HYDRA

Pixar’s Real-Time Render Engine for Feature Film Assets

Jeremy Cowles, GPU Team Lead

jcowles@pixar.com    @moldymagnet
The Rest of the Team:

David Dixon

Takahito Tejima

David Yu
Motivation

- Aging Presto render engine
- Existing “geometry cache” render engine
- New geometry cache: Universal Scene Description
- Unify GL preview in the studio
- AZDO & Vulkan

Today: Rasterization
Design Constraints

- Scenegraph agnostic
- Static, deforming, topologically varying
- Gracefully handle broken assets
- Assets optimized for beauty, not performance
- Subdivision surfaces and curves
- Transition: Fermi to Maxwell
Decouple:

- Scenegraph
- Drawing & Compute Dispatch
- Resource Management

Also see:

**Advanced Scenegraph Rendering** (GTC 2013)
Architecture Details

- Resource sharing across representations
- Thread friendly for Vulkan
- Fast visibility
Drawing Coordinate

Location of all associated surface/object attributes:

- **Constant:** per draw
- **Uniform:** per face
- **Vertex/Varying:** per vertex
- **Face Varying:** per vertex, per face
- **Instance:** per instance

RenderMan primitive variables, in GLSL.

Well defined tessellation!
Discovered Topology Instancing

- Data Fetch
  - Hash, Register
  - Share: Mem & Time

- Compute:
  - Must Resolve DAG

- Varying Topology: update in-place
Multi Draw Indirect

- **Motivation:**
  - Reduce driver overhead by reducing draw calls
  - Index buffer aggregation is horrible (old approach)

- **MDI Fine Print:**
  - Cache issues: invalidation is critical
  - Layout depends on render pass
  - You can mix in your own data!

Ok... does it actually work as well as aggregation?
## Blue Umbrella: City set

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>270,000 Meshes</td>
<td>Quadro 4000</td>
<td></td>
</tr>
<tr>
<td>29,000,000 Triangles</td>
<td>Hardware Max: 890M Tri/s</td>
<td></td>
</tr>
<tr>
<td>Old school aggregated indices:</td>
<td>17 FPS</td>
<td>493M Tri/s</td>
</tr>
<tr>
<td>Unaggregated draw calls:</td>
<td>5 FPS</td>
<td></td>
</tr>
<tr>
<td>Multi Draw Indirect:</td>
<td>16 FPS</td>
<td></td>
</tr>
<tr>
<td>+GPU Screen Presence Culling:</td>
<td>40 FPS +</td>
<td>1160M Tri/s*</td>
</tr>
</tbody>
</table>
GPU Instancing Data Layout

Prototype 1
InstanceID 0

Prototype 2
InstanceID 1

Instance Data

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translate</td>
<td>(X,Y,Z)</td>
<td>(X,Y,Z)</td>
<td>(X,Y,Z)</td>
<td>(X,Y,Z)</td>
</tr>
<tr>
<td>Rotate</td>
<td>(i, r0, r1, r2)</td>
<td>(i, r0, r1, r2)</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Scale</td>
<td>(X,Y,Z)</td>
<td>(X,Y,Z)</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Instance Offsets

<table>
<thead>
<tr>
<th>Offset</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
GPU Per-Instance Culling

- **Init:** Cull inst. count = N
  - Draw inst. count = 0

- **Cull** reorders instance offsets into a culled offset buffer

- **Draw** count doubles as an atomic head pointer to next available instance offset

![Diagram showing MDI and culled instance offsets](image-url)
GPU Instancing: Particulate

1.3 Million instances
109 Million faces per frame
Quadro K6000
10 FPS Orbit
60+ FPS Orbit w/GPU Culling
Instancing 5,000 Buzz Lightyears

- 5,000 instances of Buzz
- 72 frames of animation
- Streaming point cache
  - i.e. not rigged
  - no vertex skinning
- 40-55 FPS Playback
River Simulation Previs

- 800,000 Tris
- Topologically varying
- Vertex color per frame
- **Quadro 4000: 30 FPS**

Bottlenecks:

- Disk & GPU I/O
- Topology sharing is pure overhead
● Uniform vs. Adaptive considerations
● Two buckets for mesh topology (mesh & tags)
● GPU Compute for Animation
● 2.x and 3.x during transition

More Info:

http://graphics.pixar.com/opensubdiv
http://github.com/PixarAnimationStudios/OpenSubdiv
Universal Scene Description

- Time-sampled, like Alembic
- Layered scene data
- File format agnostic
- Thread-friendly

More Info:

http://graphics.pixar.com/usd
https://groups.google.com/forum/#!forum/.usd-interest

All content ©2015 Disney/Pixar, all rights reserved. "Universal Scene Description", “PIXAR”, “PRESTO” and the “PRESTO” logo are trademarks of Pixar Animation Studios. All other trademarks and logos are the property of their respective owners.
The GPU Team is Hiring!
(send your resume: jcowles@pixar.com)
Questions?

(Please complete the Presenter Evaluation sent to you by email or through the GTC Mobile App. Your feedback is important!)