AGGREGATE G-BUFFER ANTI-ALIASING

Cyril Crassin\textsuperscript{1}, Morgan McGuire\textsuperscript{1,2}, Kayvon Fatahalian\textsuperscript{3}, Aaron Lefohn\textsuperscript{1}

\textsuperscript{1}NVIDIA \hspace{1em} \textsuperscript{2}Williams College \hspace{1em} \textsuperscript{3}Carnegie Mellon University
Motivation
Overview

- High frequency shading is too costly
- Idea: Strong **decoupling** of shading rate
  - Shading statistical **geometry distributions**

### Diagram Description
- MSAA 32x on the left
- FXAA on the right
- Aggregate 0 andAggregate 1 visually represented
- Pixel frustum diagram with aggregate and shading attributes
- Per-sample visibility and shading attributes

**Overview Diagram Details**

- Pixel frustum diagram illustrating aggregate and shading attributes
- Comparing MSAA 32x and FXAA
- Shading statistical geometry distributions
- Decoupling of shading rate strategy
Single-sampled  MSAA 8x  AGAA (2A)
Deferred shading

Simple/Complex [Lauritzen 2010]
- Segmentation based on geometric complexity
- Shading amortization:
  - Across *same* planar surface
  - ...Or *almost* planar -> Quality reduction

- Simple scenes + large memory requirement

Credit: Crytek [Sousa 2013]
Surface Based Anti-Aliasing (SBAA) [Salvi 2012]

- High frequency visibility
  - Simplified geometry pre-pass
- Only store and shade the $n$ most contributing surfaces
- Non-shaded surface information discarded
  -> Aliasing in complex regions
AGAA overview

- Integrate **ALL** geometry samples
  - All surfaces + curvature

- Accumulate and **filter** geometry samples in pixel-space **before** shading
  - Per-sample **visibility** (Z-buffer)
  - Similar to texture/voxel-space pre-filtering
    - But *pixel-space* + *on-the-fly* aggregation
  - **Aggregate G-Buffer** statics:
    - Normal Distribution Function (*NDF*)
      - + sub-pixel position distribution.
    - Average Albedo, Metal coef., emissive, etc.
Rendering with Aggregates

- **NDF**: Mean albedo
- **Mean metal**
- **AG-Buffer**: Pixel 2x AA
- **Deferred Shading**
- **G-Buffer**: Generation
- **Depth Prepass**
- **Raster pass**
- **G-Buffer**: Early depth testing 8xAA
- **Pixel**: [8x Samples]
- **AG-Buffer**: NDF, Mean albedo, Mean metal
- **AG-Buffer**: 2xAA
- **Deferred Shading**: [8x Samples]
- **Framebuffer**: 8xAA
- **Depth**: Framebuffer
Rendering with Aggregates

- Depth Prepass
- Aggregate Definition
- AG-Buffer Generation
- Blend
- Deferred Shading

Framebuffer: 8xAA

- Depth
- Face normal
- [8x Samples]

Early depth testing

AG-Buffer: 2xAA

- NDF
- Mean albedo
- Mean metal
- [2x Aggregates]

Raster pass

[8x Samples]
Aggregate definition

- Clustering visibility samples:
  - Cross-primitives + Disjoint surfaces

- **Goal:** Minimize shading errors due to correlated attributes [Bruneton and Neyret 2012]
  - **Distance function:** 
    Orientation + distance-based clustering
  - Control over characteristic length of the scene
Deferred Shading

- Similar to shading from filtered texture maps
  - *AGAA is independent from the shading model*

- We used the Blinn-Phong BRDF model
  - Filtering **Specular** component using Toksvig [Toksvig 2005]
  - +Analytic approx. from Toksvig for **Lambertian diffuse** [Baker and Hill 2012]

- **Shadowing** must be filtered
Deferred 8x (reference)
AGAA (1A)

Memory: 41 MB
Shading events: 1/Pixel
AGAA

Shading events: 1.51 /Pixel

Memory: 71 MB
Deferred 8x (reference)

Memory: 112 MB

Shading events: 6.68 /Pixel (Simple/Complex)
Shading events: 1.51 /Pixel

Memory: 71 MB
Memory: 137 MB

Shading events: 3.32 /Pixel
Memory: 137 MB

Shading events: 3.32 /Pixel
Memory: 71 MB
Shading events: 1.51 /Pixel
Results: Performance

Deferred shading @8x MSAA  720p -  Comparison with Simple/Complex [Lauritzen 2010] - NVIDIA GTX980 (Maxwell GM204)

54% Faster rendering than Simple/Complex
(2.84x Faster shading)

<table>
<thead>
<tr>
<th>Scene</th>
<th>1. Prepass</th>
<th>2. Aggregate Def.</th>
<th>3. AG-Buffer Generation</th>
<th>4. Shading</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>OldCity</td>
<td>2.61</td>
<td>0.6</td>
<td>3.8</td>
<td>3.6</td>
<td>10.61</td>
</tr>
<tr>
<td>UE3 FoliageMap</td>
<td>2</td>
<td>0.6</td>
<td>2.47</td>
<td>3.67</td>
<td>8.74</td>
</tr>
</tbody>
</table>

Deferred Simple/Complex [Ref]

<table>
<thead>
<tr>
<th>Scene</th>
<th>G-Buffer Gen.</th>
<th>Simple/Complex</th>
<th>Shading</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>OldCity</td>
<td>5.73</td>
<td>0.41</td>
<td>10.24</td>
<td>16.38</td>
</tr>
<tr>
<td>UE3 FoliageMap</td>
<td>4.35</td>
<td>0.41</td>
<td>10.45</td>
<td>15.21</td>
</tr>
</tbody>
</table>
Results: Memory savings

- % Memory relative to super-sampled G-Buffer
  - 37% saving @8x - 2 Aggregates
  - 20% less than SBAA @2 Surfaces - ~40% less @Iso quality

AG-Buffer layout:
- [16B/Aggregate + 6B/Sample]

G-Buffer layout: [16B/sample]
Constraints and Limitations

Unified material with **unique** shading model
- No material switch (Skin, water, hair, cloth...)
- All shading inputs must be filterable!

NDF precision:
- A few very differently oriented surfaces
- But uni-modal Gaussian distrib.
  - Specular sparkling

Correlation issues:
- Lit green foliage + Shadowed red wall
Conclusion

- Path forward for very **high sampling rates** in real-time
  - Scalable: Bounded **storage** and **shading rate**

Properties:
- **Cross-primitives** + **Cross-surfaces** amortization
- All **geometric details** integrated

Remaining work on pre-filtering schemes for advanced unified shading/material models
THANK YOU!
Questions?
BACKUPS
Step 3: AG-Buffer generation

- **Rasterize @ sample frequency (Eg. 8x MSAA)**
  - Inside a set of color buffers @ Aggregate frequency (Eg. 2x)
  - Per-sample early depth testing

- **Fragment shader:**
  - Generates attributes for accum., weighted by visible coverage
  - Route output attributes into one of the aggregates

- **Additive blending:**
  - Additive blending inside AG-Buffer (weighted sum)

* [Maxwell or DX12 hardware specific features]

---

** diagram: **

- **Raster**: 8x MSAA
- **Depth Test**
- **Fragment Shader**: *Post-depth coverage*
- **Blend**: (ADDITIVE) *Output coverage (routing)*
- **AG-Buffer**: NDF, Mean albedo, Mean metal