Introduction to Computer Graphics

Radiosity
Problems with diffuse lighting

All visible surfaces, white.

Eye

A Daylight Experiment, John Ferren
Problems with diffuse lighting
Direct lighting
Global lighting
Cornell box

Photography

Simulation

Goral, Torrance, Greenberg & Battaile
Modeling the Interaction of Light Between Diffuse Surfaces
SIGGRAPH '84
Cornell box

- Calibration and measurement allows comparisons between reality and simulation
The rendering equation

\[
L(x', \omega') = E(x', \omega') + \int \rho_x(\omega, \omega') L(x, \omega) G(x, x') V(x, x') \, dA
\]

\(L(x', \omega')\) is the radiance from point \(x'\) in direction of \(\omega'\)

Radiance is measured in \([\text{W}/(\text{m}^2 \cdot \text{sr})]\)

http://en.wikipedia.org/wiki/Radiance
The rendering equation

\[ L(x', \omega') = E(x', \omega') + \int \rho_{x'}(\omega, \omega') L(x, \omega) G(x, x') V(x, x') \, dA \]

\( E(x', \omega') \) is the emitted radiance: \( E \) is greater zero for light sources.
The rendering equation

$$L(x',\omega') = E(x',\omega') + \int_\rho_x(\omega,\omega')L(x,\omega)G(x,x')V(x,x') \, dA$$

Sum of contributions from all other scene elements to the radiance from point $x'$ in direction of $\omega'$
The rendering equation

\[ L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega')L(x,\omega)G(x,x')V(x,x') \, dA \]

For every \( x \), compute \( L(x, \omega) \), the radiance in point \( x \) in direction \( \omega \) (from \( x \) to \( x' \))
The rendering equation

\[ L(x', \omega') = E(x', \omega') + \int \rho_{x'}(\omega, \omega') L(x, \omega) G(x, x') V(x, x') \, dA \]

The contribution is scaled by \( \rho_{x'}(\omega, \omega') \) (the BRDF in \( x' \)).
The rendering equation

\[ L(x',\omega') = E(x',\omega') + \int \rho_x(\omega,\omega')L(x,\omega)G(x,x')V(x,x') \, dA \]

For every \( x \), determine \( V(x,x') \), the visibility from \( x \) relative to \( x' \):
1 if there is no occlusion in direction \( \omega \), 0 otherwise.
The rendering equation

\[ L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega')L(x,\omega)G(x,x')V(x,x') \, dA \]

For every \( x \), compute \( G(x, x') \), the geometry term w.r.t. \( x \) and \( x' \)
Which constellation leads to a large exchange of light and why?
The radiosity assumptions

- Surfaces are Lambertian (perfectly diffuse)
  - Reflection occurs in all directions
- The scene is split into small surface elements
- The radiosity $B_i$, is the total radiosity that comes from element $i$
- For each element, the radiosity is constant
The radiosity equation

\[ L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x') \, dA \]

Radiosity assumption:
Perfectly diffuse surfaces – no directional dependency

\[ B_{x'} = E_{x'} + \rho_{x'} \int B_x G(x,x') V(x,x') \]
The radiosity equation

- Continuous radiosity equation

  Reflection factor

  \[ B_{x'} = E_{x'} + \rho_{x'} \int G(x,x') V(x,x') B_x \]

  Form factor

- \( G \): geometry term
- \( V \): visibility term

- Properties
  - No analytical solution, even for simple scenes
The radiosity equation

- Discretize into elements with const. radiosity

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

- Properties
  - Iterative solution
  - Expensive geometry computations
The radiosity matrix

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

- n linear equations in n unknowns \( B_i \):

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

- The solution of this LSE results in \( B_i \), which are independent of viewer position and direction
The radiosity matrix

Iterative solution

- The radiosity of an element is replaced by the multiplication of a row with the current solution vector (Gathering) (= Gauss-Seidel iteration)

\[
\begin{bmatrix}
B_1 \\
B_2 \\
\vdots \\
B_i \\
\vdots \\
B_n \\
\end{bmatrix}
= 
\begin{bmatrix}
E_1 \\
E_2 \\
\vdots \\
E_i \\
\vdots \\
E_n \\
\end{bmatrix}
+ 
\begin{bmatrix}
\rho_1 F_{i1} & \rho_1 F_{i2} & \cdots & \rho_1 F_{in} \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
\end{bmatrix}
\begin{bmatrix}
B_1 \\
B_2 \\
\vdots \\
B_i \\
\vdots \\
B_n \\
\end{bmatrix}
\]
Rendering the radiosity solution

- $B_i$ are constant per Element
- How to map to graphics hardware?
  - Average radiosity-values for each vertex
  - Extrapolate for vertices on the boundary
Form factors

- $F_{ij} = \text{Part of radiance from } j \text{ that reaches } i$
- Influenced by:
  - Geometry (area, orientation, position)
  - Visibility (other elements of the scene)
Form factors

- $F_{ij} =$ Part of radiance from $j$ that reaches $i$

\[
F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} V_{ij} \, dA_j \, dA_i
\]
Form factors
Ray casting

- Create \( n \) rays between 2 elements
  - \( n \) typically between 4 and 32
  - Determine visibility
  - Integrate point-point form factors
- Determines form factors between elements
Form factors

- Nusselt analog: the form factor is equivalent to the part of the unit circle, which the projection of the element occupies on the unit sphere

\[ A_i F_{ij} = A_j F_{ji} \]
Form factors
Hemicube algorithm

- Place hemicube at element center
- Discretize the sides into pixels
- Project and rasterize other elements into cube
- Each hemicube pixel contains precomputed form factor
- Form factor for an element is the sum of contributions
- Visibility by depth buffer
Solving the radiosity equation

Geometry

Form factor computation

Iterative solution of the LSE

Radiosity solution

Visualization (Rendering)

Radiosity Image

Reflection properties

Geometry

Form factor computation

Iterative solution of the LSE

Radiosity solution

Visualization (Rendering)

Radiosity Image

Reflection properties

Geometry

Form factor computation

Iterative solution of the LSE

Radiosity solution

Visualization (Rendering)

Radiosity Image

< 10%

> 90%

O(n²) form factors

< 10%

~ 0%

~ 0%
Progressive refinement

- Idea: instead of collecting radiosity from all sources ("gathering"), rather distribute radiosity from brightest emitters ("shooting")
Progressive refinement

\[
\begin{bmatrix}
B_1 \\
B_2 \\
\vdots \\
B_n
\end{bmatrix}
+ \begin{bmatrix}
\cdots \\
\cdots \\
\cdots
\end{bmatrix}
= \begin{bmatrix}
\cdots \\
\cdots \\
\cdots
\end{bmatrix}
\]
Progressive refinement

- Each patch has remaining radiosity $\Delta B_i$
- Start with $B_i = E_i$ and $\Delta B_i = E_i$
- Distribute $\Delta B_i$ to the scene
- Reciprocity:

$$B_i = E_i + r_i \sum_{j=1}^{n} B_j F_{ij}, \text{ for all } i$$

$$A_j F_{ji} = A_i F_{ij}$$

$$B_i = E_i + r_i \sum_{j=1}^{n} B_j F_{ji} \frac{A_j}{A_i}$$
Progressive refinement

- After sending from patch \( j \), the radiosity of elements \( A_i \) is increased

\[
B_i = B_i + r_i \Delta B_j F_{ji} \frac{A_j}{A_i}, \quad i = 1..n
\]

- The nondistributed radiosity is also increased

\[
\Delta B_i = \Delta B_i + r_i \Delta B_j F_{ji} \frac{A_j}{A_i}, \quad i = 1..n
\]

- The set undistributed radiosity of \( j \) to zero

\[
\Delta B_i = 0
\]
Progressive refinement

Advantages

- Each iteration only requires form factors $F_{ij}$ for element $i$ w.r.t. all other patches.
- Good results after few iterations, resulting in significantly less overhead when compared to Gauss-Seidel iterations.
- Only requires storing a single column of the form factor matrix.
Progressive refinement

Without ambient term
Progressive refinement

With ambient term
Discretization into patches

- Image quality depends on the size of patches
  - Smaller patches – smaller error
- Patches should be adaptively subdivided where large gradients in radiosity are evident
  - Start with regular grid
  - Subdivide based on quality criterion
Discretization into patches
Photon Mapping
Jensen 95
Examples

Lightscape  http://www.lightscape.com

Andrew Nealen, Rutgers, 2010
Examples

Mental Ray