REAL TIME VOLUMETRIC SHADOWS USING POLYGONAL LIGHT VOLUMES

Markus Billeter
Erik Sintorn
Ulf Assarsson
Chalmers University of Technology
Overview

• Introduction: Airlight
• From Ray Marching to Light Volumes
• Light Volume Algorithm
• Adaptive Tessellation
• Demo / Movie
• Results
• Conclusion
Introduction:
Airlight / Participating Media
Direct Light and Hard Shadows Only
Airlight / Participating Media Only
Airlight

- Use single scattering model
  - Integral along ray
  - Nishita et al. in 1987
- Direct solution using texture lookups
  - Gives in-scattered light (Airlight) on line segment
  - E.g. Sun et al. in 2005
- Ignores occlusion
Ray-marching

• Ray march to find illuminated regions
  – At each point, determine if illuminated

• Done with
  – alpha blended planes
    • Dobashi et al. in 2002
    • Imagire et al. in 2007
  – Loop in fragment shader
    • E.g. Toth and Umenhoffer in 2009
Ray-marching, Improved

- Limit ray marched regions
  - Use shadow volumes to bound interesting region
  - Wyman et al. in 2008
- Cases where the bounds don’t help...
Light Volumes, Introduction

• Avoid ray marching entirely
  – Lit range on ray can be evaluated directly
  – Need to find ranges/boundaries

⇒ Shadow Volumes!
Light Volumes, Introduction

• Avoid ray marching entirely
  – Lit range on ray can be evaluated directly
  – Need to find ranges/boundaries
⇒ Shadow Volumes!
Light Volumes, Introduction

- Avoid ray marching entirely
  - Lit range on ray can be evaluated directly
  - Need to find ranges/boundaries

⇒ Shadow Volumes!
Light Volumes, Introduction

• Avoid ray marching entirely
  – Lit range on ray can be evaluated directly
  – Need to find ranges/boundaries
⇒ Shadow Volumes!
Light Volumes, Introduction

• Avoid ray marching entirely
  – Lit range on ray can be evaluated directly
  – Need to find ranges/ boundaries

⇒ Shadow Volumes!

• Problems:
  – Overlapping Volumes
Light Volumes

• Fix problems by:
  – Sort/depth peel volumes
    • Venceslas et. al. in 2006
    • James in 2003

• Reconstruct volume from shadow map!
  – Reconstruction explored by McCool in 2000
  – Use bounded light volume
Light Volumes

• Fix problems by:
  – Sort/depth peel volumes
    • Venceslas et. al. in 2006
    • James in 2003

• Reconstruct volume from shadow map!
  – Reconstruction explored by McCool in 2000
  – Use bounded light volume
Light Volumes

• Fix problems by:
  – Sort/depth peel volumes
    • Venceslas et. al. in 2006
    • James in 2003

• Reconstruct volume from shadow map!
  – Reconstruction explored by McCool in 2000
  – Use bounded light volume
Light Volumes

• Fix problems by:
  – Sort/depth peel volumes
    • Venceslas et. al. in 2006
    • James in 2003

• Reconstruct volume from shadow map!
  – Reconstruction explored by McCool in 2000
  – Use bounded light volume
Problems

• Shadow maps are inherently inexact
  – Sampling errors, quantization errors

• Very important: must not miss transitions!
  – Add and subtract large values
Problems

• Shadow maps are inherently inexact
  – Sampling errors, quantization errors

• Very important: must **not** miss transitions!
  – Add and subtract large values
Solution: Clamp Transitions

- Don’t use Z-culling of volumes
  - clamp to screen-space Z
- Example Figure:
  - Two identical contributions after clamp...
    ⇒ ...will cancel
  ⇒ Airlight behind visible objects will never contribute to the final image.
Adaptive Tessellation

• Light volume: potentially lots of geometry
  – @ $1024^2$ : ~2M triangles
  – Use adaptive tessellation to reduce this

• Tessellation performed with
  – Geometry shaders and transform feedback
Adaptive Tessellation

- Find edges in shadow map (Laplacian)
- Build edge map
  - Mip-map hierarchy
- Each texel contains:
  - Subdivision required?
  - Stitching (neighborhood) information
- Ensure max 1-level of difference between neighbors
Adaptive Tessellation

Input:

Level N (x,y) + edge map at level N @ (x,y)

If( subdivide ):
• Emit four smaller quads
• Repeat (Level N+1)

Else:
• Check neighbor-information
• Emit triangles
Adaptive Tessellation Performance

- Reduces triangles by
  - Sibenik: at least 50% for 1024^2 shadow maps
    - Average: 30% triangles compared to static tessellation
  - Sponza: at least 60% for 1024^2 shadow maps
    - Average: 30% triangles compared to static tessellation
  - Not an exhaustive test

<table>
<thead>
<tr>
<th></th>
<th>Laplacian</th>
<th>Edge Map</th>
<th>Tessellate</th>
<th>Render</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sponza2 1024^2 SM</td>
<td>0.66ms</td>
<td>0.68ms</td>
<td>3.11ms</td>
<td>7.51ms</td>
</tr>
<tr>
<td>Static Tessellation:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sponza2 4096^2 SM</td>
<td>10.11ms</td>
<td>9.14ms</td>
<td>4.13ms</td>
<td>17.75ms</td>
</tr>
<tr>
<td>Static Tessellation:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- NVIDIA GTX280
- Same as paper!
Demo / Movie

- Recorded on a NVIDIA GTX 480
  - Resolution is 1024x768 in all clips
  - Performance figures on next slide!
Scene: Sponza2

- 5 light sources
- $512^2$ shadow map
- 100 FPS on average
Demo / Movie

• Recorded on a NVIDIA GTX 480
  – Resolution is 1024x768 in all clips
  – Performance figures on next slide!

• Live Demo session
  – On this notebook (9400M)
## Results - Performance

<table>
<thead>
<tr>
<th></th>
<th>Shadow Maps</th>
<th>Adaptive Tessellation</th>
<th>Draw Volumes</th>
<th>Render Scene</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sponza2 5x 512^2</td>
<td>5.2ms</td>
<td>-</td>
<td>1.3ms</td>
<td>1.0ms</td>
<td>9.7ms</td>
</tr>
<tr>
<td>Sibenik 2x 1024^2</td>
<td>0.9ms</td>
<td>-</td>
<td>3.9ms</td>
<td>1.9ms</td>
<td>9.4ms</td>
</tr>
<tr>
<td>HPG 1x 1024^2</td>
<td>0.2ms</td>
<td>0.6ms</td>
<td>0.3ms</td>
<td>0.2ms</td>
<td>3.3ms</td>
</tr>
</tbody>
</table>

- **NVIDIA GTX480**
- **View resolution:** 1024x768
- **Constant overhead:** around 2ms
  - **Information, post-processing, composition**
Conclusions

- Only uses information from the shadow and depth map
  - Handles alpha masks, etc
  - Easy to add to existing programs (additional pass)
  - Shadow map size can be scaled

- However: does not handle textured lights
  - And only homogenous media
  ⇒ Better solutions to the Single Scattering Integral?