Ray Tracing

Tim Purcell
NVIDIA

Small Sampling of GI on GPUs

- Much more detail in the included papers
- Lots of other ‘global illumination on GPUs’ in the literature
  - The Ray Engine [Carr et al. 2002]
  - GPU Algorithms for Radiosity and Subsurface Scattering [Carr et al. 2003]
  - Radiosity on Graphics Hardware [Coombe et al. 2004]
  - Photon Mapping on Graphics Hardware [Purcell et al. 2003]
  - Lots and lots of shadow papers...

Radiosity

Radiosity on Graphics Hardware [Coombe et al. 2004]

Subsurface Scattering

GPU Algorithms for Radiosity and Subsurface Scattering [Carr et al. 2003]

Ray Tracing

Ray Tracing
Implementation Options

• GPU as a ray-triangle intersection engine [Carr et al. 2002]
  - Rays and geometry streamed to GPU
  - Intersection calculation results read back
  - Acceleration structure traversal done on host CPU

• GPU as a ray tracing engine [Purcell et al. 2002]
  - Scene geometry and acceleration structure stored on GPU
  - GPU performs ray generation, acceleration structure traversal, intersection, and shading
  - Host provides camera info

Techniques Used

• Data structure navigation
  - Texture memory stores data structures
  - Dependent texture fetches walk through data

• Flow control
  - Kernel binding based on occlusion query results
  - Efficient selective execution of kernels using early-z occlusion culling
  - Difficulty in flow control disappearing with newest graphics cards
    - PS 3.0

Efficient Selective Execution

• Rendering giant screen filling quad not ideal
  - Not all pixels need to process every rendering pass

• Proposed low-overhead early fragment kill
  - Computation mask
  - Controllable early-z occlusion culling

  • Trade computation for bandwidth

Streaming Ray Tracer

Camera
Generate Eye Rays

Grid
Traverse Acceleration Structure

Triangles
Intersect Triangles

Materials
Shade Hits and Generate Shading Rays

Texture Memory Organization

<table>
<thead>
<tr>
<th>Uniform Grid 3D Luminance Texture</th>
<th>vox0</th>
<th>vox1</th>
<th>vox2</th>
<th>vox3</th>
<th>vox4</th>
<th>vox5</th>
<th>voxM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>11</td>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td>564</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Triangle List 1D Luminance Texture</th>
<th>tri0</th>
<th>tri1</th>
<th>tri2</th>
<th>tri3</th>
<th>tri4</th>
<th>tri5</th>
<th>triN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>21</td>
<td>216</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Triangles 3x 1D RGB Textures</th>
<th>v0</th>
<th>v1</th>
<th>v2</th>
</tr>
</thead>
<tbody>
<tr>
<td>xyz</td>
<td>xyz</td>
<td>xyz</td>
<td>xyz</td>
</tr>
<tr>
<td>xyz</td>
<td>xyz</td>
<td>xyz</td>
<td>xyz</td>
</tr>
<tr>
<td>xyz</td>
<td>xyz</td>
<td>xyz</td>
<td>xyz</td>
</tr>
<tr>
<td>xyz</td>
<td>xyz</td>
<td>xyz</td>
<td>xyz</td>
</tr>
</tbody>
</table>

Original System Implementation

• ATI Radeon 9700 Pro (R300)
• ATI Fragment Program
Recent GPU Ray Tracers
- k-d tree based acceleration structure
- Grid-based open source ray tracers

Nonlinear Ray Tracing

Cornell Box - Ray Traced Shadows
Rendered using a Radeon 9700 Pro

Teapotahedron
Rendered using a Radeon 9700 Pro

Quake 3 - Ray Traced Shadows
Rendered using a Radeon 9700 Pro

Quake 3 - Ray Traced Shadows
Rendered using a Radeon 9700 Pro

[Foley and Sugerman 2005]
[Christen 2005]
[Karlsson and Ljungstedt 2004]
[Weiskopf 2004]
Nonlinear Ray Tracing

Simple Volume Ray Caster

Camera → Generate Eye Rays

Grid → Traverse Acceleration Structure

Accumulate Ray Opacity

Ray Cast Volume Rendering

Unstructured Meshes

- Convex meshes [Weiler et al. 2003]
- k-buffer [Callahan et al. 2005]