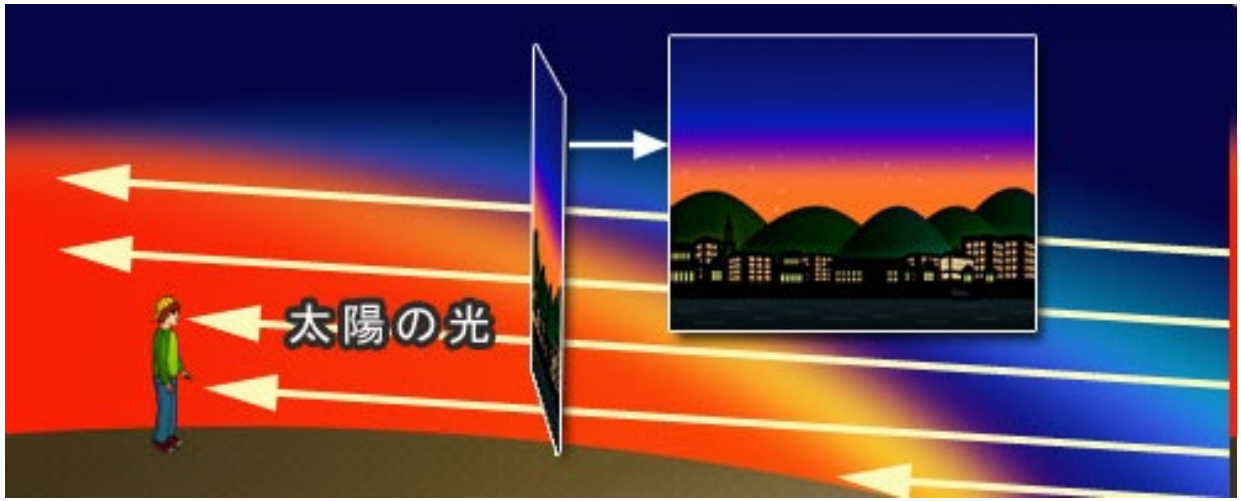
Progressive photon mapping

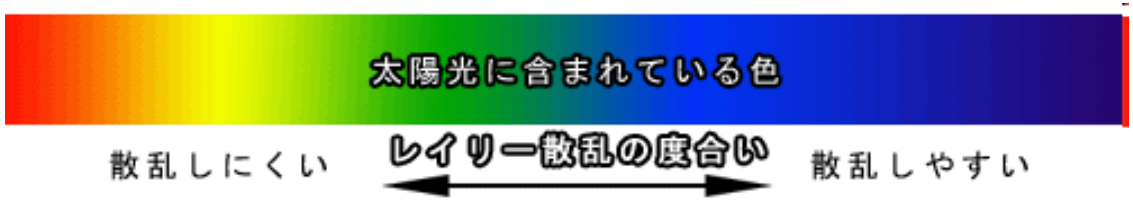
光の散乱

- ここまでは光が空気 (or 真空中) を進むと仮定
- 空気中に微粒子があると光が散乱される
  - 関与媒質 … 煙、水蒸気、炎、肌、植物、ミルク、など

参考: レイリー散乱で決まる空の色

- 日中は、太陽光の短波長の (= 青っぽい) 成分が空気中の微粒子で散乱される → 青空
- 夕焼け時には、光路長が長くなり、青い成分が途中で飛び散って赤い成分が残る → 赤い夕焼け





光の散乱・減衰現象を考慮した表示

■図4.45——光の散乱・減衰現象を考慮した表示

[a] 霧の表示

[b] 夕焼け空と夕焼けによって照らされた建物

（提供: 広島大学 知的システムモデリング研究室）

「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）

雲 (水蒸気) による光の散乱

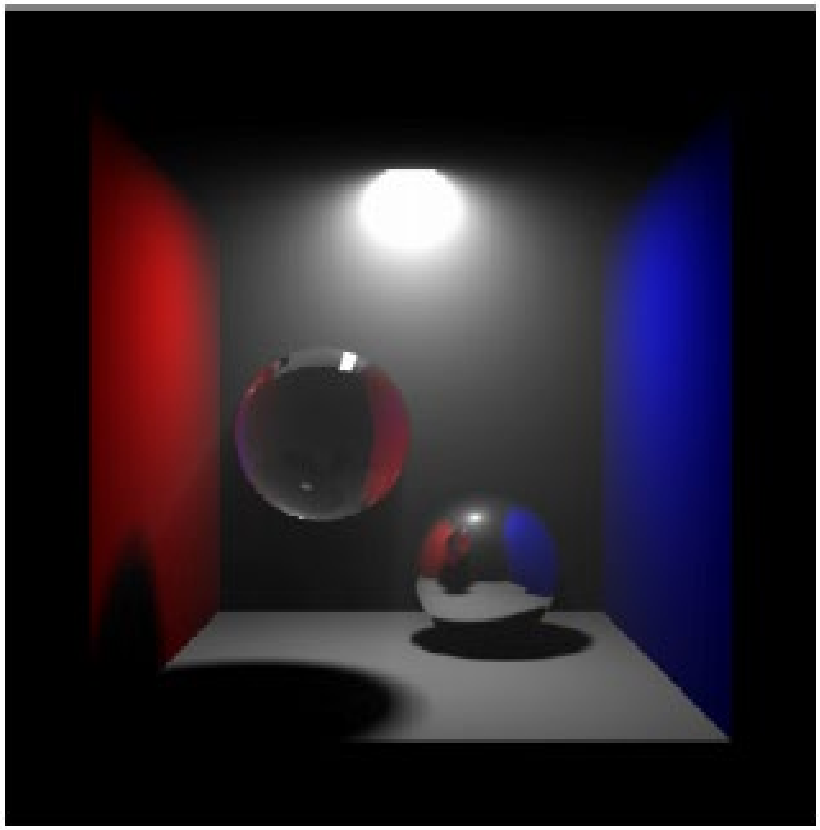
- 一次散乱: 光が微粒子で 1 回だけ散乱される
- 多重散乱: 光が微粒子に何度も衝突して散乱される
- 雲などでは多重散乱を考慮するかしないかで
  雲の陰影が大きく変わる

一次散乱

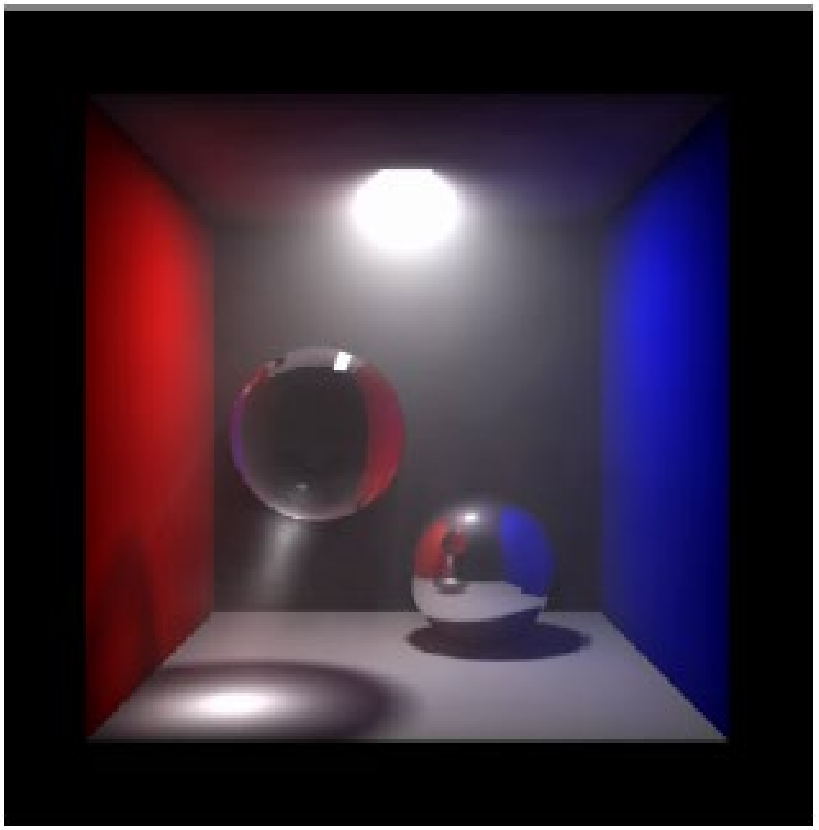
多重散乱

関与媒質を考慮した大域照明

- 関与媒質で色のにじみや集光模様が現れる



関与媒質なし

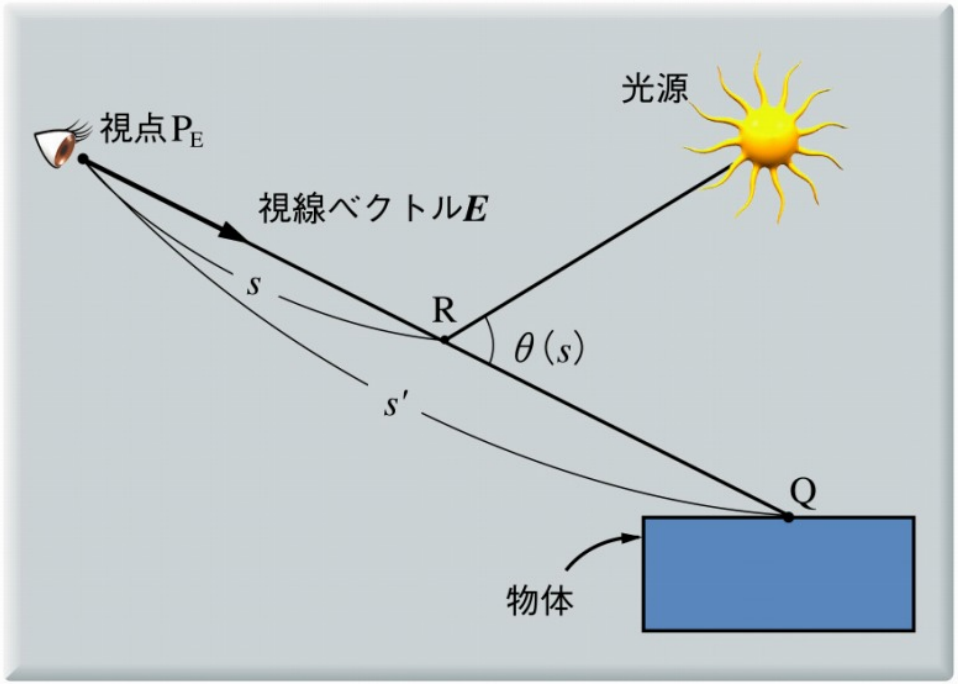


関与媒質あり

計算方法の例:レイマーチング法

- レイに沿って微小距離ずつ移動しながら、視点に到達するまでの減衰を考慮して散乱光を加算
  - 光路長に応じて光の強さが指数関数的に減衰すると仮定

■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）

1. 点Qからの反射光成分

2. 光路P_EQ間の点Rにおいて，光源からの光が視点方向へ散乱された光の成分

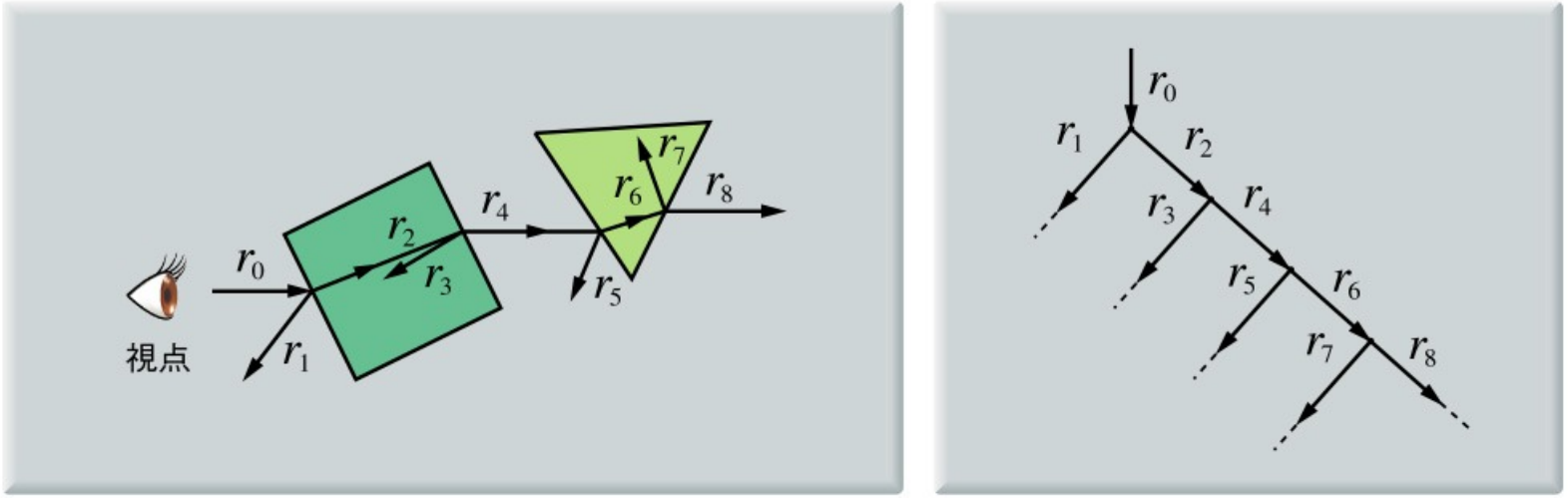
前回学習した
レイトレーシングの続き

鏡面反射、透過、屈折

- 鏡面反射：正反射方向の物体が映る
- 透過・屈折：透過した先の物体が透けて見える

- レイの追跡で実現する

■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

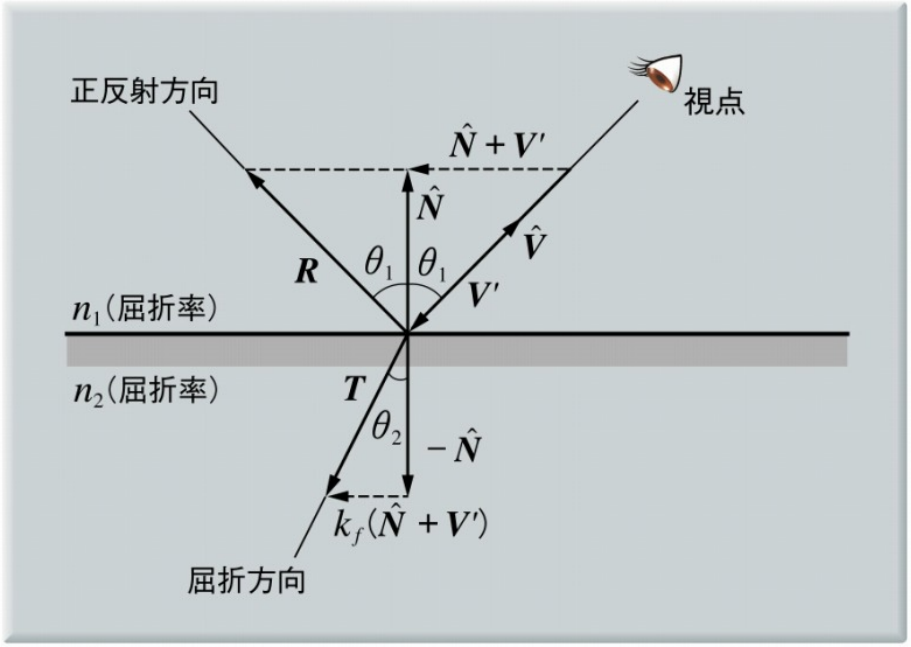
[b] 光線の追跡過程の二分木表現

「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）

正反射方向のレイの算出

- 入射角＝反射角
- 法線ベクトルと視点方向から正反射方向ベクトルを求める

■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）

正反射方向ベクトルR

$$\mathbf{R} = \mathbf{V}' + 2\hat{\mathbf{N}}$$

$$\mathbf{V}' = -\frac{\hat{\mathbf{V}}}{\hat{\mathbf{V}} \bullet \hat{\mathbf{N}}}$$

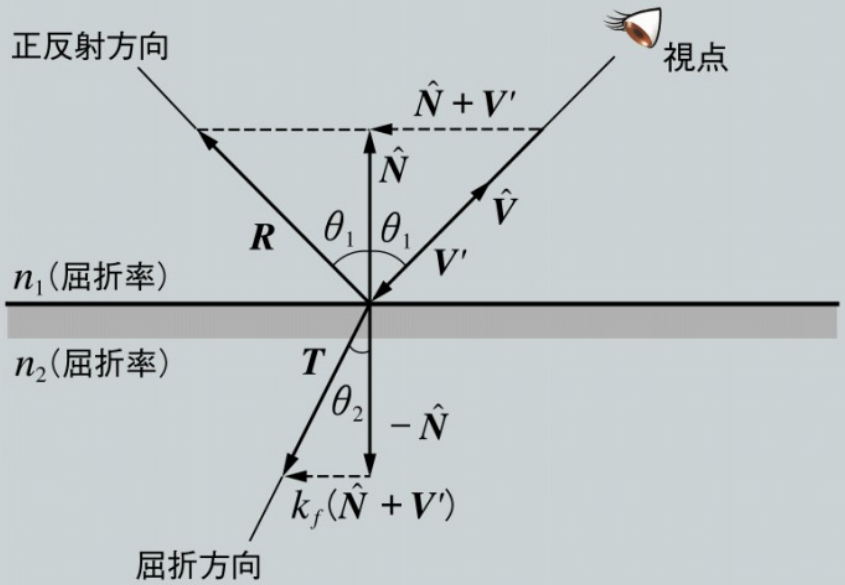
正反射方向の単位ベクトル

$$\|\mathbf{R}\| = \frac{1}{\hat{\mathbf{V}} \bullet \hat{\mathbf{N}}}$$

$$\hat{\mathbf{R}} = \frac{\mathbf{R}}{\|\mathbf{R}\|} = \hat{\mathbf{V}} \bullet \hat{\mathbf{N}}\ \mathbf{R}$$

屈折方向のレイの算出

■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）

面の法線ベクトル，視点方向ベクトル，2つの媒質の屈折率（n1, n2）を既知として，屈折方向ベクトルTを計算

$$\mathbf{T} = k_f\left(\hat{\mathbf{N}} + \mathbf{V}'\right) - \hat{\mathbf{N}}$$

k_f: 屈折の程度を表す係数，これを求れば良い

$n_1 sin\vartheta_1 = n_2 sin\vartheta_2$（スネルの法則）

$$n = \frac{n_2}{n_1} = \frac{sin\theta_1}{sin\theta_2}$$

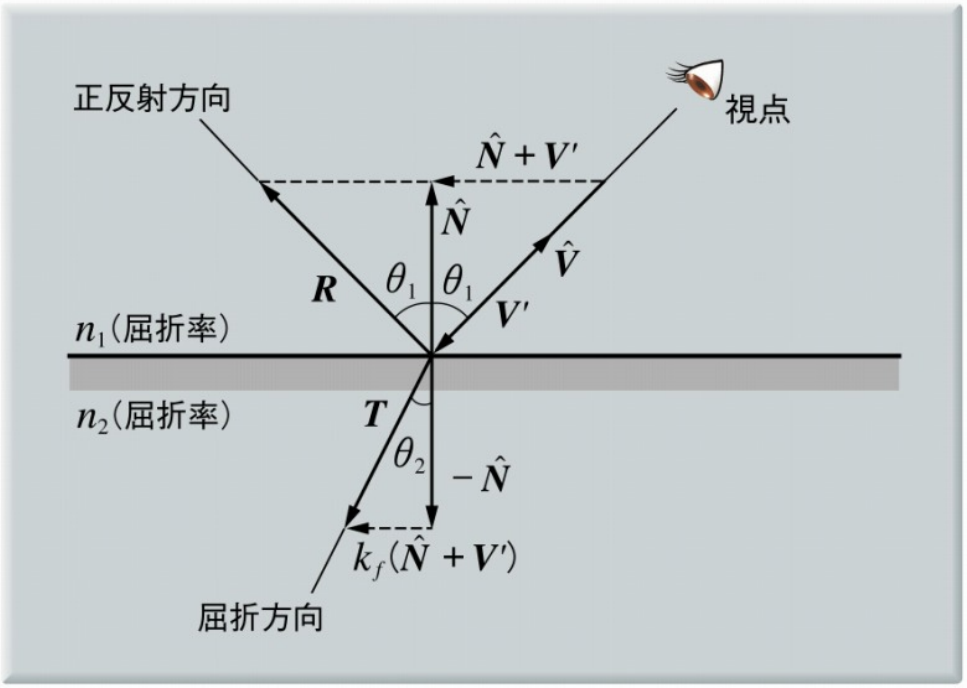
$$sin\theta_1 = \frac{\left\|\hat{\mathbf{N}} + \mathbf{V}'\right\|}{\left\|\mathbf{V}'\right\|}$$

$$sin\theta_2 = \frac{\left\|k_f\left(\hat{\mathbf{N}} + \mathbf{V}'\right)\right\|}{\|T\|} = \frac{k_f\left\|\hat{\mathbf{N}} + \mathbf{V}'\right\|}{\left\|k_f\left(\hat{\mathbf{N}} + \mathbf{V}'\right) - \hat{\mathbf{N}}\right\|}$$

$$\therefore n = \frac{\left\|k_f\left(\hat{\mathbf{N}} + \mathbf{V}'\right) - \hat{\mathbf{N}}\right\|}{k_f\left\|\mathbf{V}'\right\|}$$

屈折方向のレイの算出

■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）

$$n=\frac{\left\|k_f\left(\hat{\mathbf{N}}+\mathbf{V}'\right)-\hat{\mathbf{N}}\right\|}{k_f\left\|\mathbf{V}'\right\|}$$

両辺を2乗、$\left(\hat{\mathbf{N}}+\mathbf{V}'\right)\cdot\hat{\mathbf{N}}=\mathbf{0}$ より

$$n^2=\frac{{k_f}^2\left\|\hat{\mathbf{N}}+\mathbf{V}'\right\|^2+\left\|\hat{\mathbf{N}}\right\|^2}{{k_f}^2\left\|\mathbf{V}'\right\|^2}$$

$$\therefore k_f=\frac{1}{\sqrt{n^2\left\|\mathbf{V}'\right\|^2-\left\|\hat{\mathbf{N}}+\mathbf{V}'\right\|^2}}$$

反射率と透過率の算出

- 異なる屈折率を持つ媒質境界での反射率

フレネルの式から算出

$$k_r = \frac{1}{2}\left\{\frac{sin^2(\theta_1-\theta_2)}{sin^2(\theta_1+\theta_2)} + \frac{tan^2(\theta_1-\theta_2)}{tan^2(\theta_1+\theta_2)}\right\}$$

- 各レイにおける最終的な光の強さ

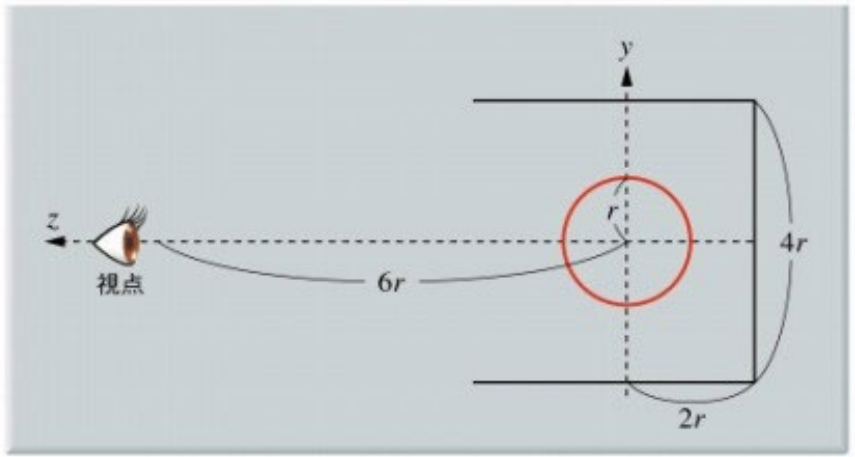
$$I_v = k_r I_r + k_t I_t \qquad \text{ただし } k_t + k_r = 1$$

$k_r I_r$：反射方向からの光の強さ

$k_t I_t$：屈折方向からの光の強さ

■図4.42——屈折率の違いによる表示例

[a] 屈折率の違いによる画像の変化



[b] 上図[a]における球と背景の配置

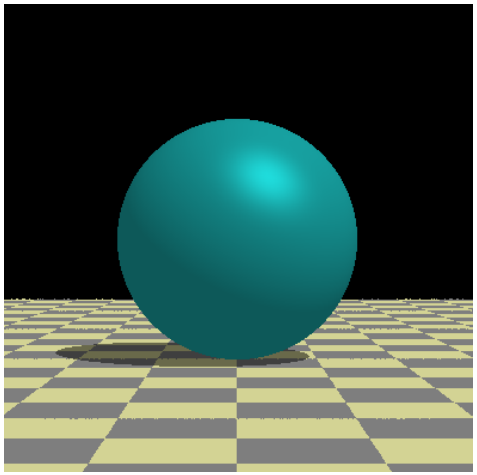
「コンピュータグラフィックス」2004年 / 財団法人画像情報教育振興協会（CG-ARTS協会）

課題について

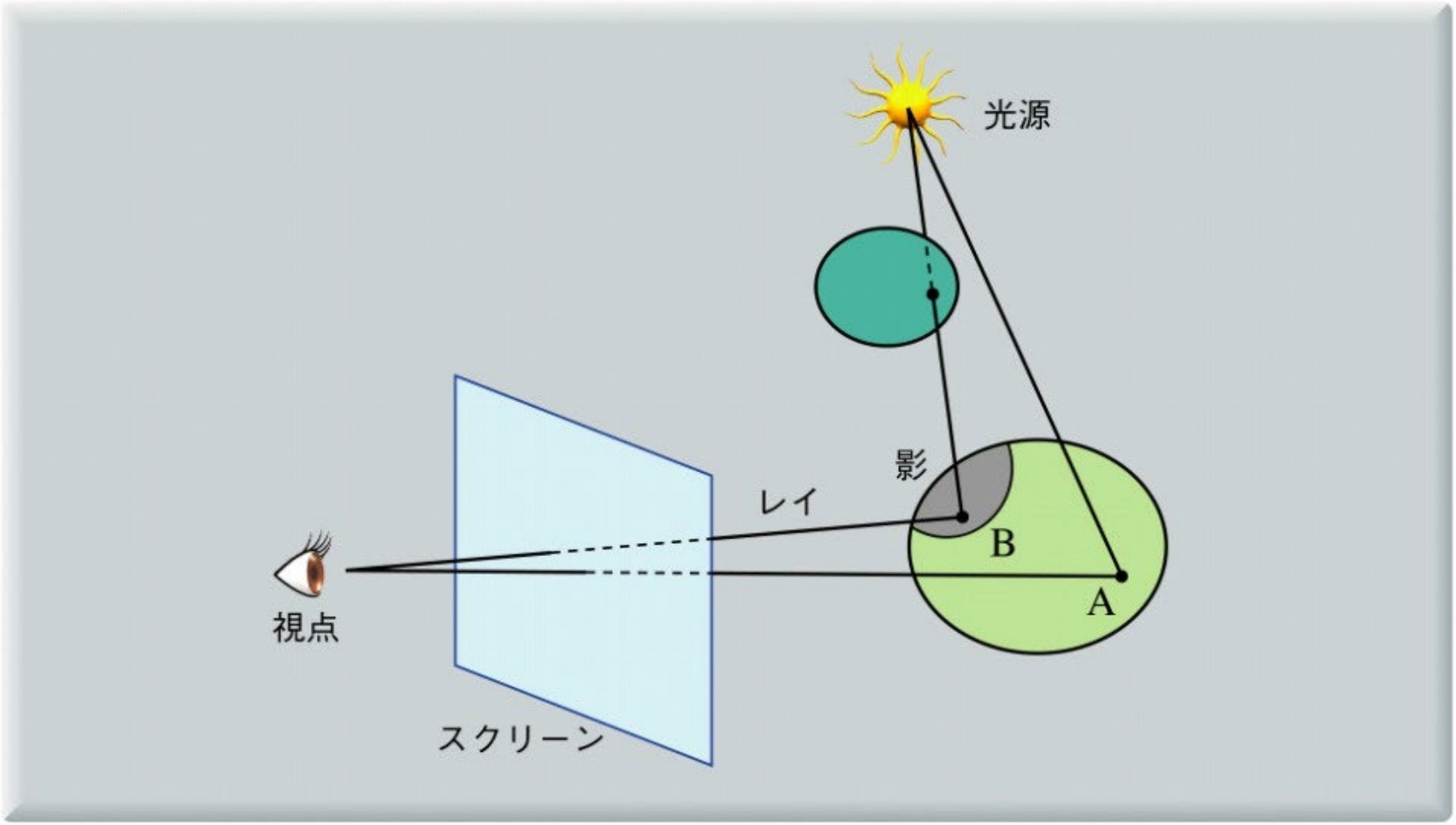
課題

- 床面に置かれた球体を表示する
- 床面には球体の影を描画する
- オーバーサンプリング (スーパーサンプリング) による、アンチエイリアシングを実装する
  - 1つのピクセルを細かく分割（サブピクセル）
  - サブピクセル単位で色を計算（通常のレイトレーシングと同じ方法）
  - サブピクセルの色の平均を、ピクセルの色とする

課題のシーン



■図4.52――レイトレーシング法による影付け



「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）

注）課題は平行光

オーバーサンプリング