LOCAL VS. GLOBAL ILLUMINATION

Light = emitted + reflected + yadda yadda yadda

Global methods much better than local methods...
Local vs. Global Illumination

Local Illumination

Global illumination
+ RT post-process

Pure global illumination (view independent)

Radiosity??

It is the name of a measure of light energy...
...and an algorithm:
- **Radiosity** = flux (energy flow per unit time) per unit area radiated from a surface.
- These are wavelength-dependent quantities.
  - R,G,B channels
**Radiosity — Set-up**

Models lighting in scenes for (almost exclusively) diffusely reflecting surfaces.

View independent!!

Assume polygonal scene
- polygons divided into $n$ small ‘patches’
- patch $i$
  - has area $A_i$
  - has radiosity $B_i$
  - emits energy $A_i E_i$
  - has surface reflectance $\rho_i$ (diffuse reflection)

---

**Just a Tad of Physics...**

Energy (i) = Emitted Energy (i) + Reflected Energy (i)

\[
i = 1,2,\ldots,n
\]

Reflected Energy(i) = $\rho_i$ * (Total Energy in from Environment)

Total Energy In (i) = Sum of proportions of energy from all other objects that reach i

\[
\sum_{j=1}^{n} A_j B_j F_{ji}
\]

$F_{ji}$ = proportion of energy from j that reaches i per unit area of j (form-factor)
Form Factors

Kinda tough to compute..

\[ F_{ij} = F_{A_iA_j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \cos \phi_i \cos \phi_j \frac{dA_j dA_i}{\pi r^2} \]

Remarks:
- involves visibility computations
- \( A_i F_{ij} = A_j F_{ji} \)
- sum of all \( F_{ij} \) is 1
  - closed environment — no escape for light

And the Result is...

\[ A_iB_i = A_iE_i + \rho_i \sum_{j=1}^{n} A_j B_j F_{ji} \quad i = 1, 2, ..., n \]

but since
\[ A_iF_{ij} = A_j F_{ji} \]

\[ B_i = E_i + \rho_i \sum_{j=1}^{n} B_j F_{ij} \]

Radiosity Equation
Solution?

The $B_i$ are unknown
Then can be rewritten as system of $n$ linear equations with $n$ unknowns.
- solution = equilibrium

Hence patches can be rendered
- ideally with smooth shading.

One set of eqns for each wavelength!
- usually, just RGB

Matrix System

\[
\begin{bmatrix}
1 - \rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\
-\rho_2 F_{21} & 1 - \rho_2 F_{22} & \cdots & -\rho_2 F_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
-\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1 - \rho_n F_{nn}
\end{bmatrix}
\begin{bmatrix}
B_1 \\
B_2 \\
\vdots \\
B_n
\end{bmatrix}
= 
\begin{bmatrix}
E_1 \\
E_2 \\
\vdots \\
E_n
\end{bmatrix}
\]

- $E_i$: input illumination to the system
- $F_{ij}$: form factor, $F_{ii}=0$ for plane patch
- $\rho_i$: reflectivity
- $E_i$ and $\rho_i$ are wavelength dependent; e.g., (R,G,B)

Solution: $B_i$ — a single radiosity value for each patch
Problem: Scalability

Have been assuming the full matrix solution - impractical!

Suppose there are 10,000 polygons.
- Each divided into 10 patches (say).
- \( n = 100,000 \)
- \( n^2 = 10,000,000,000 \)
- Each form factor 4 bytes (at least)
- \( 40,000,000,000 \) bytes = 40Gb for the matrix.

One Solution: Progressive Radiosity

Solve equation column by column (shooting) or row by row (gathering) without ever storing the full matrix.
Examples of Shooting Strategy

Examples of Progressive Radiosity

1 bounce, 2 bounces, 4 bounces
**Other Idea: Refinement**

Initial discretization may be bad

---

**Refinement: Example**

Refine where there’s sudden change in radiosity...
Other Idea: Hierarchical Radiosity

Interactions need to be computed precisely only for close surfaces! (P. Haranhan)

Other Extensions

Mostly, post-processing to allow for arbitrary objects (diffusive AND specular)
First, radiosity to provide initial view-independent illumination
Then, do ray-tracing to “enhance” the solution
  - allows specular effects
  - even scattering (dust, smoke, etc...)
Some Classical Results

CSE 872 Fall 2011

Some Classical Results

CSE 872 Fall 2011
Some Classical Results

Implicit Visibility

[Dachsberger 07]
What's Next?

More recently, many other techniques have been proposed:
- Monte-Carlo ray tracing techniques
- Photon Mapping

Can handle almost any kind of objects (translucent, glossy, caustics, etc...)

CSE 872 Fall 2011