

Lecture Notes #13 - Global Illumination

Reading: Angel: 16.10; Foley: p.792

Topics:

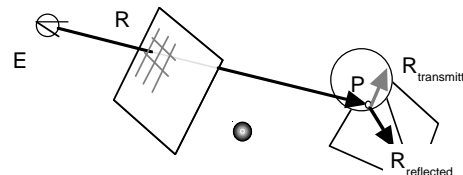
- Rendering equation

- Approximations

Ray tracing

Radiosity

Ray Tracing



Let $I(b, r)$ = intensity seen along direction r from point b .

$$I(E, R) = I_{direct} + I_{indirect}$$

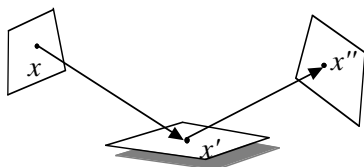
$$\approx I_{direct} + I_{reflected} + I_{transmitted}$$

I_{direct} = computed from the Phong model

$$I_{reflected} = \text{Reflectance} * I(P, R_{reflected})$$

$$I_{transmitted} = \text{Transmittance} * I(P, R_{transmitted})$$

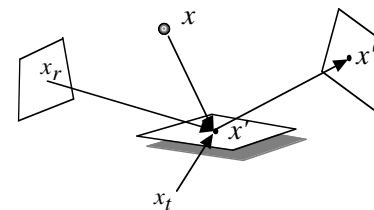
The Rendering Equation (Kajiya '86)



$$L(x' \rightarrow x'') = E(x' \rightarrow x'') + \int_x f_r(x, x', x'') L(x \rightarrow x') V(x, x') G(x, x') dx$$

- $L(x' \rightarrow x'')$ is the total *radiance* from x' to x''
- $E(x' \rightarrow x'')$ is the *emitted* radiance from x' to x''
 - non-zero for light sources
- f_r is the BRDF
- $V(x, x')$ is a *visibility* term:
 - 1 if x is visible from x' ; 0 otherwise
- $G(x, x')$ is a *geometry* term

Ray Tracing as Approximation to Rendering Equation



$$L(x' \rightarrow x'') = E(x' \rightarrow x'') + \int_x f_r(x, x', x'') L(x \rightarrow x') V(x, x') G(x, x') dx$$

- Sample the integrand where it is likely to be large: at points x such that:
 - x is at a light source
 - x is in the reflected direction
 - x is in the refracted direction

Ray Tracing

Gross approximation to the rendering equation.

accounts for:

- shadows
- refraction
- inter-object reflection

doesn't account for:

- caustics (focusing)
- finite light sources
- diffuse inter-object reflection

Relatively easy to implement.

Produces pretty nice images, although they're a little "too perfect".

Radiosity

- Radiosity = energy/unit area leaving a surface:

$$B(x) = \int_{\omega} L(x \rightarrow x') \cos \theta d\omega$$

The Radiosity Equation

- *Radiosity assumption*: all surfaces totally diffuse
- Simplifications to rendering equation:
 - BRDF comes outside the integral
 - Radiance leaving each surface is equal in all directions
 - => Radiosity describes everything
- Rendering equation becomes the radiosity equation:

$$B(x) = E(x) + \rho(x) \int_{x'} B(x') \frac{V(x, x') G(x, x')}{\pi} dx'$$

$$\rho(x) = \text{"hemispherical reflectance"} = k_d \pi \quad 0 \leq \rho(x) \leq 1$$

Solving the Radiosity Equation

$$B(x) = E(x) + \rho(x) \int_{x'} B(x') \frac{V(x, x') G(x, x')}{\pi} dx'$$

- Relatively hard:
 - Unknown $B(x)$ appears on both sides, once inside the integral, once outside.
 - Analytic solutions generally don't exist.
 - Must approximate using numerical methods.
- The "finite element" approach
 - Divide scene up into small surface patches (discretize).
 - Assume on each patch $i=1, \dots, n$:
 - Radiosity B_i is constant.
 - Emission E_i is constant.
 - Reflectance ρ_i is constant.

Finite Element Approach

$$\frac{1}{A_i} \int_{x \in i} B(x) dx = \frac{1}{A_i} \int E(x) dx + \frac{1}{A_i} \int_{x \in i} \rho \int_{x' \notin i} B(x) \frac{V(x, x') G(x, x')}{\pi} dx' dx$$

$$B_i = E_i + \rho_i \frac{1}{A_i} \sum_{j=1}^n \int_{x \in i} \int_{x' \in j} B_j \frac{V(x, x') G(x, x')}{\pi} dx' dx$$

$$= E_i + \rho_i \sum_{j=1}^n B_j \left[\int_{x \in i} \int_{x' \in j} \frac{V(x, x') G(x, x')}{\pi} dx' dx \right]$$

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

F_{ij}

"form factor"

Finite Element Approach

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

$$F_{ij} = \frac{1}{A_i} \iint_{x \in i, x' \in j} \frac{V(x, x') G(x, x')}{\pi} dx dx'$$

- In matrix form:

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix} + \begin{bmatrix} \rho_1 F_{11} & \rho_1 F_{12} & \cdots & \rho_1 F_{1n} \\ \rho_2 F_{21} & \rho_2 F_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \rho_n F_{n1} & \cdots & \rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix}$$

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n1} & \cdots & 1 - \rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

Issues

- How to compute form factors?
- How many patches to use?
- How best to solve the linear system?
 - It is large and dense

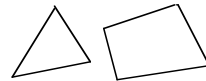
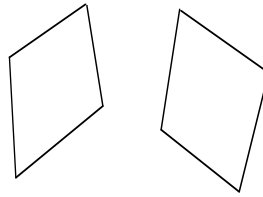
Computing Form Factors

$$F_{ij} = \frac{1}{A_i} \iint_{x \in i, x' \in j} \frac{V(x, x') G(x, x')}{\pi} dx dx'$$

- Quadrature using ray casting for visibility
- Use z-buffer hardware (hemi-cube [Cohen&Greenberg '85])

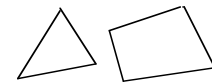
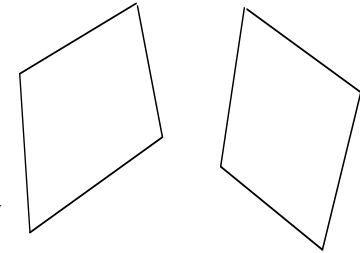
Standard Radiosity

- Each entry of form factor matrix represents an "interaction"
- n^2 interactions
- Small patches needed for accurate close interaction
- Small patches are overkill for distant interactions



Hierarchical Radiosity

- Hanrahan, Salzman, Aupperle, Siggraph '91
- Hierarchically decompose
 - patches (into a quadtree)
 - FF matrix (matrix never explicitly formed)
- Claim: $O(f(e)n)$ interactions for accuracy tolerance e



Basic Algorithm

- Compute interactions:

```
Refine(Patch *r, *s, eps) {
    Frs = FormFactor( r, s)
    Fsr = FormFactor( s, r)
    if (accurate to within eps) {
        Link(r,s);
    } else {
        if (SplitAndRefine(s)) {
            Subdiv(s)
            Refine(r, s->ne, eps)
            Refine(r, s->nw, eps)
            Refine(r, s->se, eps)
            Refine(r, s->sw, eps)
        } else {...}
    }
}
```

- Solve:

```
Solve() {
    repeat {
        foreach top level patch r {
            Gather(r)
        }
        foreach top level patch r {
            Push(r)
            Pull(r)
        }
    } until converged;
}
```

Basic Algorithm

```
Gather(Patch *r) {
    /* Gather radiosity to r */
    if (r) {
        r->B = r->emission
        for each link L btw r & s {
            r->B += L->FF * r->rho * s->B
        }
        Gather(r->nw)
        Gather(r->ne);
        Gather(r->sw)
        Gather(r->se);
    }
}
```

```
Push(Patch *r) {
    /*
        Add radiosity at r to radiosities of
        children
    */
}

Pull(Patch *r) {
    /*
        Set radiosity of each node to
        area weighted average of
        children
    */
}
```