Hierarchical Radiosity with Multiresolution Meshes

Andrew Willmott
Thesis Proposal

Thesis Committee:
Paul Heckbert (CMU)
David O’Hallaron (CMU)
Andy Witkin (CMU)
Francois Sillion (INRIA)
Overview

• State of the art radiosity methods can’t handle scenes composed of complex objects well
  – Complex means 100,000-1,000,000 triangles.
  – Memory consumption is critical

• I propose to make radiosity useable for such scenes through the use of surface simplification.
Roadmap

• What’s Radiosity anyway?
• Problems with Existing Methods
• Motivation
• Approach
• Current Results
• Plans and Schedule
Global Illumination

- **Local illumination**
  - Light source to surface to eye, nothing but that.

- **Global illumination**
  - Consider secondary bounces of light
  - Reflections (sharp / soft), refractions...
Illumination Methods

• **Ray-tracing**
  - Cast rays from eye out into the scene.
  - Best at specular, can be adapted for diffuse surfaces
  - Point light sources

• **Radiosity**
  - Best at diffuse, can be adapted for glossy surfaces
  - Area light sources
Radiosity: Que?

- Discretise scene into elements, calculate transfer coefficients between elements
- Solve system of linear equations for radiosity
- Linearly interpolate the result for display
The Radiosity Method

• Definitions
  - Radiosity: diffuse light radiated by an area (Watts m\(^{-2}\)). Irradiance: incoming radiosity
  - Have \(k\) input polygons, decimate into \(n\) elements

• Solving the system
  - Initially used standard matrix techniques (Jacobi, Gauss-Seidel)
  - Solver iterates until solution converges
  - **But** this is \(O(n^2)\) in time and space
Progressive Radiosity

- Little used Southwell relaxation technique
- Track unshot radiosity in scene
- Repeatedly shoot element with most unshot radiosity: can see results improving
- $O(n^2)$ speed, $O(n)$ memory
Hierarchical Radiosity

- Use adaptive, hierarchical mesh (quadtree)
- Distant surfaces: use coarse level of quadtree, close surfaces: use fine level
- $O(k^2 + n)$ time and space complexity
- $k^2$ is a problem for $k > 1000$ polygons.
Example Hierarchy

Hierarchical Radiosity

Flatland

• polygon/element
□ cluster
•• link
Clustering

- **Cluster groups of polygons together into volumes**
  - Use these to unify separate hierarchies
  - $O(k \log k + n)$ time, $O(k + n)$ space complexity
  - Makes $>100,000$ polygon scenes practical

- **However...**
  - Must correct for projected area of cluster in direction of link: $O(k \log k)$ process
  - Must **touch all input geometry** on each iteration
Example Cluster Hierarchy

Hierarchical Radiosity with Clustering

Flatland

polygon/element
cluster
link
Current State of the Art

• **Research**
  - Hierarchical/wavelet radiosity systems

• **High-end: Lightscape, Lightworks**
  - Progressive radiosity, 1,000-100,000 polygon scenes
  - Raytracing post-pass to add specular component, 2-3 hour renders is fine.

• **Virtual worlds (read: games)**
  - Progressive radiosity, 10,000 polygon scenes
  - Quick previews, 10 minute final renders.
Lightscape Rendering
Alternative Diffuse Methods

• **Use raytracing, cache diffuse samples**
  - RADIANCE [Ward], Photon maps [J ensen]
  - As fast as or faster than existing, progressive radiosity methods
  - Hierarchical methods should be faster, but often are not, because of memory consumption.
  - If radiosity can’t match RADIANCE soon, perhaps it’s best forgotten as a general purpose technique.
Problems with Existing Radiosity Methods

• **Speed**
  - Above 10,000 polygons, progressive gets very slow due to $k^2$ performance.
  - Hierarchical better, but higher memory use means it’s still impractical for large scenes
  - Large scenes in research: 200,000 polygons; 1.5 hours, 170Mb.

• **Quality**
  - Not satisfactory! Shadows cause problems
  - Discontinuity meshing can help
My Motivation

• **Want to apply radiosity methods to industrial-strength scenes**
  - Models are 100,000 -> 1,000,000 polygons. Scenes have many models, texture maps, bump maps
  - Most rendering done on 64Mb->128Mb workstations
  - Render times must be minutes, not hours!

• **Hierarchical Radiosity has promise**
  - links are useful for recalculating shadows
Problem: Complex Models

204,000 triangle model. Medium resolution version!
Problem: Poor Meshing
My Approach

• **Use multiresolution models**
  - Avoid correcting for projected area: No $k \log k$
  - Much better locality during simulation; no longer touch all input polygons on each iteration
  - Makes possible sublinear performance in $k$

• **Use directional refinement (not quadtree)**
  - Adapt to shadow discontinuities better
  - Avoid explicitly locating discontinuities
Original Goal

• Somehow perform radiosity on simplified version of original model
• Most models have large, smooth regions which can be approximated well
• Only use detailed geometry when necessary

108,000 triangles, 707s

1000 triangles, 7s
Multiresolution Hierarchy

Hierarchical Radiosity with Clustering

Multiresolution Radiosity

- polygon/element
- cluster
- link

NOT USED
Multiresolution Meshes

• **Algorithm**
  - Start with original model
  - Progressively simplify with edge collapses

• **Output**
  - Log of simplification operations
  - Can be written as a binary tree of vertices
  - Cuts across this tree give models of various resolutions
Multiresolution Meshes

- Use Garland's Technique (Quadrics)
  - Fast, has the properties we want

2320 polygons  500 polygons  110 polygons
A New Hierarchy

OLD: Hierarchical Radiosity with Clustering

NEW: Multiresolution Radiosity

Simplify

Refine
Above the Input Polygons

Volume Clusters

Simplify

Input polygons

Face Clusters
A Real Volume Hierarchy

7 levels deep
5800 polygon cow
Below the Input Polygons

Input polygons

Refine

Regular, quadtree refinement

Edge-split refinement
But Wait!

- Traditional approach to radiosity transfer leads to faceted appearance.
- Scalar transfer of radiosity ill-suited to clusters of directionally independent polys.
- Luckily: Vector radiosity to the rescue.

108,000 triangles, 7s
Scalar Transfer

- Consider the following two face clusters

- $A_i$ all considered to have the same irradiance, $E_i$
Vector Transfer

- Use Vector approximations in transfer

- Irradiance of $A_i$ is now $E \cdot n_i$
Improved Dragon

Before

After

7s

8s
Initial Results

• Prototype of multires. radiosity system up and running
• Compared it to HRC implementation
• 200,000 polygon scene, 7 complex models
• Both methods share code where possible
## Current Results

<table>
<thead>
<tr>
<th>Parameter</th>
<th>HRC</th>
<th>MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Memory</td>
<td>31Mb</td>
<td>10Mb</td>
</tr>
<tr>
<td>Rays cast</td>
<td>2.1 million</td>
<td>1.7 million</td>
</tr>
<tr>
<td>Execution time</td>
<td>80 minutes</td>
<td>9 minutes</td>
</tr>
<tr>
<td>Links used</td>
<td>134860</td>
<td>117491</td>
</tr>
<tr>
<td>Volume clusters</td>
<td>22774</td>
<td>331</td>
</tr>
<tr>
<td>Face clusters/faces</td>
<td>199124</td>
<td>8800</td>
</tr>
</tbody>
</table>
Tasks

• Establish best face-cluster hierarchy to use
• Eliminate remaining shading discontinuities
• Find good link visibility representation
• Extend use of mesh past input polygons
• Establish sublinear performance empirically
## Schedule

<table>
<thead>
<tr>
<th>Activity</th>
<th>Months</th>
<th>Start Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find Best Clustering Method</td>
<td>1</td>
<td>April ‘98</td>
</tr>
<tr>
<td>Seam Elimination</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Visibility Scheme</td>
<td>2</td>
<td>August ‘98</td>
</tr>
<tr>
<td>Input-polygon Refinement</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Slack time/addition of features to system</td>
<td>2</td>
<td>January ‘99</td>
</tr>
<tr>
<td>Experiments and testing</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Writing Dissertation</td>
<td>4</td>
<td>June ‘99</td>
</tr>
<tr>
<td><strong>Total/Finish</strong></td>
<td><strong>18</strong></td>
<td><strong>October ‘99</strong></td>
</tr>
</tbody>
</table>
Usenet Posting:

It's not so much that Lightscape is too slow, but it really does not seem to like the fine meshes generated by Rhino on curvy surfaces. If you generate a sparser polygon object in Rhino it works better but the outlines of the curves get angly in Lightscape; if you get a fine enough mesh to smooth out all the curves, Lightscape (in my limited experience) leaves out lots of polygons when you raytrace even if you get it to import the polygons successfully to begin with.
Best Case, Worst Case?

• **Best Case**
  - Radiosity becomes practical with very large scenes
  - Animation houses start using it for soft-shadow illumination

• **Worst Case**
  - Surface refinement methods don’t prove to be beneficial
  - Can’t improve quality of the results enough