

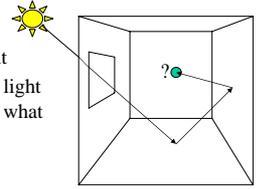
コンピュータグラフィックス (ビジュアルコンピューティング論) (エンターテインメントテクノロジー研究)

大局照明

西田 友是

大局照明: Global Illumination

- Light sources emit light
- Surface reflect or absorb light
- We want to know how much light reaches the image plane, and what color it is
 - Depends on the geometric arrangement and the surface properties
- First, we need to describe lights and surfaces



The 'Rendering Equation'

- Jim Kajiya (Current head of Microsoft Research) developed this in 1986

$$I(x, x') = g(x, x') \left[\varepsilon(x, x') + \int_S \rho(x, x', x'') I(x', x'') dx'' \right]$$



- $I(x, x')$ = total intensity from point x' to x
- $g(x, x')$ = 0 when x/x' are occluded
= $1/d^2$ otherwise (d = distance between x and x')
- $\varepsilon(x, x')$ = intensity emitted by x' to x
- $\rho(x, x', x'')$ = intensity of light reflected from x'' to x through x'
- S = all points on all surfaces

The 'Rendering Equation'

- The light that hits x from x' is the direct illumination from x' and all the light reflected by x' from all x''
- To implement:
 - Must handle recursion effectively
 - Must support diffuse and specular light
 - Must model object shadowing

The 'Rendering Equation'

- What's really hard about computing this?

$$I(x, x') = g(x, x') \left[\varepsilon(x, x') + \int_S \rho(x, x', x'') I(x', x'') dx'' \right]$$

- The integral...
 - How can one compute $I(x, x')$ for all points x and x' ?
 - Approximate!!!

相互反射とは



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

相互反射について

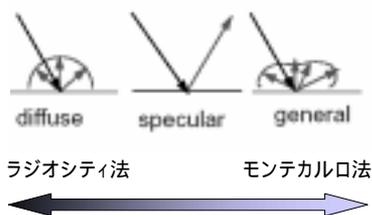
- Rendering Equation

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

- 難しさ: 反射光 = 他の点への入射光

相互反射について

- 反射モデル



Radiosity法(1)

- 面をelementに分割する
- 各elementのradiosity B_i 間の関係を定める

$$B_i = E_i + \sum_j B_j F_{ij}$$

- form factor F_{ij} を効率よく求める必要がある

Radiosity法(2)

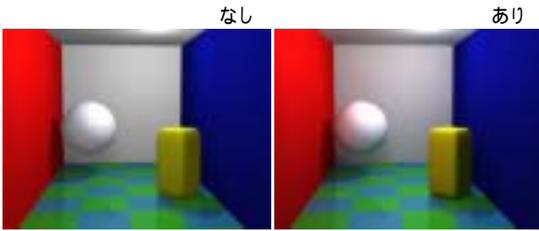
- 次の連立方程式が得られる

$$\begin{bmatrix} 1 - {}_1F_{11} & - {}_1F_{12} & \dots & - {}_1F_{1n} \\ - {}_2F_{21} & 1 - {}_2F_{22} & \dots & - {}_2F_{2n} \\ \dots & \dots & \dots & \dots \\ - {}_nF_{n1} & - {}_nF_{n2} & \dots & 1 - {}_nF_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \dots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \dots \\ E_n \end{bmatrix}$$

Radiosity法(3)

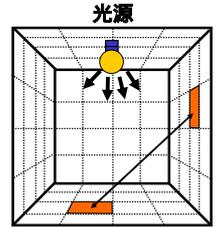
- 連立方程式を解いてradiosity B_i を求める
- renderingを行なう
- 視点の位置を変えるだけなら, rendering部分のみ再計算すればよい

Radiosity法の効果



ラジオシティ法:基礎

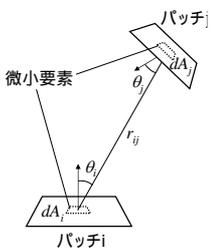
- View-independent solution
- パッチに分割
- パッチ間の光の授受を計算
- 連立一次方程式
- 鏡面反射は難しい



ラジオシティ法:基礎

- 拡散反射:

$$B_{out} = \int (\rho_d / \pi) B_{in} \cos \theta_m d\omega_m$$



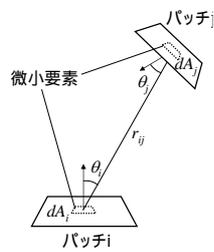
$$\begin{cases} B_{in} = B_j \cos \theta_j \\ d\omega_m = \frac{dA_j}{r_{ij}^2} \\ \theta_m = \theta_i \end{cases}$$

$$B_i = \frac{\rho_d}{\pi} \frac{B_j \cos \theta_i \cos \theta_j}{r_{ij}^2} dA_j$$

ラジオシティ法:基礎

- 拡散反射:

$$B_i = \frac{\rho_d}{\pi} \frac{B_j \cos \theta_i \cos \theta_j}{r_{ij}^2} dA_j$$



$$B_i = \int_{A_j} \frac{\rho_d}{\pi} \frac{B_j \cos \theta_i \cos \theta_j}{r_{ij}^2} dA_j$$

$$\int_{A_i} B_i dA_i = \int_{A_i} \int_{A_j} \frac{\rho_d}{\pi} \frac{B_j \cos \theta_i \cos \theta_j}{r_{ij}^2} dA_j dA_i$$

$$B_i = \rho_d \frac{B_j}{\pi A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{r_{ij}^2} dA_j dA_i$$

フォームファクタ(形態係数)

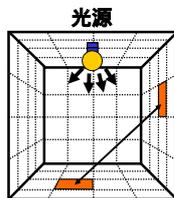
ラジオシティ法:基礎

- 拡散反射
- 連立一次方程式

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

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

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



ラジオシティ法:基礎

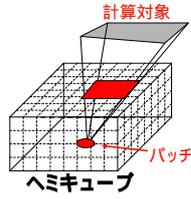
$$B_i = E_i + \sum_{j=1, j \neq i}^n \rho_j F_{ij} B_j$$

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

- 数値解法
 - フォームファクタの計算
ヘミキューブ法とレイキャスティング法
 - 方程式の解き方
GatheringとShooting

フォームファクタの計算

- ヘミキューブ法
 - パッチを中心にヘミキューブ(半立方体)を設置, 周囲の物体をキューブ表面に投影
 - 投影画像中のパッチ面積からフォームファクタを算出
 - 一画素がもつフォームファクタ
- 利点
 - グラフィックスハードウェアの利用が可能
- 欠点
 - 計算精度は投影面の解像度に依存

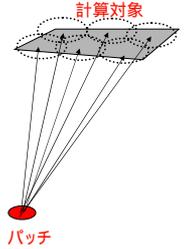


フォームファクタの計算

- レイキャスティング法
 - サンプル点を発生
 - 円盤と点とのフォームファクタを計算

$$dF_{ij} = V_{ij} \frac{\cos \theta_i \cos \theta_j}{\pi r_{ij}^2 + a}$$

- 利点
 - 誤差が少ない
- 欠点
 - 計算に時間がかかる

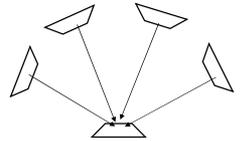


方程式の解き方

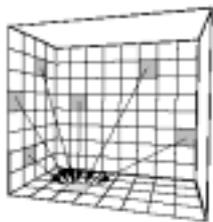
- 連立一次方程式 $B_i = E_i + \sum_{j=1, j \neq i}^n \rho_j F_{ij} B_j$
- 巨大な係数マトリックス
 - パッチ数xパッチ数の係数マトリックス
 - Ex. $10^4 \times 10^4 \times 3 \times 4 [\text{byte}] = 1.2 \times 10^9 > 1 \text{GB!}$
 - 全ての係数の記憶は難しい
- フォームファクタを逐次計算しながら解を更新
 - Gathering法
 - Shooting法 (Progressive Refinement)

方程式の解き方

- Gathering法
 - 直射光計算(B_i の初期値)
 - 収束するまで以下を繰り返す
 - 各パッチについて
 - B_i の更新
- 遅い
 - $B_i = E_i + \sum_{j=1, j \neq i}^n \rho_j F_{ij} B_j$



Gathering

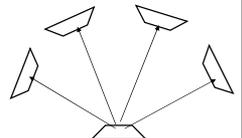


```

1  for ( all i ) B_i = starting guess ;
2  while ( not converged ) {
3    for ( each i )
4      B_i = E_i - \sum_{j=1, j \neq i}^n \rho_j F_{ij} B_j ;
5  }
6  output B ;
    
```

方程式の解き方

- Shooting法
 - 直射光計算(B_i の初期値)
 - unshotエネルギー $U_i = B_i$
 - 収束するまで以下を繰り返す
 - Shootingパッチ k の選択
 - 全パッチの B_i, U_i を更新
 - $B_i = B_i + \rho_i F_{ik} U_k$
 - $U_i = U_i + \rho_i F_{ik} B_k$
 - $U_k = 0$
- 早い段階で真値に近づく



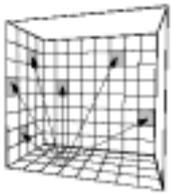
Progressive refinement

```

1: for all i:
2:    $\Delta_i = E_i$ 
3:    $r_i = E_i$ 
4:   }
5: while (not converged)
6:   pick a patch with most  $\Delta_i$  in scene
7:    $\Delta = \Delta_i + r_i \Delta_i$ 
8:    $r_i = r_i + \Delta_i$ 
9:   for every other j:
10:     $\Delta_j = \Delta_j + r_j \Delta_i + \Delta_i$ 
11:   }
12: output  $\Delta$ 

```

Technique for radiosity solution



```

1: for all i:
2:    $\Delta_i = E_i$ 
3:    $\Delta_i = \Delta_i$ 
4:   }
5: while (not converged)
6:   pick a patch with most  $\Delta_i + \Delta_i$  in scene
7:   for every other j:
8:     $\Delta_j = \Delta_j + r_j \Delta_i$ 
9:     $\Delta_j = \Delta_j + \Delta_i$ 
10:   }
11:   }
12:    $\Delta_i = \Delta_i$ 
13:   }
14:   }
15:   }
16:   }
17:   }
18:   }
19:   }
20:   }
21:   }
22:   }
23:   }
24:   }
25:   }
26:   }
27:   }
28:   }
29:   }
30:   }
31:   }
32:   }
33:   }
34:   }
35:   }
36:   }
37:   }
38:   }
39:   }
40:   }
41:   }
42:   }
43:   }
44:   }
45:   }
46:   }
47:   }
48:   }
49:   }
50:   }
51:   }
52:   }
53:   }
54:   }
55:   }
56:   }
57:   }
58:   }
59:   }
60:   }
61:   }
62:   }
63:   }
64:   }
65:   }
66:   }
67:   }
68:   }
69:   }
70:   }
71:   }
72:   }
73:   }
74:   }
75:   }
76:   }
77:   }
78:   }
79:   }
80:   }
81:   }
82:   }
83:   }
84:   }
85:   }
86:   }
87:   }
88:   }
89:   }
90:   }
91:   }
92:   }
93:   }
94:   }
95:   }
96:   }
97:   }
98:   }
99:   }
100:  }

```

Technique for progressive refinement

Gathering vs Shooting

Gathering

- For each patch:
 - Compute incoming energy from the rest of the scene
 - multiply by reflectivity
 - update radiosity
- Straightforward writing of the formula
- Keep going until convergence or bored.

shooting

- Store unshot radiosity
- Initially, unshot radiosity=emittance
- Pick patch with greatest unshot energy
 - send radiosity to all other patches
 - set unshot radiosity to 0
 - radiosity received is added to unshot B
- Keep going until convergence or good enough picture

計算例 (gathering)



計算例 Shooting

- フォームファクタ計算

- レイキャスティング

$$F_{ij} = \frac{\cos\theta_i \cos\theta_j}{\pi r_{ij}^2} V_{ij} A_j$$

- 解法

- Shooting
(Progressive Radiosity)

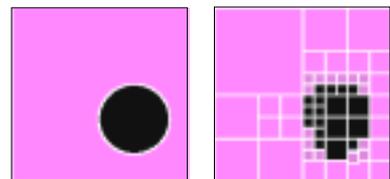


ラジオシティ法: 応用

- 高精度化・高速化手法
 - Hierarchical Radiosity
 - Higher Order Radiosity
- 鏡面反射の考慮
 - Ideal Specular Reflection
 - Glossy Specular Reflection

高精度化・高速化法

- Adaptive Refinement



高精度化・高速化法

- Hierarchical Radiosity法

- フォームファクタの性質

$$F_{ij} = \frac{\cos\theta_i \cos\theta_j}{\pi r_{ij}^2} V_{ij} A_j$$

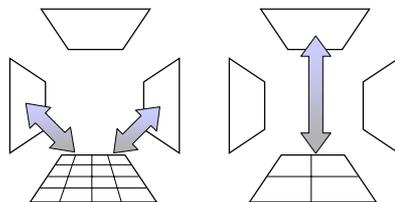
- 遠方の面同士 フォームファクタ小
誤差の影響小

- 近傍の面同士 フォームファクタ大
誤差の影響大

高精度化・高速化法

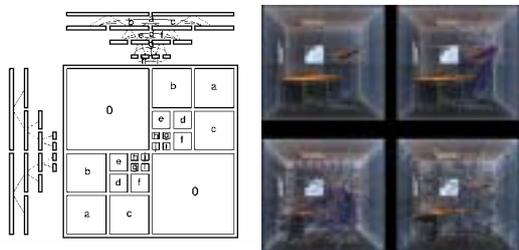
- Hierarchical Radiosity法

- 同じ面でもパッチ分割が変わる



高精度化・高速化法

- Hierarchical Radiosity



高精度化・高速化

- Hierarchical Radiosity



高精度化・高速化法

- Higher Order Radiosity

- 基本的ラジオシティ法は各パッチに単一のラジオシティ

- 複雑な輝度変化 パッチ数を増加 (h-refinement)

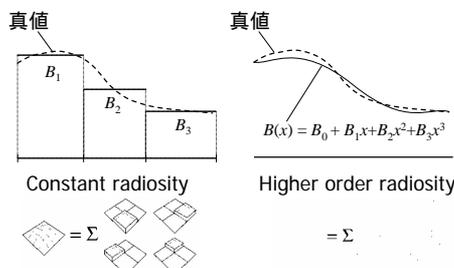


- 各パッチの輝度変化を基底関数表現

- 複雑な輝度変化 関数の次数を増加 (p-refinement)

高精度化・高速化法

- Higher Order Radiosity



高精度化・高速化法

- Higher Order Radiosity

$$B_{\text{total}}(s, t) = \sum_i B_i^l \mathcal{T}_i(s, t). \quad (\mathcal{T}_i \text{ は正規直交基底})$$

$$B_i(s, t) = E_i(s, t) + \sum_j \iint K_{ij}(s, t, u, v) B_j(u, v) du dv.$$

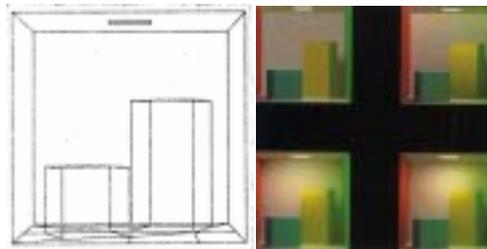
$$B_i(s, t) = E_i(s, t) + \sum_{j \neq i} B_j^l \iint K_{ij}(s, t, u, v) \mathcal{T}_j(u, v) du dv.$$

$$B_i^k = E_i^k + \sum_{j \neq i} B_j^l \left\langle \iint K_{ij}(s, t, u, v) \mathcal{T}_j(u, v) du dv \middle| \mathcal{T}_i(s, t) \right\rangle_{\mathcal{W}}. \quad \left. \vphantom{B_i^k} \right) \text{ (内積を計算)}$$

$$B_i^k - E_i^k = \sum_{j \neq i} B_j^l K_{ij}^{kl}.$$

高精度化・高速化法

- Higher Order Radiosity



高精度化・高速化法

- Higher Order Radiosity



ラジオシティ法: 応用

- 高精度化・高速化手法
 - Hierarchical Radiosity
 - Higher Order Radiosity
- 鏡面反射の考慮
 - 仮想空間による方法(完全鏡面反射)
 - Two Pass Method

鏡面反射の考慮

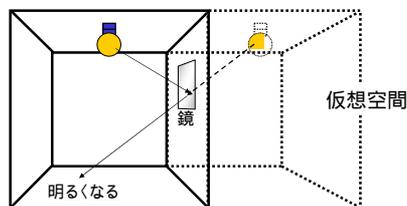
- Rendering Equation

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

- 反射光が方向に依存
- 反射光分布を覚えなくてはならない

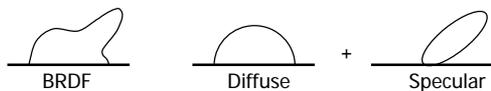
鏡面反射の考慮

- 仮想空間による方法(完全鏡面反射)



鏡面反射の考慮

- Two Pass Method



- Diffuse: 通常のRadiosity法
- Specular: レイキャスティング法

鏡面反射の考慮

- Two Pass Method



曲面光源の例 (相互反射を考慮)

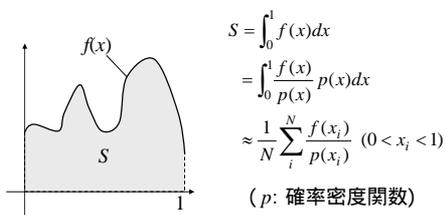


相互反射計算法

- 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(ラジオンシティ法)
 - **モンテカルロ法**

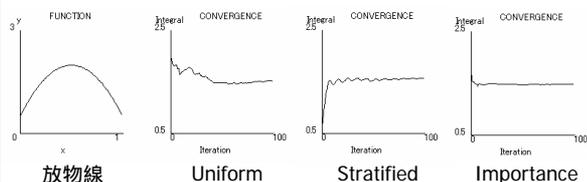
モンテカルロ法

- 積分計算を行う手法の一つ



モンテカルロ法

- 計算例



モンテカルロ法と相互反射

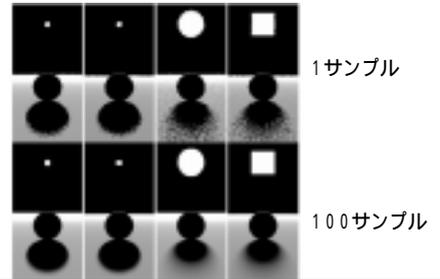
- Rendering Equation

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

- 一種の積分計算 (積分方程式)

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

- 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

Monte Carlo

What is Monte Carlo?

Simulation ... a random system is simulated

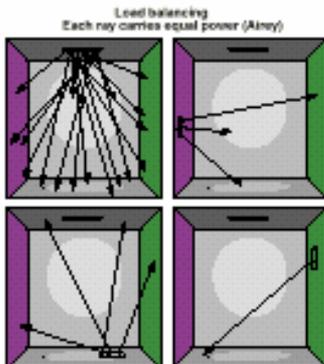
Integration ... random sampling is used to numerically evaluate an integral

Multiple Reflections

Change results with each change in each pixel in the path

Energy builds from each ray path with each reflection

Monte Carlo Radiosity



Monte Carlo Radiosity

Radiosity solution (no sampling)

Importance sampling

Monte Carlo Integration

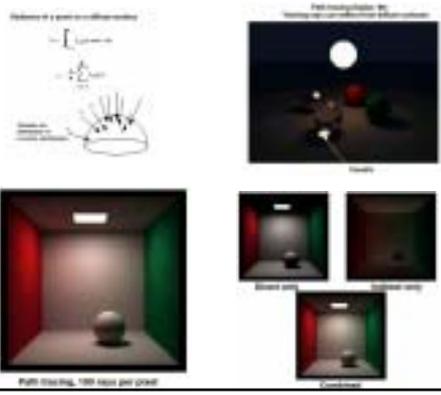
$$\int f(x) dx \approx \frac{1}{N} \sum_{i=1}^N f(x_i)$$

The random points x_i have probability density function p

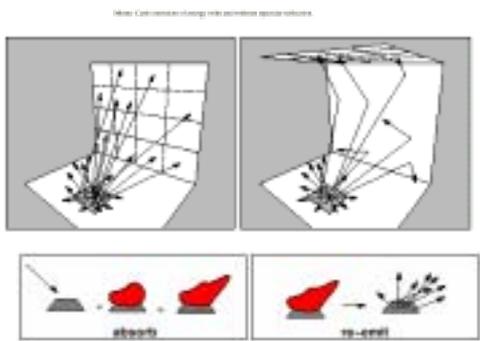
FORGET IT (for now)

Catch 22 of non-diffuse radiosity: as surfaces get more specular, there needs to be more zones, and each zone needs more directional precision.

Monte Carlo Radiosity



Monte Carlo Radiosity



Monte Carlo Radiosity

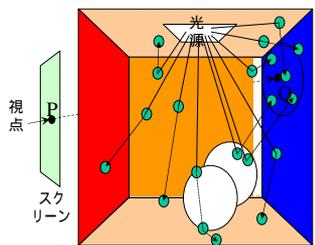


Figure 10: Constant indirect, and direct lighting. Note that the less sharp shading changes are in the direct component.



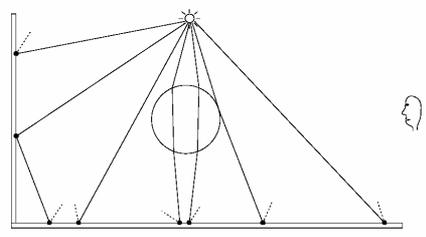
Figure 11: Left: Low resolution radiosity solution. Middle: Medium resolution solution. Right: Path tracing.

Photon Map法



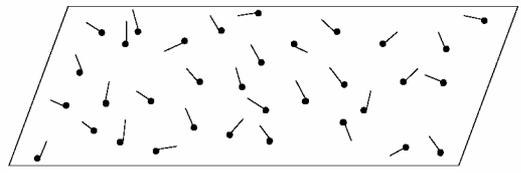
Photon Map法

- 光源から光をトレースする方法



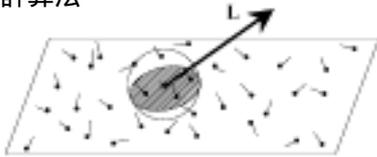
Photon Map法

- フォトンの場所と光の方向/強さを記憶



Photon Map法

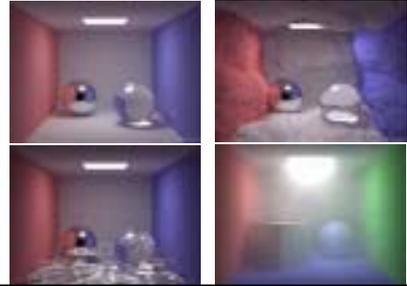
- 輝度計算法



$$L(\mathbf{x}, \vec{\omega}) \approx \sum_{p=1}^N f_r(\mathbf{x}, \vec{\omega}, \vec{\omega}_p) \frac{\Delta\Phi_p(\mathbf{x}, \vec{\omega}_p)}{\pi r^2}$$

Photon Map法

- 計算例



Photon Map法

- 計算例



Photon Map法



Refraction caustic

シェーディング(まとめ)

- 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
- 大域的照明モデル(Global Illumination)