GPU-Accelerated High-Quality Hidden Surface Removal

Daniel Wexler, Larry Gritz, Eric Enderton, Jonathan Rice

Requirements

Comparable quality to CPU-only renderers:

- Depth of field and motion blur
- Transparency
- Filters with wide support
- Robustness for real production scenes
- Occlusion culling to avoid excess shading
- Spectral opacity and arbitrary outputs
- Complete feature set (shadows, GI, RT…)

Requirements

Using hardware to accelerate computations:

- Commodity hardware
- Hide all hardware limitations
- No loss in precision (fp32 everywhere)

→ Not the best mode for most GPUs!
Hider Architectural Overview

- REYES-style geometry processing
- Supersampling for anti-aliasing
- Accumulation buffer for MB & DOF
- Enhanced depth peeling for transparency
- Two-pass downsampling for filtering
- Occlusion query for culling

Prior Work

- Accumulation Buffer [Haeberli, Akeley '90]
- Depth Peeling [Mammen '89, Everitt '01]
- REYES [Cook, Carpenter, Catmull '87], [Apodaca, Gritz '99]
- Two-Pass Filtered Downsampling [Wexler, Enderton '05]

REYES Algorithm

[Diagram showing the REYES algorithm flow]
GPU Hiding Algorithm

- ALL OPAQUE
- ALL TRANSPARENT
- DEPTH PEEL
- MB/DOF
- ACCUMULATE
- FILTER
- OUTPUT

more passes done
more layers done

Grids, Pixels & Samples

Graphica Hardware 2005
Grids, Pixels & Samples

Transparency

- Opaque pre-processing
  - One additional texture-z test
  - Reduces number of depth peeling passes
  - Occlusion culling remove hidden surfaces
Z-Batches

- For N grids processed in batches of B grids:
  \( O((N/B)B^2) = O(BN) = O(N) \)
- Problem: grids overlap into multiple batches
- Opacity thresholding between batches

Transparency

Transparency Artifacts
Poor Performance Cases

Extensions

- Two-pass depth peel for average-z
- Volumetric shadow map generation
- Multiple camera (stereo) rendering
- Workqueue-based latency hiding
- Adaptive motion and DOF sampling

Challenges

- Hiding Latency
  - Occlusion Query
  - Orthogonal computations
- Hybrid Algorithms
  - Batch size vs. excessive computation
  - Starving and Readback
- Programming Environment
  - Debugging and profiling
  - Support and stability
Thank You

Q & A