18. Radiosity

Reading

Recommended:

- Watt, Chapter 10.

Optional:

- M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press. 1993.

Physically-based rendering

Basic optics

- Physics of light and color
- Geometrical optics
  - Ray "metaphor"
  - Reflection and transmission
- Radiative transfer
  - Measurement: radiometry and photometry
  - Transport theory and integral equations

Physically-based rendering, cont'd

Why physically-based?

- Insights into problem of rendering
- Illumination engineering
- Quest for realism
  - A "grand challenge" problem in graphics
  - The light holodeck...someday

Radiometry

So far we have considered bouncing of light around room using ray tracing.

What is the fundamental quantity that is bouncing around?

This quanitity is a function of some variables. What are they?

Radiance

In graphics (and illumination engineering), the measure of energy along a ray is called **radiance**.

Rougly speaking, radiance is the power per unit area per unit direction passing through a point $\mathbf{p}$ in a particular direction $\mathbf{u}$:

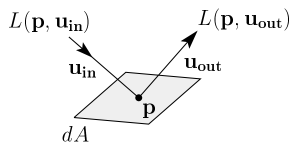
$$L(\mathbf{p}, \mathbf{u})$$

For the moment, we are ignoring wavelength dependence.

One of the most important properties of radiance: *radiance is constant along a ray travelling through empty space*.

The BRDF

Given an incoming direction and an outgoing direction, only a portion of the light is reflected depending on the surface material properties.



$L(\mathbf{p}, \mathbf{u_{in}})$
$L(\mathbf{p}, \mathbf{u_{out}})$
$\mathbf{u_{in}}$
$\mathbf{u_{out}}$
$\mathbf{p}$
dA

This reflection function is called the BRDF, or Bi-directional Reflectance Distribution Function.

$$f_r(\mathbf{p}, \mathbf{u_{in}} \to \mathbf{u_{out}}) \approx \frac{\mathbf{L}(\mathbf{p}, \mathbf{u_{out}})}{\mathbf{L}(\mathbf{p}, \mathbf{u_{in}})}$$

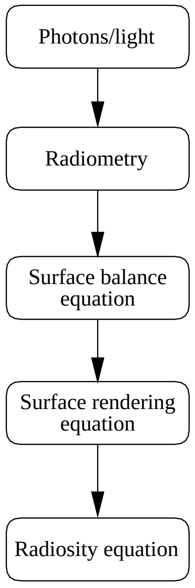
I use the $\approx$ sign here, because, strictly speaking we need to include a differential and a cosine. Let's stick with our intuitive definition.

The grand scheme

Physics
  Transport theory

Mathematics
  Integral equations

Computer Science
  Algorithms
  Numerics



Surface balance equation

We can decompose the radiance coming from a surface in terms of:

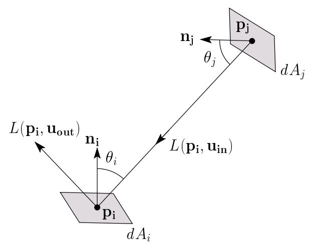
$$\begin{array}{ccccccc} \text{[outgoing]} & = & \text{[emitted]} & + & \text{[reflected]} & + & \text{[transmitted]} \\ L(\mathbf{p}, \mathbf{u_{out}}) & = & L_e(\mathbf{p}, \mathbf{u_{out}}) & + & L_r(\mathbf{p}, \mathbf{u_{out}}) & + & L_t(\mathbf{p}, \mathbf{u_{out}}) \end{array}$$

In our derivations, we will ignore transmission.

The surface rendering equation

Our objective is to solve for the radiance function, $L(\mathbf{p}, \mathbf{u_{out}})$ over all surfaces in all directions.

Given the geometric relationship between two patches:



The equation we are trying to solve is:

$$\begin{aligned} L(\mathbf{p_i}, \mathbf{u_{out}}) &= L_e(\mathbf{p_i}, \mathbf{u_{out}}) \\ &\quad + \int_M f_r(\mathbf{p_i}, \mathbf{u_{in}} \to \mathbf{u_{out}}) G(\mathbf{p_i}, \mathbf{p_j}) L(\mathbf{p_i}, \mathbf{u_{in}}) dA_j \end{aligned}$$

Where M refers to all surfaces in the scene and $G(\mathbf{p_i}, \mathbf{p_j})$ describes how efficiently light is transported from surface j to surface i. This equation is known as **the surface rendering equation**.

Radiosity

Solving the full blown rendering equation is *very hard*. What happens when we have only diffuse surfaces in a scene?

Light is reflected equally in all directions, so:

$$f_r(\mathbf{p}, \mathbf{u_{in}} \to \mathbf{u_{out}}) = \mathbf{f_{r,d}}(\mathbf{p})$$

In addition, the radiance from each point is equal in all directions:

$$L(\mathbf{p}, \mathbf{u_{out}}) = L_d(\mathbf{p})$$

After dropping angular dependencies, the rendering equation becomes:

$$L_d(\mathbf{p_i}) = L_e(\mathbf{p_i}) + \int_M f_{r,d}(\mathbf{p_i}) G(\mathbf{p_i}, \mathbf{p_j}) L_o(\mathbf{p_j}) dA_j$$

Thus, we now need to solve for a function $L_d(\mathbf{p})$ over 2D (surfaces) only.

Radiosity and reflectance

We could just go ahead and work on solving the new rendering equation, but we must obey some conventions first...

Radiosity, $B(\mathbf{p})$, is the power per unit area leaving a surface.

For a diffuse surface, it is related to the radiance by:

$$B(\mathbf{p}) = \pi L_d(\mathbf{p})$$

Reflectance, $\rho(\mathbf{p})$, is the ratio of the power per unit area leaving a surface to the power per unit area striking a surface.

Reflectance varies from 0 to 1 and is related to the BRDF of a diffuse surface by:

$$\rho(\mathbf{p}) = \pi f_{r,d}(\mathbf{p})$$

The radiosity equation

After making some subsitutions, we arrive at:

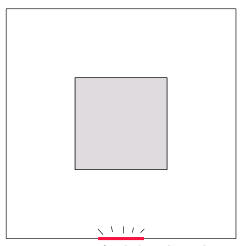
$$\begin{aligned} B(\mathbf{p_i}) &= E(\mathbf{p_i}) + \int_M \rho(\mathbf{p_i}) \frac{G(\mathbf{p_i}, \mathbf{p_j})}{\pi} B(\mathbf{p_j}) dA_j \\ B(\mathbf{p_i}) &= E(\mathbf{p_i}) + \rho(\mathbf{p_i}) \int_M F(\mathbf{p_i}, \mathbf{p_j}) B(\mathbf{p_j}) dA_j \end{aligned}$$

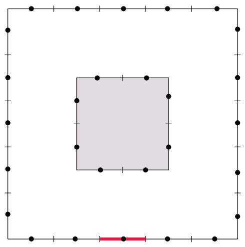
The discrete form of this equation is:

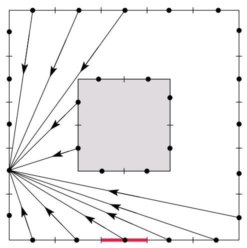
$$B_i = E_i + \rho_i \sum_j F_{ij} B_j$$

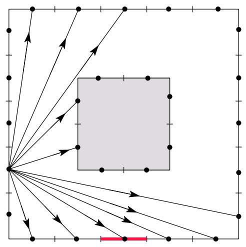
Intuition for the radiosity equation

To solve for the discrete radiosity function, we will break a scene into patches and solve a light transport equation.


A room containinfg a light and one object

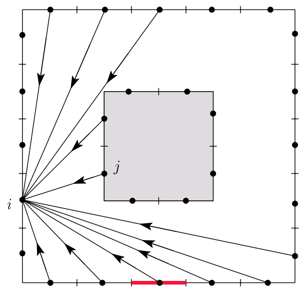

Break up the room into "patches"


Each patch receives light from other patches...


...and reflects light back to these patches.

Intuition for the radiosity equation, cont'd

Consider the light impinging on one patch:



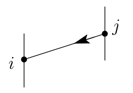
We can compute the reflected radiosity, B_i, by reflecting and summing the contributions from every other patch:

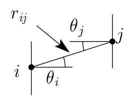
$$B_i = E_i + \rho_i \sum_j F_{ij} B_j$$

The F_{ij} are called the **form factors** and tells us how radiosity from one patch is transported to another. But what form does it take?

The form factor

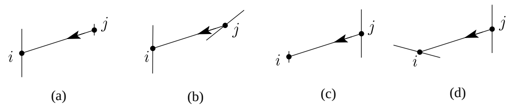
Consider the transport of energy from patch j to i:

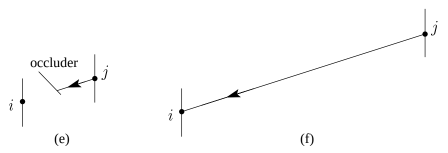

Transport path



Geometrical notation

Now let's vary the size, orientation, visibility, and distance between the patches:





How does the transport vary in each of (a)-(f)?

The form factor, cont'd

Let's put all these terms together:

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

Matrix form of the radiosity equation

Here again is the radiosity equation:

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

We can re-write this as:

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

Now we can write a matrix equation and solve!

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdot & \cdot & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & \cdot & & & -\rho_2 F_{2n} \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ -\rho_{n-1} F_{n-1,1} & & & & -\rho_{n-1} F_{n-1,n} \\ -\rho_n F_{n1} & \cdot & \cdot & -\rho_n F_{n,n-1} & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ \cdot \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \cdot \\ \cdot \\ \cdot \\ E_n \end{bmatrix}$$

Matrix form of the radiosity equation, cont'd

If we define:

$$K = \begin{bmatrix} \rho_1 F_{11} & \cdot & \cdot & \rho_1 F_{1n} \\ \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ \rho_n F_{n1} & \cdot & \cdot & \rho_n F_{nn} \end{bmatrix}$$

Then we can re-write the equation as:

$$(I - K)B = E$$

One way to solve this is to compute $B = (I - K)^{-1}E$ directly.

This is really expensive. The matrix, $(I - K)$ is generally fairly dense ($O(n^2)$ interactions), and inversion requires $O(n^3)$ operations.

Solving the radiosity equation

Iterative methods will generally get you close to the solution in fewer steps.

One possibility is to do a von Neumann expansion:

$$(I - K)^{-1} = I + K + K^2 + K^3 + ...$$

so that we could compute a truncated series:

$$B = E + KE + K^2E + K^3E + ...$$

Solving the radiosity equation, cont'd

We can write this as iterative matrix multiplication:

$$\begin{aligned} B^{(0)} &= E \\ B^{(1)} &= B^{(0)} + KB^{(0)} \\ B^{(2)} &= B^{(1)} + KB^{(1)} \\ B^{(3)} &= B^{(2)} + KB^{(2)} \end{aligned}$$

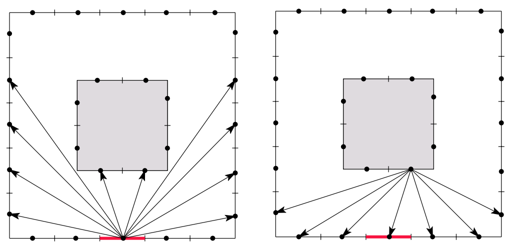
Note that each multiplication by K corresponds to bouncing the radiosity through the scene one more time.

This still converges fairly slowly and runs in $O(n^2)$ assuming you can stop after a number of iterations much less than n.

Solving the radiosity equation, cont'd

Whenever we multiply a row of K by B, we can think of this as gathering energy into a patch.

A faster converging method is based on the notion of "shooting" energy from patches.



Solving the radiosity equation, cont'd

The idea is:

1. Sort the patches by the amount of radiosity they currently have.
2. Shoot the energy from the brightest patch to all the other patches.
3. Mark this patch as having zero "unshot" radiosity.
4. Choose the next patch with the largest unshot radiosity and iterate.

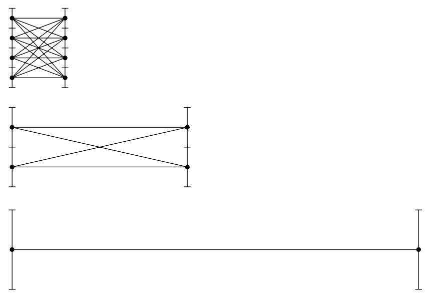
This approach is called "progressive radiosity".

The convergence is substantially faster, closer to $O(n)$ in practice.

Solving the radiosity equation, cont'd

Another idea is to compute the interactions *hierarchically*.

If two patches are close together, then link them in the usual way. If they are further apart, then link their parent patches.



The complexity can be shown to be linear in n, the number of leaf node patches.

However, the original input patches are the parent patches, so if there are k input patches, there are at least $O(k^2)$ interactions. Total complexity is then $O(n + k^2)$.

Final rendering

So, what can you do with the solution when you're done?

Once we've solved for the radiosity, we can ray trace the scene from any viewpoint without bouncing any rays. The results of all the bounces have been pre-computed.

We call this a "view indedpendent solution."

Final rendering, cont'd

Better yet, we can do an interactive walkthrough:

1. Given the radiosity at the center of patches (e.g., triangles or rectangles), estimate the radiosity at the vertices.
2. Use graphics hardware and Gouraud shading and do an interactive walkthrough.

[Show video.]

Final rendering, cont'd

We can also use hybrid methods. Use:

- radiosity methods (great for diffuse interreflections, but no modeling of specularity)
- and ray tracing (lousy for diffuse interreflections, but great for specularity)

to get some of the most realistic renderings ever produced!