A Lazy Object-Space Shading Architecture With Decoupled Sampling

Christopher A. Burns
Kayvon Fatahalian†
William R. Mark

Intel Corporation
†Stanford University
I. Motivation
The Long-Term Objective

- High geometric detail
- Motion blur and depth-of-field
- Efficient support for these in a real-time system

*Ray traced by Gilles Tran

*Photo by “Austinii” @ http://austinii.deviantart.com
Two Possible Approaches...

<table>
<thead>
<tr>
<th></th>
<th>GPU</th>
<th>Reyes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Blur, DOF</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Detailed Geometry</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Adaptive Tessellation</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Fine Occlusion Culling</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
1. Start with GPU pipeline, add missing features

<table>
<thead>
<tr>
<th>Motion Blur, DOF</th>
<th>Detailed Geometry</th>
<th>Adaptive Tessellation</th>
<th>Fine Occlusion Culling</th>
<th>GPU</th>
<th>Reyes</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Ragan-Kelley ’10]</td>
<td>[Fatahalian ’10]</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Two Possible Approaches...

1. Start with GPU pipeline, add missing features
2. Start with Reyes pipeline, improve overall efficiency

<table>
<thead>
<tr>
<th></th>
<th>GPU</th>
<th>Reyes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Blur, DOF</td>
<td>[Ragan-Kelley '10]</td>
<td>X</td>
</tr>
<tr>
<td>Detailed Geometry</td>
<td>[Fatahalian '10]</td>
<td>X</td>
</tr>
<tr>
<td>Adaptive Tessellation</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Fine Occlusion Culling</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Instead, Modify the Reyes Pipeline

We evolve Reyes in two ways:

1. Decouple shading from triangle vertices (Decoupling)
2. Shade partial grids after rasterization (Lazy Shading)
Instead, Modify the Reyes Pipeline

We evolve Reyes in two ways:

1. Decouple shading from triangle vertices (Decoupling)
2. Shade partial grids after rasterization (Lazy Shading)

We achieve two things:

1. Eliminate the need for half-pixel polygons
2. Eliminate redundant shading via fine-grained occlusion culling
II. Reyes Overview
Reyes Pipeline Overview

- Split / Dice
- Surface Evaluation
- Cull
- Shade @ Vertices
- Rast
Reyes Pipeline Overview

Split / Dice → Surface Evaluation → Cull → Shade @ Vertices → Rast
Reyes Pipeline Overview

- **Split / Dice**
- **Surface Evaluation**
- **Cull**
- **Shade @ Vertices**
- **Rast**

**Split / Dice**

**Surface Evaluation**

**Cull**

**Shade @ Vertices**

**Rast**
Reyes Pipeline Overview

Split / Dice → Surface Evaluation → Cull → Shade @ Vertices → Rast
Reyes Pipeline Overview

- Split / Dice
- Surface Evaluation
- Cull
- Shade @ Vertices
- Rast

- uv only
- Surface Evaluation
- all vertex attributes
Reyes Pipeline Overview

- Split / Dice
- Surface Evaluation
- Cull
- Shade @ Vertices
- Rast
Reyes Pipeline Overview
Reyes Pipeline Overview
Reyes Pipeline Overview

Split / Dice → Surface Evaluation → Cull → Shade @ Vertices → Rast
Reyes Pipeline Overview

Split / Dice → Surface Evaluation → Cull → Shade @ Vertices → Rast
Reyes Pipeline Overview

- Split / Dice
- Surface Evaluation
- Cull
- Shade @ Vertices
- Rast
Reyes Pipeline Overview

Split / Dice -> Surface Evaluation -> Cull -> Shade -> Rast

[Diagram showing the pipeline process with shaded and non-shaded surfaces]
III. Decoupled Shading
Micropolygons are Expensive

Sample Test Efficiency v. Triangle Area

- 0.5 pixels/triangle
- 7.7 pixels/triangle
Decouple Shading From Vertices

- Reyes gives one dial: shading rate
- We give two: shading rate and tessellation rate
- Micropolygons unnecessary for good shading

Reyes Vertex Shading

Decoupled w/ Shading Grid
Decoupled Shading Results

Larger Triangles = Fewer Triangles

0.5 pixels per triangle

7 pixels per triangle
Decoupled Shading Results

Larger Triangles = Greater Sample Test Efficiency

<table>
<thead>
<tr>
<th>Scene</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Spheres”</td>
<td>3.1x</td>
</tr>
<tr>
<td>“Army”</td>
<td>3.2x</td>
</tr>
<tr>
<td>“Furball”</td>
<td>3.0x</td>
</tr>
</tbody>
</table>

0x  0.5x  1.0x  1.5x  2.0x  2.5x  3.0x  3.5x
Decoupled Shading Analysis

10x Fewer Triangles + 3x Improved Efficiency ≈ 4x Reduced Rast Costs

Derived from timing scalar implementation of UVT-interleave at 16 samples per pixel
Decoupled Shading Analysis

Rasterization Speedup vs. Triangle Size

- Interleave UV
- NoBlur

Marginal cost of adding blur support to rasterization decreases as triangle size increases.

Implication: Micropolygons make blur more expensive.
IV. Shading Post-Rasterization
If any part of a grid is visible, the entire grid is shaded.
If any part of a grid is visible, the entire grid is shaded.
Shading Lazily in Reyes

Typical Reyes

Split / Dice → Evaluate → Cull → Shade → Rast

Modified Reyes

Split / Dice → Evaluate → Cull → Rast → Shade

Frag Buffer
Shader Execution is Reduced

Traditional Reyes

Reyes w/ Post-Rasterization Shading

Shader Executions Per Pixel

Lazy Shading

Chris Burns
Tiling Worsens Overshading

No Tiling  Tiling  Tiling + Lazy

Lazy Shading
We Shade Less Post-Rasterization

Reduction in Shader Execution

<table>
<thead>
<tr>
<th></th>
<th>Untiled</th>
<th>Tiled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Furball</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Army</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spheres</td>
<td>1.0x</td>
<td>1.5x</td>
</tr>
<tr>
<td></td>
<td>2.0x</td>
<td>2.5x</td>
</tr>
<tr>
<td></td>
<td>3.0x</td>
<td>3.5x</td>
</tr>
<tr>
<td></td>
<td>4.0x</td>
<td></td>
</tr>
</tbody>
</table>

Lazy Shading

Chris Burns
Conclusion - Summary

1. Object-space shading can be done w/out micropolygons
   - Significantly reduces rasterization costs
   - Reduced marginal cost of stochastic rasterization

2. Redundant shading can be significantly reduced
   - Especially in a tiled renderer
Final Thoughts

- Micropolygons mostly unnecessary
  - Displaced geometry approx. by shading at non-silhouettes
  - Need fancy tessellator to optimize adaptive tessellation

- GPUs and Reyes may converge
  - We want best of both worlds
  - Much recent work in this direction [Fatahalian10, Ragan-Kelley10]