GPU TECHNOLOGY CONFERENCE

HYDRA

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

Jeremy Cowles, GPU Team Lead

jcowles@pixar.com @moldymagnet

PIXAR ANIMATION STUDIOS

The Rest of the Team:

David Dixon

Takahito Tejima

David Yu

Motivation

GPU TECHNOLOGY CONFERENCE

- 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**


Toy Story 3

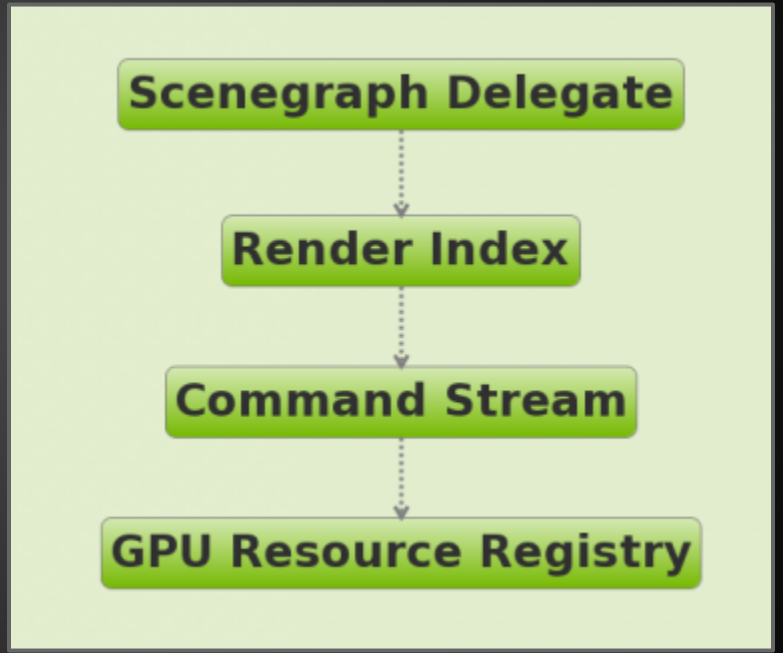
Design Constraints

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

Architecture Big Picture

Decouple:

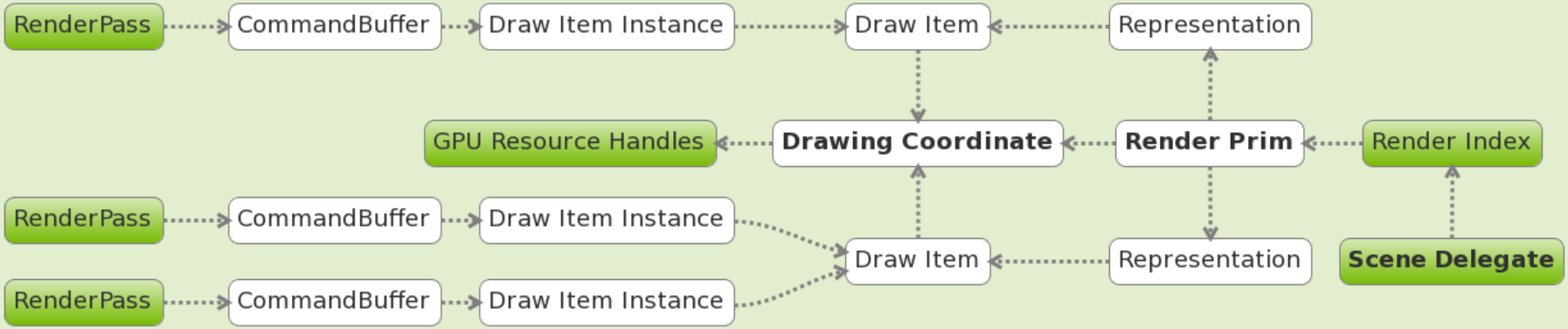
- Scenegraph
- Drawing & Compute Dispatch
- Resource Management



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:

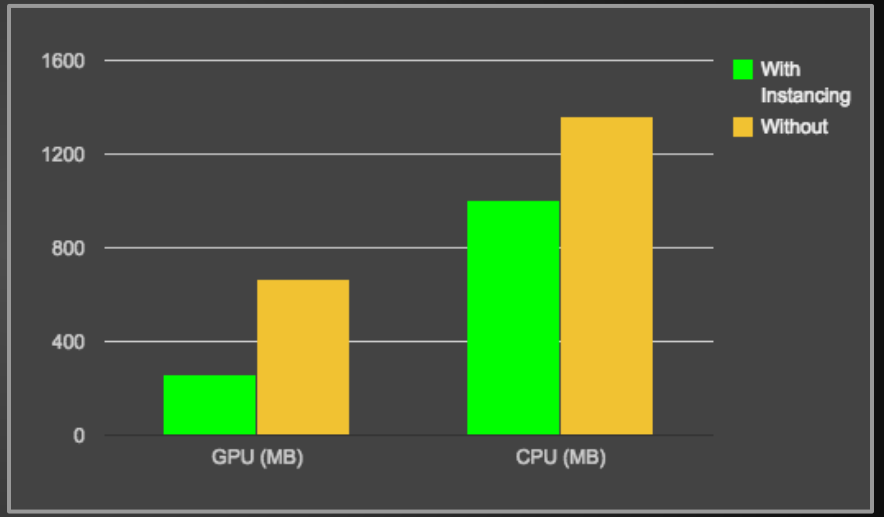
- Constant: per draw
- Uniform: per face
- Vertex/Varying: per vertex
- Face Varying: per vertex, per face
- Instance: per instance

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



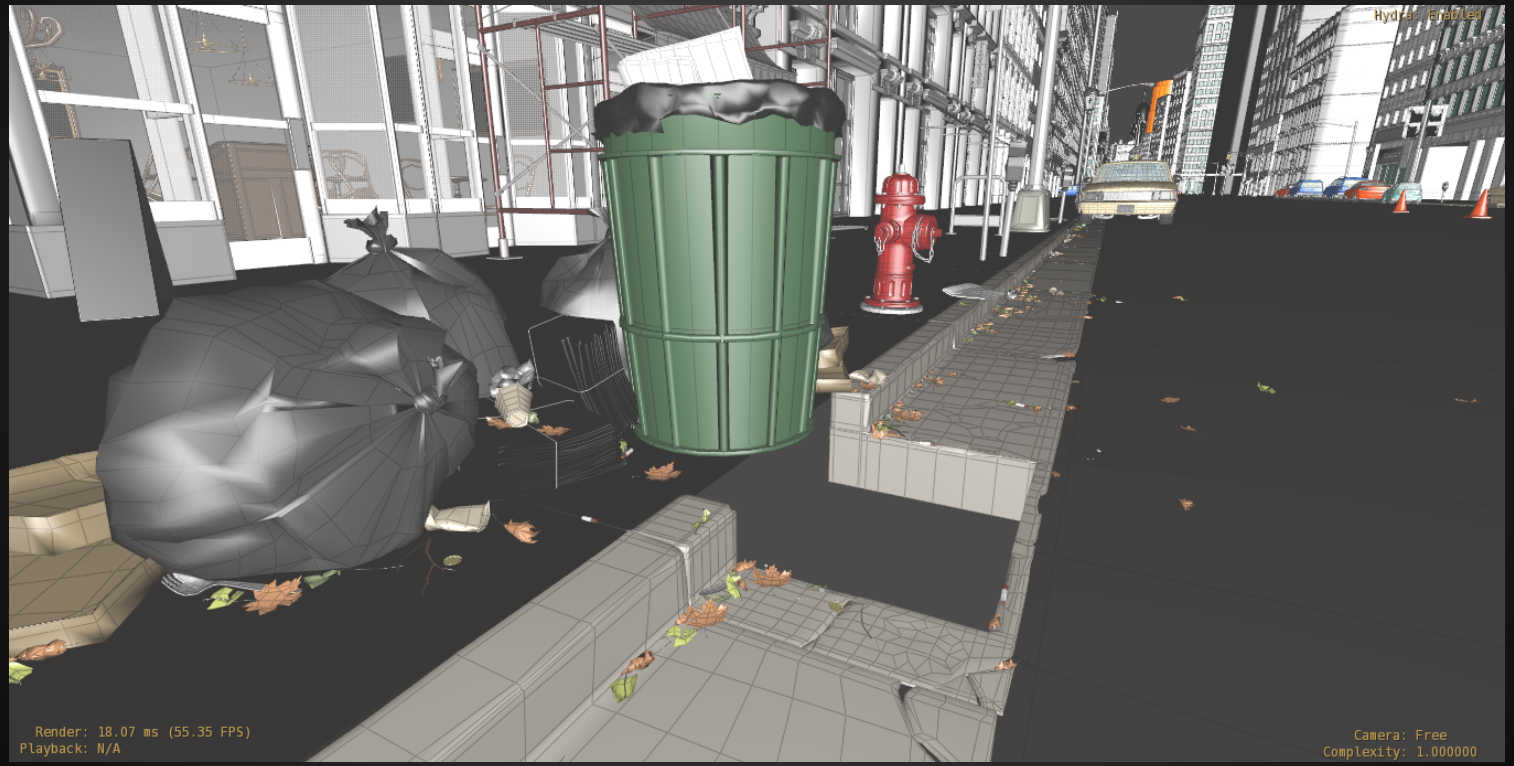
Multi Draw Indirect

GPU TECHNOLOGY CONFERENCE

- 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



Blue Umbrella: City set

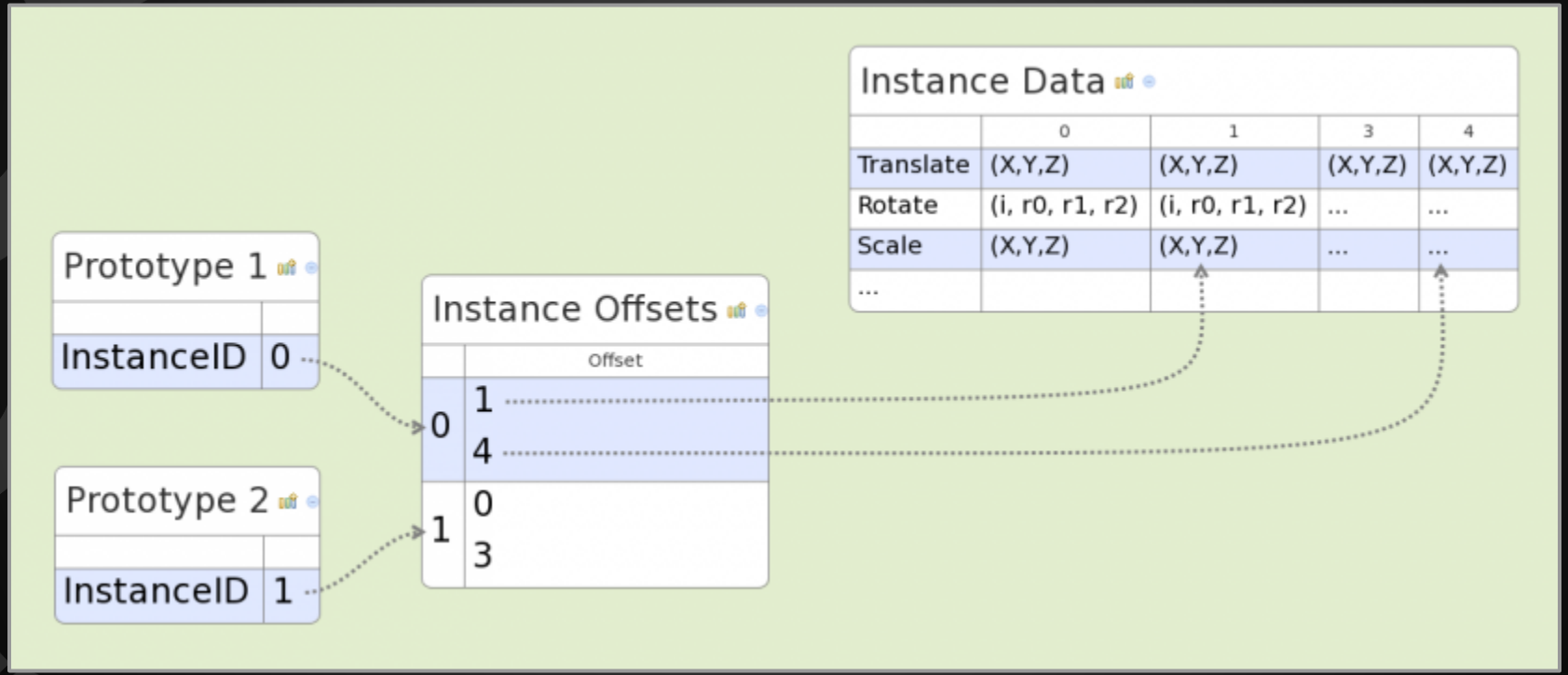
GPU TECHNOLOGY CONFERENCE

270,000 Meshes
29,000,000 Triangles

Quadro 4000
Hardware Max: 890M Tri/s

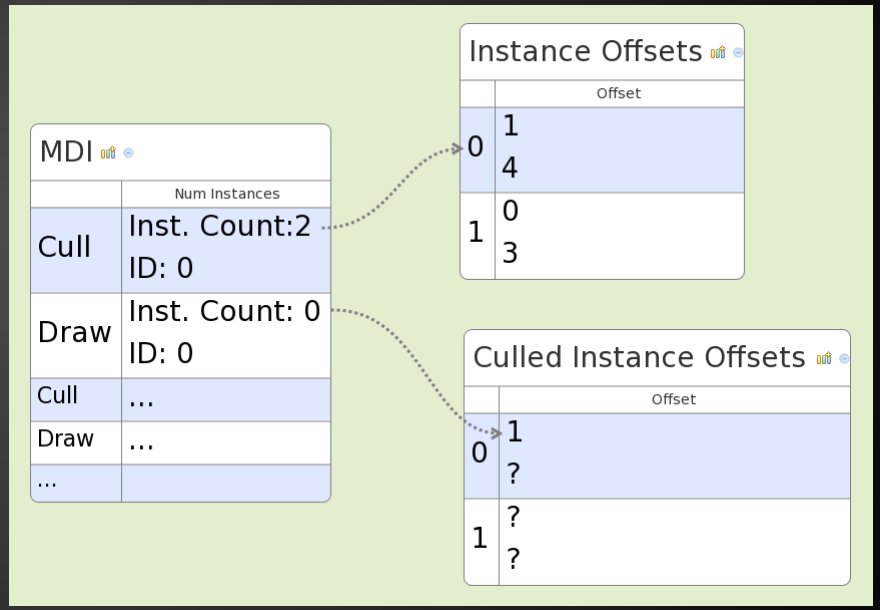
Old school aggregated indices:	17 FPS	493M Tri/s
Unaggregated draw calls:	5 FPS	
Multi Draw Indirect:	16 FPS	
+GPU Screen Presence Culling:	40 FPS +	1160M Tri/s*

GPU Instancing Data Layout



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



Demos!

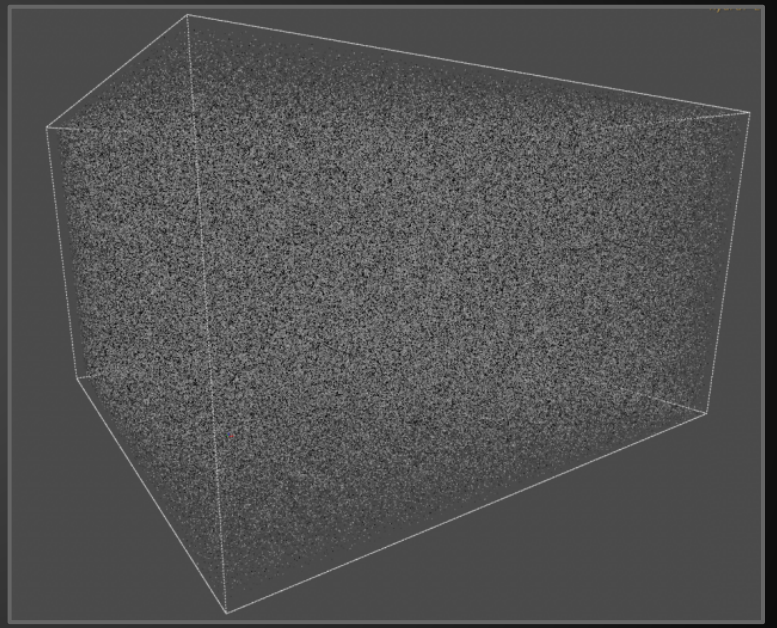
GPU Instancing: Particulate

1.3 Million instances

109 Million faces per frame

Quadro K6000

- 10 FPS Orbit
- 60+ FPS Orbit w/GPU Culling



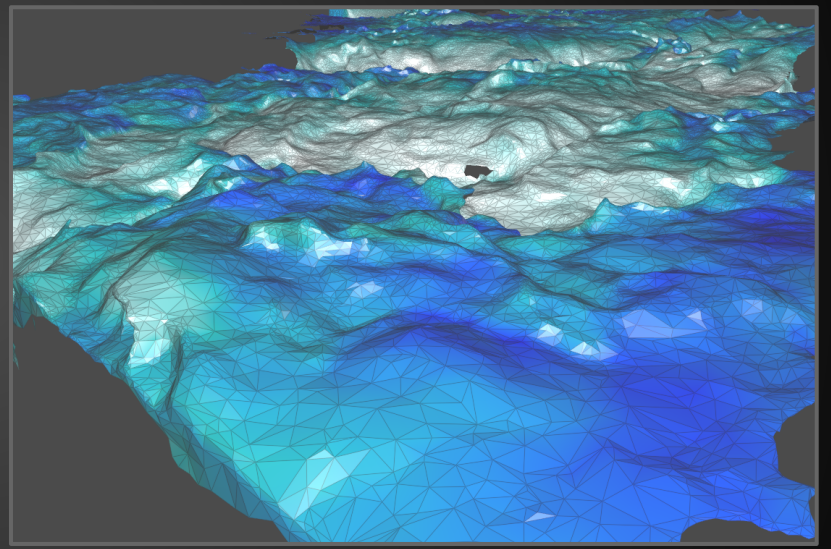
Instancing 5,000 Buzz Lightyears

- 5,000 instances of Buzz
- 72 frames of animation
- Streaming point cache
  - i.e. not rigged
  - no vertex skinning

- 40-55 FPS Playback

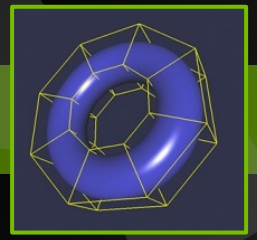
River Simulation Previs

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



Bottlenecks:

- Disk & GPU I/O
- Topology sharing is pure overhead



OpenSubdiv 3.0

- Uniform vs. Adaptive considerations
- Two buckets for mesh topology (mesh & tags)
- GPU Compute for Animation
- 2.x and 3.x during transition

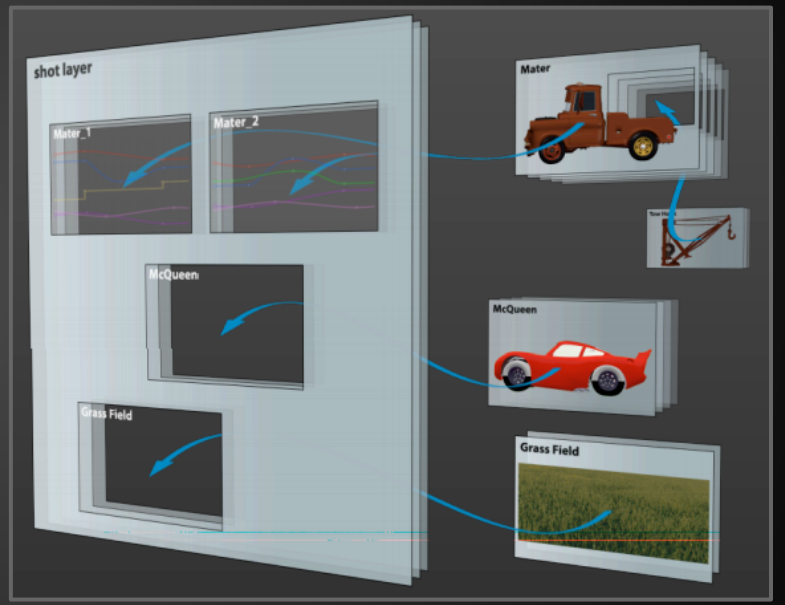
More Info:

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

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!)