Classical Radiosity Method
- Divide surfaces into patches
- Model light transfer between patches as system of linear equations
- Important assumptions (so far):
  - Reflection and emission are diffuse
  - No participating media (no fog)
  - No transmission (only opaque surfaces)
  - Radiosity is constant across each patch
  - Solve for R, G, B separately

Radiosity Equation
- For each patch i:
  \[ B_i = E_i + \rho_i \sum_j (F_{ij}A_j/A_i)B_j \]
- Variables
  - \( B_i \) = radiosity (unknown)
  - \( E_i \) = emittance of light sources (given)
  - \( \rho_i \) = reflectance (given)
  - \( F_{ij} \) = form factor from i to j (computed)
  - fraction of light emitted from patch i arriving at patch j
  - \( A_i \) = area of patch i (computed)

Form Factors via Hemicubes

Idealized Radiosity Computation
- Division into patches
- Scene
- Geometry
- Reflectance Properties
- Form factor calculation
- Solution of radiosity eqn
- Radiosity
- Image
- Visualization
- Viewing Conditions

Outline
- Substructuring
- Progressive Refinement
- Bidirectional Reflectance Distribution Function
- Combining Radiosity and Ray Tracing
Substructuring
- Radiosity assumed constant across patch
  - Impact of number of patches
    - Few: fast, but very inaccurate (blocky)
    - Many: slow O(n^2), but much more accurate
- Substructuring
  - Introduce elements as a substructure for patches
  - Use adaptively where radiosity varies rapidly
  - Distinguish elements and patches to avoid explosion

Elements vs. Patches
- Analyse transport from patch onto elements
- Do not analyze element-to-element detail
  - This means
    - Compute form factors from elements to patches
    - Do not compute form factors from patches to elements
    - Use weighted patch to parent-of-element
    - Complexity O(m \cdot n) for m elements, n patches
  - Typically substructured areas
    - Near lights
    - Shadow boundaries

Outline
- Substructuring
- Progressive Refinement
- Bidirectional Reflectance Distribution Function
- Combining Radiosity and Ray Tracing

Matrix Radiosity Revisited
- Compute all form factors \( F_{ij} \)
- Make initial approximation to radiosity
  - Emitting elements \( B_i = E_i \)
  - Other elements \( B_i = 0 \)
- Apply equation to get next approximation
  \[ B_i' = E_i + \rho_i \sum_j F_{ij} B_j \]
- Iterate with new approximation
  - Intuitively
    - Gather incoming light for each element \( i \)
    - Base new estimate on previous estimate

Progressive Refinement
- Shoot light instead of gathering light
- Basic algorithm
  - Initialize emitting element with \( B_i = E_i \)
  - Initialize others with \( B_i = 0 \)
  - Pick source \( i \) (start with brightest)
  - Using hemicube around source, calculate \( F_{ij} \)
  - For each \( j \neq i \), approximate \( B_j' = \rho_j B_i F_{ij} (A_i / A_j) \)
  - Pick next source \( i \) and iterate until convergence
- Each iteration is \( O(n) \)
- May or may not keep \( F_{ij} \) after each iteration

Progressive Refinement Corrected
- Problem: double-count if source is used more than once as source
- Solution: compute and use difference from last time a patch was used as a source \( (\Delta B_i) \)
  - Initialize \( \Delta B_i, B_i = E_i \)
  - Pick source \( i \) with maximum unshot power
  - Using hemicube, calculate \( F_{ij} \) for each \( j \)
    - \( \Delta R = \rho_i \Delta B_i F_{ij} (A_i / A_j) \)
    - \( B_i = B_i + \Delta R \)
    - \( \Delta B_i = \Delta B_i + \Delta R \)
    - \( \Delta B_i = 0 \)
Some Special Cases

- Image after we have iterated through all light sources?
  - Shadows, but no interreflections
- Can incrementally display image while iterating
  - Add ambient light at each stage for visibility
  - Ambient shading if progressively refined
- Incremental form factor computation

Radiosity Algorithms Summary

- Matrix radiosity algorithm
  - Pre-compute all form factors
  - Iterative solution (Gauss-Seidel)
    - Start with emission
    - Each object gathers light from all other objects
- Progressive refinement
  - Pick brightest patch
  - Compute outgoing form factors
  - Shoot light from this patch to all other patches
  - Repeat for next brightest batch
- Combine substructuring and progressive refinement

Outline

- Substructuring
- Progressive Refinement
- Bidirectional Reflectance Distribution Function
- Combining Radiosity and Ray Tracing

Bidirectional Reflectance Distribution

- General model of light reflection
- Hemispherical function
  - 6-dimensional (location, 4 angles, wavelength)
  \[ f(\omega_i \rightarrow \omega_r) = \frac{L_r(\omega_i)}{L_i(\omega_i) \cos\theta_i d\omega_i} \]

BRDF Examples

- Measure BRDFs for different materials

BRDF Isotropy

- Rotation invariance of BRDF
- Reduces 4 angles to 2
- Holds for a wide variety of surfaces
- Anisotropic materials
  - Brushed metal
  - Others?
- How many parameters for
  - Ideal specular?
  - Ideal diffuse?
Outline

- Substructuring
- Progressive Refinement
- Bidirectional Reflectance Distribution Function
- Combining Radiosity and Ray Tracing

Light Transport and Global Illumination

- Diffuse to diffuse
- Diffuse to specular
- Specular to diffuse
- Specular to specular
- Ray tracing (viewer dependent)
  - Light to diffuse
  - Specular to specular
- Radiosity (viewer independent)
  - Diffuse to diffuse
- Inherent limitations

Specular Radiosity

- Diffuse radiosity
  - Light reflected equally in all directions
  - Relationship between patches limited to form factor
- Specular radiosity
  - Retain viewer independence (unlike ray tracing)
  - Light reflected differently in different directions
  - For each source and each direction, need to calculate interaction
  - Not practical

Two-Pass Approach

- View-dependent specular is tractable
- View-independent diffuse is tractable
- First pass view independent
  - Enhanced radiosity
- Second pass is view dependent
  - Enhanced ray tracing

Pass 1: Enhanced Radiosity

- Diffuse transmission (translucent surfaces)
  - Backwards diffuse form factor
- Specular transmission
  - Extended form factor computation
  - Consider occluding translucent surfaces
  - Window form factor
- Specular reflection
  - Create "virtual" (mirror-image) environment
  - Use specular transmission technique
  - Mirror form factor

Pass 1 Result

- Account only for one specular reflection between surfaces (diffuse-specular-diffuse)
- Accurate diffuse component
- Solve enhanced radiosity equation as before
- Viewer independent solution
Pass 2: Enhanced Ray Tracing

- Classical ray tracing
  - Specular to specular light transport
- For diffuse-to-specular transport:
  - Should integrate incoming light over hemisphere
  - Approximate by using small frustum in direction of ideal reflection
  - Use radiosity of pixels calculated in Pass 1
  - Apply recursively if visible surface is specular

Two-Pass Global Illumination

- Still several approximating assumptions
- Appropriate for scenes with few specular reflecting or transmitting surfaces
- More expensive than already expensive methods

Two-Pass Radiosity Example

Summary

- Substructuring
- Progressive Refinement
- Bidirectional Reflectance Distribution Function
- Combining Radiosity and Ray Tracing

Preview

- Tuesday: Guest Lecture by Steve Sullivan
  - http://www.evenhouse.com/sullivan/
  - Works at Industrial Light + Magic (ILM)
    http://www.ilm.com/
- Next Lecture: Image Processing (tentative)