Global Illumination and Radiosity

CS535

Daniel G. Aliaga
Department of Computer Science
Purdue University
Recall: Lighting and Shading

• Light sources
  – Point light
    • Models an omnidirectional light source (e.g., a bulb)
  – Directional light
    • Models an omnidirectional light source at infinity
  – Spot light
    • Models a point light with direction

• Light model
  – Ambient light
  – Diffuse reflection
  – Specular reflection
Recall: Lighting and Shading

- Diffuse reflection
  - Lambertian model

\[ I_D = K_D (N \cdot L)I_L \]
Recall: Lighting and Shading

- Specular reflection
  - Phong model

\[ I_S = K_S (V \cdot R)^n I_L \]
Recall: Lighting and Shading

• Well....there is much more
For example...

- Reflection -> Bidirectional Reflectance Distribution Functions (BRDF)
- Diffuse, Specular -> Diffuse Interreflection, Specular Interreflection
- Color bleeding
- Transparency, Refraction
- Scattering
  - Subsurface scattering
  - Through participating media
- And more!
Illumination Models

• So far, you considered mostly local (direct) illumination
  – Light directly from light sources to surface
  – No shadows (actually is a global effect)

• Global (indirect) illumination: multiple bounces of light
  – Hard and soft shadows
  – Reflections/refractions (you kinda saw already)
  – Diffuse and specular interreflections
Welcome to Global Illumination

- *Direct illumination + indirect illumination*; e.g.
  - Direct = reflections, refractions, shadows, ...
  - Indirect = diffuse and specular inter-reflection, ...

direct illumination  
with global illumination  
only diffuse inter-reflection
Global Illumination

• *Direct illumination* + *indirect illumination*; e.g.
  – Direct = reflections, refractions, shadows, ...
  – Indirect = diffuse and specular inter-reflection, ...
Reflectance Equation

\[ L_r(x, \omega_r) = L_e(x, \omega_r) + L_i(x, \omega_i) f(x, \omega_i, \omega_r)(\omega_i \cdot n) \]

Reflected Light (Output Image)  Emission  Incident Light (from light source)  BRDF  Cosine of Incident angle

[Slides with help from Pat Hanrahan and Henrik Jensen]
Reflectance Equation

\[ L_r(x, \omega_r) = L_e(x, \omega_r) + \sum L_i(x, \omega_i) f(x, \omega_i, \omega_r)(\omega_i \cdot n) \]

- \( L_r(x, \omega_r) \): Reflected Light (Output Image)
- \( L_e(x, \omega_r) \): Emission
- \( L_i(x, \omega_i) \): Incident Light (from light source)
- \( f(x, \omega_i, \omega_r)(\omega_i \cdot n) \): BRDF (Cosine of Incident angle)
Reflectance Equation

\[
L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\Omega} L_i(x, \omega_i) f(x, \omega_i, \omega_r) \cos \theta_i d\omega_i
\]

- **Reflected Light** (Output Image)
- **Emission**
- **Incident Light** (from light source)
- **BRDF**
- **Cosine of Incident angle**
Reflectance Equation

\[ L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\Omega} L_i(x, \omega_i) f(x, \omega_i, \omega_r) \cos \theta_i d\omega_i \]
The Challenge

\[ L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\Omega} L_i(x, \omega_i) f(x, \omega_i, \omega_r) \cos \theta_i \, d\omega_i \]

- Computing reflectance equation requires knowing the incoming radiance from surfaces

- ...But determining incoming radiance requires knowing the reflected radiance from surfaces
Global Illumination

Surfaces (interreflection)

\[ L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\Omega} L_r(x', -\omega_i) f(x, \omega_i, \omega_r) \cos \theta_i d\omega_i \]

Reflected Light (Output Image)  Emission  Reflected Light (from prev surface)  BRDF  Cosine of Incident angle
Rendering Equation

Surfaces (interreflection)

\[
L_r(x, \omega_r) = L_e(x, \omega_r) + \int_\Omega L_r(x', -\omega_i) f(x, \omega_i, \omega_r) \cos \theta_i d\omega_i
\]

Reflected Light (Output Image)
Emission
Reflected Light
BRDF
Cosine of Incident angle

UNKNOWN
KNOWN
UNKNOWN
KNOWN
KNOWN
Rendering Equation (Kajiya 1986)

Figure 6. A sample image. All objects are neutral grey. Color on the objects is due to caustics from the green glass balls and color bleeding from the base polygon.
Rendering Equation as Integral Equation

\[ L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\Omega} L_r(x', -\omega_i) f(x, \omega_i, \omega_r) \cos \theta_i \, d\omega_i \]

Reflected Light (Output Image) Emission Reflected Light BRDF Cosine of Incident angle
UNKNOWN KNOWN UNKNOWN KNOWN KNOWN

Is a Fredholm Integral Equation of second kind [extensively studied numerically] with canonical form

\[ l(u) = e(u) + \int l(v) K(u, v) \, dv \]

Kernel of equation
Linear Operator Equation

\[ l(u) = e(u) + \int l(v) K(u, v) dv \]

Kernel of equation

\[ L = E + KL \]

which is effectively a simple matrix equation (or system of simultaneous linear equations) where

L, E are vectors,
K is the light transport matrix (more on this later!)
Solving the Rendering Equation
(=how to compute L?)

• In general, too hard for analytic solution
• But there are approximations and some nice observations...
Solving the Rendering Equation

(=how to compute L?)

\[ L = E + KL \]

\[ IL - KL = E \]

\[ (I - K)L = E \]

\[ L = (I - K)^{-1} E \]

(using Binomial Theorem)

\[ L = (I + K + K^2 + K^3 + ...) E \]

\[ L = E + KE + K^2E + K^3E + ... \]

where term n corresponds to n-th bounces of light
Ray Tracing

\[ L = E + KE + K^2 E + K^3 E + \ldots \]

- **Emission directly from light sources**
- **Direct Illumination on surfaces**
- **Global Illumination**
  - (One bounce indirect)
  - [Mirrors, Refraction]
- **(Two bounce indirect)**
  - [Caustics, etc…]
Ray Tracing

\[ L = E + KE + K^2E + K^3E + \ldots \]

- **Emission directly from light sources**
- **Direct Illumination on surfaces**
- **OpenGL Shading**
- **Global Illumination**
  - (One bounce indirect)
  - [Mirrors, Refraction]
- **(Two bounce indirect)**
  - [Caustics, etc…]
Successive Approximation

$L_e$

$K \circ L_e$

$K \circ K \circ L_e$

$K \circ K \circ K \circ L_e$

$L_e$

$L_e + K \circ L_e$

$L_e + \cdots K^2 \circ L_e$

$L_e + \cdots K^3 \circ L_e$
Global Illumination and Related Concepts

• Example based:
  – BRDFs

• Analytical:
  – Radiosity

• Making it faster:
  – Path tracing
  – Ambient occlusion
Radiosity

- Radiosity, inspired by ideas from heat transfer, is an application of a finite element method to solving the rendering equation for scenes with purely diffuse surfaces.

\[
L_o(x, \omega, \lambda, t) = L_e(x, \omega, \lambda, t) + \int_{\Omega} f_r(x, \omega', \omega, \lambda, t) L_i(x, \omega', \lambda, t)(-\omega' \cdot n) d\omega'
\]

(rendering equation)

[Radiosity slides heavily based on Dr. Mario Costa Sousa, Dept. of CS, U. Of Calgary]
Radiosity

- Calculating the overall light propagation within a scene, for short **global illumination** is a very difficult problem.

- With a standard ray tracing algorithm, this is a very time consuming task, since a huge number of rays have to be shot.
Radiosity

• For this reason, the radiosity method was invented.

• The main idea of the method is to store illumination values on the surfaces of the objects, as the light is propagated starting at the light sources.
Radiosity

- Equation: \( B_i \, dA_i = E_i \, dA_i + R_i \int_j B_j F_{ji} \, dA_j \)
- Diffuse Interreflection
Radiosity (Thermal Heat Transfer)

• The "radiosity" method has its basis in the field of thermal heat transfer.

• Heat transfer theory describes radiation as the transfer of energy from a surface when that surface has been thermally excited.
Radiosity (Computer Graphics)

• **Assumption #1**: surfaces are diffuse emitters and reflectors of energy, emitting and reflecting energy uniformly over their entire area.

• **Assumption #2**: an equilibrium solution can be reached; that all of the energy in an environment is accounted for, through absorption and reflection.

• Also **viewpoint independent**: the solution will be the same regardless of the viewpoint of the image.
The Radiosity Equation

• The "radiosity equation" describes the amount of energy which can be emitted from a surface, as the sum of the energy inherent in the surface (a light source, for example) and the energy which strikes the surface, being emitted from some other surface.

• The energy which leaves a surface (surface "j") and strikes another surface (surface "i") is attenuated by two factors:
  – the "form factor" between surfaces "i" and "j", which accounts for the physical relationship between the two surfaces
  – the reflectivity of surface "i“, which will absorb a certain percentage of light energy which strikes the surface.
The Radiosity Equation

\[ B_i = E_i + \rho_i \sum B_j F_{ij} \]

- **Radiosity of surface i**
- **Emissivity of surface i**
- **Reflectivity of surface i**
- **Radiosity of surface j**
- **Form Factor of surface j relative to surface i**

accounts for the physical relationship between the two surfaces

will absorb a certain percentage of light energy which strikes the surface
The Radiosity Equation

\[ B_i = E_i + \rho_i \sum B_j F_{ij} \]

- Energy emitted by surface \( i \)
- Surface \( i \)
- Surface \( j \)
The Radiosity Equation

\[ B_i = E_i + \rho_i \sum B_j F_{ij} \]

Energy reaching surface \( i \) from other surfaces
The Radiosity Equation

\[ B_i = E_i + \rho_i \sum B_j F_{ij} \]

- Energy reflected by surface i
- Surface j
- Surface i
Classic Radiosity Algorithm

- Mesh Surfaces into Elements

- Compute Form Factors Between Elements

- Solve Linear System for Radiosities

- Reconstruct and Display Solution
Classic Radiosity Algorithm

- Mesh Surfaces into Elements
- Compute Form Factors Between Elements
- Solve Linear System for Radiosities
- Reconstruct and Display Solution
The Form Factor:

The **fraction** of energy leaving one surface that reaches another surface

It is a purely geometric relationship, independent of viewpoint or surface attributes
Between differential areas, the form factor equals:

\[ F dA_i dA_j = \cos \theta_i \cos \theta_j \]

\[ = \frac{\pi |r|^2}{\text{differential area of surface } i, j} \]

angle between Normal$_i$ and $r$

angle between Normal$_j$ and $r$
Between differential areas, the form factor equals:

\[ F_{ij} \, dA_i \, dA_j = \frac{\cos \theta_i \cos \theta_j}{\pi \left| r \right|^2} \]

The overall form factor between i and j is found by integrating

\[ F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi \left| r \right|^2} \, dA_i \, dA_j \]
Form Factors in (More) Detail

\[ F_{ij} = \frac{1}{A_i} \int \int \frac{\cos \theta_i \cos \theta_j}{\pi |r|^2} dA_i dA_j \]

\[ F_{ij} = \frac{1}{A_i} \int \int \frac{\cos \theta_i \cos \theta_j}{\pi |r|^2} V_{ij} dA_i dA_j \]

where \( V_{ij} \) is the visibility (0 or 1)
Form Factors in (More) Detail

• Several ways to find form factors

• **Hemicube** was original method
  + Hardware acceleration
  + Gives $F_{dAiAj}$ for all $j$ in one pass
  - Aliasing

• **Area sampling** methods now preferred
  → Slower than hemicube but GPU-able
  → As accurate as desired since adaptive
Area Sampling

Subdivide $A_j$ into small pieces $dA_j$
For all $dA_j$
  cast ray $dA_j$ to determine $V_{ij}$
  if visible
  compute $F_{dA_ijdA_j}$
$$F_{dA_ijdA_j} = \frac{\cos \theta_i \cos \theta_j}{\pi r^2} V_{ij} dA_j$$
sum up
$$F_{dAiAj} += F_{dAijdA_j}$$

We have now $F_{dAiAj}$
Next

• We have the form factors
• How do we find the radiosity solution for the scene?
  – The "Full Matrix" Radiosity Algorithm
  – Gathering & Shooting
  – Progressive Radiosity
• Meshing
Next Step:
Learn ways of computing **form factors**

- Recall the Radiosity Equation:

\[ B_i = E_i + \rho_i \sum B_j F_{ij} \]

- The \( F_{ij} \) are the form factors

- Form factors independent of radiosities (depend only on scene geometry)
Radiosity Matrix

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

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

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

• The "full matrix" radiosity solution calculates the form factors between each pair of surfaces in the environment, then forms a series of simultaneous linear equations.

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

• This matrix equation is solved for the "B" values, which can be used as the final intensity (or color) value of each surface.
Radiosity Matrix

• This method produces a complete solution, at the substantial cost of
  – first calculating form factors between each pair of surfaces
  – and then the solution of the matrix equation.

• This leads to substantial costs not only in computation time but in storage.
Next

• We have the form factors
• How do we find the radiosity solution for the scene?
  – The "Full Matrix" Radiosity Algorithm
  – Gathering & Shooting
  – Progressive Radiosity
• Meshing
Solve \([F][B] = [E]\)

• **Direct methods: \(O(n^3)\)**
  - Gaussian elimination
    • Goral, Torrance, Greenberg, Battaile, 1984

• **Iterative methods: \(O(n^2)\)**

  *Energy conservation*
  \(\rightarrow \text{"diagonally dominant"} \rightarrow \text{iteration converges}\

  – Gauss-Seidel, Jacobi: Gathering
    • Nishita, Nakamae, 1985
    • Cohen, Greenberg, 1985

  – Southwell: Shooting
    • Cohen, Chen, Wallace, Greenberg, 1988
Gathering

- In a sense, the light leaving patch $i$ is determined by gathering in the light from the rest of the environment

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

$B_i$ due to $B_j = \rho_i B_j F_{ij}$
Gathering light through a hemi-cube allows one patch radiosity to be updated.

\[
B_i = E_i + \sum_{j=1}^{n} \left( \rho_i F_{ij} \right) B_j
\]
Gathering

Row of $F$ times $B$

Calculate one row of $F$ and discard
Successive Approximation

\[
\begin{align*}
L_e & \quad K \circ L_e & \quad K \circ K \circ L_e & \quad K \circ K \circ K \circ L_e \\
L_e & \quad L_e + K \circ L_e & \quad L_e + K^2 \circ L_e & \quad L_e + K^3 \circ L_e
\end{align*}
\]
Shooting

- **Shooting light** through a single hemi-cube allows the whole environment's radiosity values to be updated simultaneously.

\[
B_j = B_j + B_i \left( \rho_j E_{ji} \right)
\]

where \( F_{ji} = \frac{F_{ij} A_i}{A_j} \)
for (i=0; i<n; i++) {
    B[i] = dB[i] = Be[i];
    while (!converged) {
        set i st dB[i] is the largest;
        for (j=0; j<n; j++)
            if (i!=j) {
                db = rho[j]*F[j][i]*dB[i];
                dB[j] += db;
                B[j] += db;
            }
        dB[i]=0;
    }
}
Progressive Radiosity

(a) Traditional Gauss-Seidel iteration of 1, 2, 24 and 100.
(b) Progressive Refinement (PR) iteration of 1, 2, 24 and 100.

From Cohen, Chen, Wallace, Greenberg 1988
Next

• We have the form factors
• How do we find the radiosity solution for the scene?
  – The "Full Matrix" Radiosity Algorithm
  – Gathering & Shooting
  – Progressive Radiosity

• Meshing
Accuracy

Reference Solution

Table in room sequence from Cohen and Wallace

Uniform Mesh
Artifacts

A. Blocky shadows
B. Missing features
C. Mach bands
D. Inappropriate shading discontinuities
E. Unresolved discontinuities
Increasing Resolution
Adaptive Meshing
Some Radiosity Results
The Cornell Box

- This is the original Cornell box, as simulated by Cindy M. Goral, Kenneth E. Torrance, and Donald P. Greenberg for the 1984 paper *Modeling the interaction of Light Between Diffuse Surfaces*, Computer Graphics (SIGGRAPH '84 Proceedings), Vol. 18, No. 3, July 1984, pp. 213-222.

- Because form factors were computed analytically, no occluding objects were included inside the box.
The Cornell Box


- The hemi-cube allowed form factors to be calculated using scan conversion algorithms (which were available in hardware), and made it possible to calculate shadows from occluding objects.
Discontinuity Meshing

Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg created this image for the 1992 paper *Discontinuity Meshing for Accurate Radiosity*. It depicts a scene that represents a pathological case for traditional radiosity images, many small shadow casting details. Notice, in particular, the shadows cast by the windows, and the slats in the chair.
Opera Lighting

• This scene from *La Bohème* demonstrates the use of focused lighting and angular projection of predistorted images for the background.

• It was rendered by Julie O'B. Dorsey, Francois X. Sillion, and Donald P. Greenberg for the 1991 paper *Design and Simulation of Opera Lighting and Projection Effects*. 
Radiosity Factory

- These two images were rendered by Michael F. Cohen, Shenchang Eric Chen, John R. Wallace and Donald P. Greenberg for the 1988 paper *A Progressive Refinement Approach to Fast Radiosity Image Generation*.

- The factory model contains 30,000 patches, and was the most complex radiosity solution computed at that time.

- The radiosity solution took approximately 5 hours for 2,000 shots, and the image generation required 190 hours; each on a VAX8700.
Museum

• Most of the illumination that comes into this simulated museum arrives via the baffles on the ceiling.

• As the progressive radiosity solution executed, users could witness each of the baffles being illuminated from above, and then reflecting some of this light to the bottom of an adjacent baffle.

• A portion of this reflected light was eventually bounced down into the room.

• The image appeared on the proceedings cover of SIGGRAPH 1988.