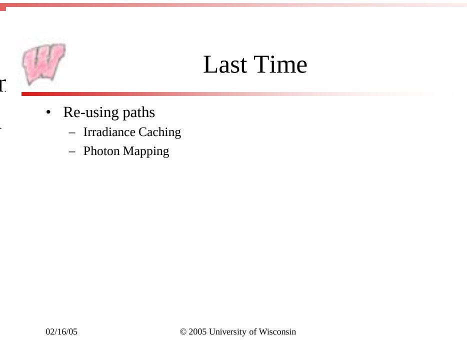
Last Time

- Re-using
  - Irradiar
  - Photon



Last Time

- Re-using paths
  - Irradiance Caching
  - Photon Mapping

Today

- Radiosity
  - A very important method in practice, because it is so much more efficient than Monte Carlo for diffuse environments
  - Can also be used in conjunction with Monte Carlo, if you're very careful about partitioning the LTE into different components

Radiosity

- **Radiosity is the total power leaving a surface, per unit area on the surface**
  - Usually denoted *B*
  - The outgoing version of irradiance
- To get it, integrate radiance over the hemisphere of outgoing directions:

$$B(\mathbf{x}) = \frac{d\Phi}{dx} = \int_{H(\mathbf{n})^2} L(\mathbf{x}, \omega_o) \cos\theta d\omega_o$$

Exitance

- Light sources emit light, they are sources of radiance
- Exitance is the equivalent of radiosity for emitters:

$$E(\mathbf{x}) = \int_{H(\mathbf{n})^2} L_e(\mathbf{x}, \omega_o) \cos\theta d\omega_o$$

- Distinguish exitance from radiosity to simplify equations
- Different from Intensity, which is power per unit solid angle
- Exitance is not ill-defined for point light sources

Radiosity Algorithms

- Radiosity algorithms solve the global illumination equation under a restrictive set of assumptions
  - All surfaces are perfectly diffuse
  - We only care about the radiosity at surfaces
    - Some form of rendering pass is required to transfer to the image plane
  - Surfaces can be broken into patches with constant radiosity
    - Some algorithms extend this to linear combinations of basis functions
- These assumptions allow us to linearize the global illumination equation

Diffuse Surface Radiosity

- Diffuse surfaces, by definition, have outgoing radiance that does not depend on direction

$$B(\mathbf{x}) = \int_{H(\mathbf{n})^2} L(\mathbf{x}, \omega_o) \cos\theta d\omega_o = \pi L_o(\mathbf{x})$$

- Same can be said for diffuse emitters

$$E(\mathbf{x}) = \int_{H(\mathbf{n})^2} L_e(\mathbf{x}, \omega_o) \cos\theta d\omega_o = \pi L_e(\mathbf{x})$$

- And recall the definition of the diffuse BRDF in terms of directional hemispheric reflectance

$$f(\mathbf{x}) = \frac{\rho_{hd}}{\pi}$$

Radiosity Light Transport

- Simplifying the global illumination equation gives:

$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_\Omega f(\mathbf{x}, \omega_o, \omega) L(\mathbf{x}, \omega) \cos\theta d\omega$$

$$\pi L(\mathbf{x}) = \pi L_e(\mathbf{x}) + \pi \int_\Omega \frac{\rho_{hd}(\mathbf{x})}{\pi} L(\mathbf{x}, \omega) \cos\theta d\omega$$

$$B(\mathbf{x}) = E(\mathbf{x}) + \rho_{hd}(\mathbf{x}) \int_\Omega L(\mathbf{x}, \omega) \cos\theta d\omega$$

- We have removed almost all the angular dependence, but we still have an integral of directions computing irradiance

Switch the Domain

- We can convert the integral over the hemisphere of solid angles into one over all the surfaces in a scene:

$$d\omega = \frac{\cos\theta' dy}{r^2}$$

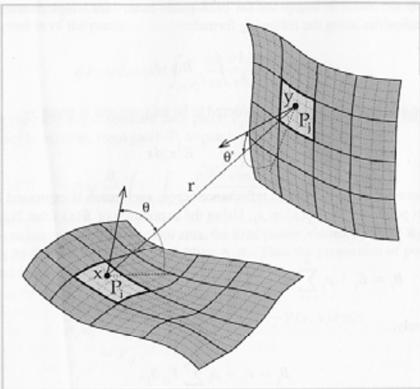
$$V(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if x and y are mutually visible} \\ 0 & \text{otherwise} \end{cases}$$

$$B(\mathbf{x}) = E(\mathbf{x}) + \rho_{hd}(\mathbf{x}) \int_{\mathbf{y} \in S} B(\mathbf{y}) \frac{\cos\theta \cos\theta'}{\pi r^2} V(\mathbf{x}, \mathbf{y}) dA$$

Discretize Radiosity

- Assume world is broken into N disjoint patches, P_i, $i=1..N$, each with area A_i
- Assume radiosity is constant over patches
- Define: $B_i = \frac{1}{A_i}\int_{x \in P_i} B(x)dx$

$$E_i = \frac{1}{A_i}\int_{x \in P_i} E(x)dx$$



Discrete Formulation

- Change the integral over surfaces to a sum over patches:

$$B(\mathbf{x}) = E(\mathbf{x}) + \rho_{hd}(\mathbf{x})\sum_{j=1}^{N}\int_{\mathbf{y}\in P_j} B(\mathbf{y})\frac{\cos\theta\cos\theta'}{\pi r^2}V(\mathbf{x},\mathbf{y})d\mathbf{y}$$

$$\frac{1}{A_i}\int_{\mathbf{x}\in P_i} B(\mathbf{x})d\mathbf{x} = \int_{\mathbf{x}\in P_i}\left[E(\mathbf{x}) + \rho_{hd}(\mathbf{x})\sum_{j=1}^{N}\int_{\mathbf{y}\in P_j} B(\mathbf{y})\frac{\cos\theta\cos\theta'}{\pi r^2}V(\mathbf{x},\mathbf{y})d\mathbf{y}\right]d\mathbf{x}$$

$$B_i = E_i + \rho_i\sum_{j=1}^{N} B_j\frac{1}{A_i}\int_{\mathbf{x}\in P_i}\int_{\mathbf{y}\in P_j}\frac{\cos\theta\cos\theta'}{\pi r^2}V(\mathbf{x},\mathbf{y})d\mathbf{y}d\mathbf{x}$$

The Form Factor

$$F_{ij} = \frac{1}{A_i}\int_{x\in P_i}\int_{y\in P_j}\frac{\cos\theta\cos\theta'}{\pi r^2}V(x,y)dydx$$

F_{ij} is the proportion of the total power leaving patch P_i that is received by patch P_j

- Note that we use it the other way: the form factor F_{ij} is used in computing the energy *arriving* at *I*
- Also called the configuration factor

Form Factor Properties

- Depends only on geometry
- Reciprocity: $A_iF_{ij}=A_jF_{ji}$
- Additivity: $F_{i(j\cup k)}=F_{ij}+F_{ik}$
- Reverse additivity is not true
- Sum to unity (all the power leaving patch i must get somewhere):

$$\forall i, \sum_{j=1}^{N} F_{ij} = 1$$

The Discrete Radiosity Equation

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

- This is a linear equation!

$$\mathbf{B} = \mathbf{E} + \mathbf{F}\mathbf{B}$$

$$\mathbf{E} = \mathbf{M}\mathbf{B} \quad \text{where} \quad \mathbf{M} = (\mathbf{I} - \mathbf{F})$$

- Dimension of *M* is given by the number of patches in the scene: *N*x*N*
  - It's a big system
  - But the matrix **M** has some special properties

Solving for Radiosity

- First compute all the form factors
  - These are view-independent, so for many views this need only be done once
  - Many ways to compute form factors
- Compute the matrix **M**
- Solve the linear system
  - A range of methods exist
- Render the result using Gourand shading, or some other method – but no additional lighting, it's baked in
  - Each patch's diffuse intensity is given by its radiosity

Solving the Linear System

- The matrix is very large – iterative methods are preferred
- Start by expressing each radiance in terms of the others:

$$\sum_{j=1}^{N} M_{ij} B_j = E_i, \quad 1 \le i < N, \quad M_{ij} = \delta_{ij} - \rho_i F_{ij}$$

$$B_i = -\sum_{\substack{j=1 \\ j \ne i}}^{N} \frac{M_{ij}}{M_{ii}} B_j + \frac{E_i}{M_{ii}}, \quad 1 \le i < N$$

Relaxation Methods

- Jacobi relaxation: Start with a guess for B_i, then (at iteration *m*):

$$B_i^{(m)} = -\sum_{\substack{j=1 \\ j \ne i}}^{N} \frac{M_{ij}}{M_{ii}} B_j^{(m-1)} + \frac{E_i}{M_{ii}}, \quad 1 \le i < N$$

- Gauss-Siedel relaxation: Use values already computed in this iteration:

$$B_i^{(m)} = -\sum_{j=1}^{i-1} \frac{M_{ij}}{M_{ii}} B_j^{(m)} - \sum_{j=i+1}^{N} \frac{M_{ij}}{M_{ii}} B_j^{(m-1)} + \frac{E_i}{M_{ii}}, \quad 1 \le i < N$$

Gauss-Seidel Relaxation

- Allows updating in place
- Requires **strictly diagonally dominant:**

$$|M_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{N} |M_{ij}|, \quad 1 \leq i < N$$

- It can be shown that the matrix **M** is diagonally dominant
  - Follows from the properties of form factors

Displaying the Results

- Color is handled by discretizing wavelength and solving each channel separately
- Smooth shading:
  - Patch radiosities are mapped to vertex colors by averaging the radiosities of the patches incident upon the vertex
  - Per-vertex colors then used to Gourand shade

Value for Computation

- Most of the time is spent computing form factors - must solve N visibility problems
- However, same form factors for different illumination conditions, and no color dependence
- Result is view independent - have radiosities for all patches. May be good or may be wasteful

Form Factors

- Computing form factors means solving an integral

$$F_{i,j} = \frac{1}{A_i} \int_{x \in P_i} \int_{y \in P_j} \frac{\cos\theta \cos\theta'}{\pi r^2} V(x,y)\,dy\,dx$$

- We have had plenty of practice at this kind of thing
- Also a point-patch form: the proportion of the power from a differential area about point *x* received by *j*

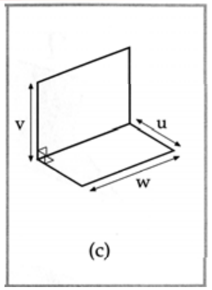
$$F_{x,j} = \int_{y \in P_j} \frac{\cos\theta \cos\theta'}{\pi r^2} V(x,y)\,dy$$

Form Factor Computations

- Unoccluded patches:
  - Direct integration
  - Conversion to contour integration
  - Form factor algebra – set operations on areas correspond to numerical operations on form factors – not really useful
- Occluded patches:
  - Monte Carlo integration
  - Projection methods (essentially numerical quadrature)
    - Hemisphere
    - Hemicube

Direct Integration – e.g. Rect-Rect

- Note that we can do this only under the constant radiosity over patch assumption
- There is a formula for 2 isolated polygons, but it assumes they can see each other fully!



$$F_{\perp} = \frac{1}{\pi X}\left\{ X\tan^{-1}\left(\frac{1}{X}\right) + Y\tan^{-1}\left(\frac{1}{Y}\right) - \sqrt{X^2+Y^2}\tan^{-1}\left(\frac{1}{\sqrt{X^2+Y^2}}\right)\right\}$$

$$+\frac{1}{4\pi X}\left\{\ln\left[\frac{(1+X^2)(1+Y^2)}{1+X^2+Y^2}\right] + X^2\ln\left[\frac{X^2(1+X^2+Y^2)}{(1+X^2)(X^2+Y^2)}\right] + Y^2\ln\left[\frac{Y^2(1+X^2+Y^2)}{(1+Y^2)(X^2+Y^2)}\right]\right\}$$

Contour Integral

- Use Stokes' theorem to convert the area integrals into contour integrals

$$F_{ij} = \frac{1}{2\pi A_i}\oint_{C_i}\oint_{C_j}\ln r\, d\vec{x}\cdot d\vec{y}$$

- For point to polygon form factors, the contour integral is not too hard
- Care must be taken when $r \rightarrow 0$

Projection Methods

- For patches that are far apart compared to their areas, the inner integral in the form factor doesn't vary much
  - That is, the form factor is similar from most points on a surface *i*
- So, compute point to patch form factors and weigh by area

$$F_{x,P} = \int_{y\in P}\frac{\cos\theta\cos\theta'}{\pi r^2}V(x,y)dy$$

Nusselt’s Analogy

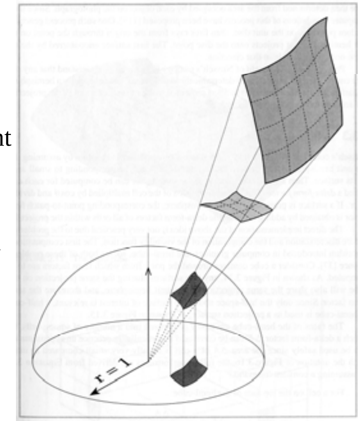
- Integrate over visible solid angle instead of visible patch area:

$$F_{x,P} = \frac{1}{\pi} \int_{\Omega_P} \cos\theta d\omega$$

$F_{x,P}$ is the fraction of the area of the unit disc in the base plane obtained by projecting the surface patch P onto the unit sphere centered at x and then orthogonally down onto the base plane.

Same Projection – Same Form Factor

- Any patches with the same projection onto the hemisphere have the same form factor
  - Makes sense: put yourself at the point and look out – if you see equal amounts, they get equal power
- It doesn’t matter what you project onto: two patches that project the same have the same form factor

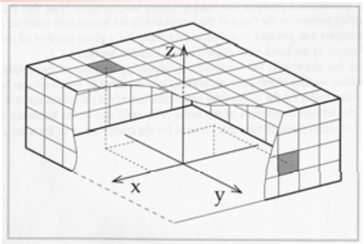


Monte-Carlo Form Factors

- We can use Monte-Carlo methods to estimate the integral over visible solid angle
- Simplest method – cosine weighted sampling:
  - Sample the disc about the point
  - Project up onto the hemisphere, then cast a ray out from the point in that direction
  - Form factor for each patch is the weighted sum of the number of rays that hit the patch
- There are even better Monte-Carlo methods that we will see later

The Hemicube

- We have algorithms, and even hardware, for projecting onto planar surfaces
- The hemicube consists of 5 such faces
- A “pixel” on the cube has a certain projection, and hence a certain form factor
- Something that projects onto the pixel has the same form factor



$$\Delta F_{top-face} = \frac{\Delta A}{\pi(1+x^2+y^2)^2}$$

$$\Delta F_{side-face} = \frac{z\Delta A}{\pi(1+z^2+y^2)^2}$$

Hemicube, cont.

$$F_{x,P_j} = \sum_{q \in C(j)} \Delta F_q$$

- Pretend each face of the hemicube is a screen, and project the world onto it
- Code each polygon with a color, and count the pixels of each color to determine $C(j)$
- Quality depends on hemicube resolution and z-buffer depth

Next Time

- Progressive Radiosity