Announcements

- HW 3 out
  - Due next Friday
Soft Shadows

- Soft shadows appear natural
- Hard to get soft shadows in hardware
- Slow in software

Heckbert and Herf

- Use accumulation buffer
- Render shadows from multiple point lights over the area light (like MC)
- Accumulate shadows

| 2 x 2 samples | average | 16 x 16 samples |
Soler and Sillion

- Shadows as convolution

Penumbra Maps

- Wyman and Hansen
- Use shadow map and Haines technique for soft shadows on arbitrary surfaces
- Penumbra map
- Stores intensity of shadow
- Overall:
  - 3 pass: shadow map and penumbra map
  - Render image using depth from shadow map and intensity from penumbra map
Geometry-Based Soft Shadow Volume

- Assarsson and Moller
- Shadow volume approach

V-buffer stores visibility factor $[0,1]$
Computing Visibility: 2 passes

- Shadow volume quads are rendered into the V-buffer: overestimates umbra
- Penumbra wedges are rendered to compensate

Method Details: Wedge Example

512 x 512 at 5 FPS (software)
Visibility Passes

Pass 1: Render shadow volume quads

Pass 2: Compute visibility for each pixel inside the shadow wedges:
Point \( p = (x,y,z) \): find visibility of \( p \)

Precompute 4D coverage textures to accelerate visibility computation
Can handle textured lights, video textures

Assumptions

- Silhouettes are constant
- Overlapping objects
Results

Image of fire used as light source

© Kavita Bala, Computer Science, Cornell University

Results

Soft Shadow Volume 256 Samples 1024 Samples

Resolution: 512 x 512 @ 0.14 FPS
100 x 100 @ 3.00 FPS
256 x 256 @ 0.51 FPS

© Kavita Bala, Computer Science, Cornell University
Summary

• Hard shadows
  – Adaptive shadow maps
  – Edge-and-point rendering
  – Silhouette shadow maps
  – ...

• Soft shadows
  – Accumulation Buffer
  – Convolution
  – Penumbra Maps
  – Penumbra Wedges
  – ...
Many Lights

Motivation

- Most techniques work for single light source

- Many light sources
  - Treat it is a single integration domain
  - Importance sample lights
  - Importance sampling (with visibility) still hard problem

© Kavita Bala, Computer Science, Cornell University
Research on many lights

- Ward ‘91
- Shirley, Wang, Zimmerman ‘94
- Fernandez, Bala, Greenberg ‘02
- Wald and Slusallek ‘03
- Environment Map Sampling…

Ward ‘91

- Many lights in RADIANCE
- But all contributions not important

- Ignore some lighting at a point
  – User-defined cutoff: x%
- Sort lights according to potential contribution
  – Include G, cosine, L
  – EXCLUDE visibility
Ward ‘91

• Go through sorted list from the biggest potential contribution
  – Keep running count of visible contribution: V
  – Remainder of list (if fully visible) = R
  – Stop if R < x% of V

Ward ‘91

• But just can’t ignore remainder R

• Estimate remainder using visibility statistics from previous shadow tests: hack!

• Performance: 2x to 5x
• But, requires computing all potential contributions
  – Can be expensive for many lights
Shirley, Wang, Zimmerman ‘94

- Try to avoid linear cost of evaluating lights
- Separate lights into
  - Set of important lights (a small set)
  - Set of “dim” lights (large set)
- Construct pdf using:
  - all important lights
  - 1 out of all the dim lights
- Importance sample these lights

Shirley, Wang, Zimmerman ‘94

- Region of influence for important lights
  - Octree cells in region of influence have light in important set
- However, the partitioning into important and dim sets remains hard
- Also, still are not taking visibility into account
• Local Illumination Environment (LIE): lights and blockers that affect octree cell

Takes visibility into consideration!

• All lights/shadows are not visually important
• Weber’s law: 2% cutoff

\[ \text{Light 1} = \text{Light 2} = \text{Light 3} = \ldots \]
Using Masking

- Bright lights can mask out shadow details
- Weber's Law: variations in lighting are not visible if ambient lighting is bright enough
  - Conservative: 2% cutoff

- LIE: remove relatively dim lights (fully/partially visible)
  - Cheaper shading
  - Maximum light contribution < 2% of dimmest point in cell
  - Actually, cumulative maximum light contribution < 2% of dimmest point in cell

Using Masking

<table>
<thead>
<tr>
<th>Point 1</th>
<th>Light 1</th>
<th>Light 2</th>
<th>Light 3</th>
<th>Light 4</th>
<th>Light 5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1</td>
<td>16</td>
<td>30</td>
<td>20</td>
<td>66.1</td>
<td></td>
</tr>
<tr>
<td>Point 2</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>25</td>
<td>42</td>
</tr>
<tr>
<td>Point 3</td>
<td>0.1</td>
<td>0.1</td>
<td>9.8</td>
<td>12</td>
<td>18</td>
<td>40</td>
</tr>
<tr>
<td>Point 4</td>
<td>0.1</td>
<td>0.5</td>
<td>14</td>
<td>32</td>
<td>23</td>
<td>69.6</td>
</tr>
<tr>
<td>Max</td>
<td>0.1</td>
<td>0.5</td>
<td>17</td>
<td>32</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Sorted</td>
<td>0.1</td>
<td>0.5</td>
<td>17</td>
<td>25</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>
Fernandez, Bala, Greenberg ‘02

Without Masking  With Masking

Image

Cost

Cheap  Expensive

© Kavita Bala, Computer Science, Cornell University

Wald Slusallek ‘03

• PDF for sampling visible lights in interactive setting
  – Assume significant occlusion
  – Each room influenced by few lights
• 2-step algorithm (every frame)
  • 1st step: Determine important (unoccluded) lights by crude path tracing
  • 2nd step: Importance samples these lights
    – Completely ignore (probably) occluded lights

© Kavita Bala, Computer Science, Cornell University
Wald Slusallek ‘03

Rendering w/ Environment Maps

• High lighting complexity

• Rich: captures real world
Ambient Occlusion

• Interactive hardware rendering with many lights?

• Traditionally “fake” diffuse illumination using an ambient term

• But this just results in a constant addition

• Ambient occlusion adds some visibility to the fake diffuse illumination

Ambient Occlusion

• Pre-compute the ambient term

• At each vertex, shoot rays over hemisphere (cosine weighted)
  – MC sampling: sample hemisphere

• Does it hit a surface or escape? Compute average visibility \( V = 1 – \text{hits/samples} \)

• Ambient Out = Ambient In \( \times V \)
Problem

- Can move object around without deforming it

- But, slow!

- How to render interactively with many lights?
  - Open question