GPU光线追踪技术
助力渲染流程改进
SIGGRAPH 2019

What’s new with Arnold GPU

Adrien Herubel
Arnold Core Lead - Autodesk

Steven Parker
Vice President, Professional Graphics - NVIDIA
Why GPU?

- Towards real-time lookdev
- More interactive lighting
- Strong demand
- GPUs are getting bigger
- Easier to scale for small & medium facilities
AUTODESK ARNOLD & NVIDIA OPTIX COLLABORATION
Leveraging OptiX
NVIDIA Partnership

- Benefit from NVIDIA expertise
- OptiX is a compiler: good for complexity
- Great reactivity for missing features
  - Motion blur
  - Faster JIT compilation
  - OSL support
- RTX!
RTX Speedup

- Traditionally 30%-50% time is tracing rays
- RTX makes tracing negligible
- Up to 2x speed-up
- Render time is now mostly shading
Arnold GPU Beta on OptiX

Goal: feature parity
OPTIX / ARNOLD COLLABORATIVE FEATURES

A fruitful engagement

Demand-loaded buffers and textures for large scenes
Separable compilation for large / many shaders
Memory reduction for efficient resource utilization
Stack memory optimizations
Call-graph optimizations
Memory reduction
Performance optimization
AI denoiser optimization
Testing, testing, testing
Initial results courtesy AutoDesk
## Initial results courtesy AutoDesk

<table>
<thead>
<tr>
<th></th>
<th>Near cam</th>
<th>Far Cam</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pre-load textures</td>
<td>On-demand textures</td>
</tr>
<tr>
<td>Time to first pixel (mm:ss)</td>
<td>1:57</td>
<td>0:14</td>
</tr>
<tr>
<td>Time to completion (mm:ss)</td>
<td>2:29</td>
<td>0:59</td>
</tr>
<tr>
<td>GPU memory usage (MiB)</td>
<td>17040</td>
<td>12878</td>
</tr>
</tbody>
</table>
## Texture on-demand loading

### WIP

<table>
<thead>
<tr>
<th></th>
<th>Load all textures</th>
<th>On-demand loading</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to first pixel</td>
<td>01:57</td>
<td>00:14</td>
</tr>
<tr>
<td>Render time</td>
<td>02:29</td>
<td>00:59</td>
</tr>
<tr>
<td>GPU Memory (MiB)</td>
<td>17040</td>
<td>12878</td>
</tr>
</tbody>
</table>
## Texture on-demand loading

### WIP

<table>
<thead>
<tr>
<th></th>
<th>load all textures</th>
<th>On-demand loading</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to first pixel</td>
<td>01:57</td>
<td>00:12</td>
</tr>
<tr>
<td>Render time</td>
<td>02:03</td>
<td>00:33</td>
</tr>
<tr>
<td>GPU Memory (MiB)</td>
<td>17040</td>
<td>6386</td>
</tr>
</tbody>
</table>
RTX Experiments
Increasing shadow rays

- Promising results for direct specular
- Cheap rays unlocking new research
GPU Arnold

Credit: The Mill - Free Yourself - Chemical Brothers
Feature Compatibility

- Final frame renderer
- Support all features
- Some features are hard
- Arnold is very too flexible/customizable
- Focus on character lookdev first
- Eventually support everything
Pixel Compatibility

- Levels of compatibility
  - Good enough for lookdev ← Beta
  - A device per shot ← First release
  - A device per frame
  - Hybrid ← Eventually

- Performance tradeoffs
  - Hardware texturing
  - Filtering
  - Splitting
Shading Networks
OSL on GPU

- Introducing OSL support on GPU in 5.4

Coming later
- OSL textures
- More transforms
- OSL
Motion Blur
CPU vs GPU
Motion Blur
CPU vs GPU
Sampling Patterns
CPU vs GPU
CPU vs GPU
GTC Robot
CPU vs GPU
Nova Girl

CPU

GPU
CPU vs GPU
Cloth
Improved Adaptive Sampling
CPU vs GPU Benchmarks

CPU 6-3-3-3-3-0  15 min 14 sec  1k render

GPU 20-1-1-1-1-0  2 min 56 sec  1k render
CPU vs GPU Benchmarks

CPU 5-3-3-2-2-2 3 min 46 sec

RTX-GPU 20-1-1-1-1-1 46 sec
CPU vs GPU

Cloth
CPU vs GPU
Chad Ashley
GPU Atmospherics

CPU

GPU
GPU Atmospherics
Improved Adaptive Sampling
Transformation Motion Blur
CPU vs GPU

CPU

GPU
Light Sampling
Light Path Expressions

Beauty: C.*
Diffuse Albedo: C<RD>A
Diffuse Direct: C<RD>L
Diffuse Indirect: C<RD>[DSV08].*
Diffuse: C<RD>*
Direct: C[DSV]L
Emission: C[LO]
Indirect: C[DSV][DSV08].*
Specular Direct: C<RS>L
Specular Indirect: C<RS>[DSV08].*
Specular: C<RS>*
Specular Direct: C<RS>L
SSS Albedo: C<TD>A
Direct: C[DSV]L
Indirect: C[DSV][DSV08].*
SSS: C<TD>*
Transmission: C<TB>*
Volume Integrator

WIP

CPU

GPU
Volume Displacement

CPU

GPU
OpenVDB on GPU
New in Arnold 5.4
NEW FEATURES & SUPPORT

- Improved Open Shading Language (OSL) and OpenVDB support
- On-demand texture loading
- New Shadow Matte shader
- Support for most LPEs, lights, and shaders
- Support for all cameras
- Reduced noise
API Compatibility

- Arnold is a library
- Binary compatible
- Device selection API

```c
#include <ai_api.h>

int main(int argc, char* argv[])
{
    int error;

    // start arnold API
    AiBegin();

    // load scene
    error = AiASSLoad(argv[1]);

    // switch to GPU rendering
    AiNodeSetUInt(Ai UniverseGetOptions(), "render_device", 1);

    // select the second GPU
    AtArray* ids = AiArray(1, 0, AI_TYPE_UINT, 1);
    AiDeviceSelect(AI_DEVICE_TYPE_GPU, ids);

    // render
    AiRender(AI_RENDER_MODE_CAMERA);

    // close down API
    AiEnd();

    return 0;
}
```
Shader API

- Arnold C API compatibility
- Shared shader codebase
- Arnold API generates PTX
- Third-party shader support is in sight
USD Components
Now Available on GitHub
Shared Scene

- Scene is shared between CPU/GPU
  - Procedural
  - Instancing
  - Operators
- No overhead if GPU is not used
- Live device switching
Shared Scene

- Scene is shared between CPU/GPU
  - Procedural
  - Instancing
  - Operators
- No overhead if GPU is not
- Live device switching
Arnold GPU

**FLEXIBILITY**

- Switch seamlessly between CPU and GPU rendering
- Choose the type of rendering best suited to your needs
Bringing Arnold to the GPU

Goals

- A single renderer
- Feature compatibility
- Pixel compatibility
- API compatibility
GPU光线追踪技术 助力渲染流程改进

谢谢

ying.shen@autodesk.com