Beyond Programmable Shading: Fundamentals

Aaron Lefohn
Intel

SIGGRAPH2008

Disclaimer about the Course Notes

- **The material in this course is bleeding edge**
  - Unfortunately, that means we could not share most of the details with you until now
  - Most talks are missing from the published notes
  - The talks that were included changed substantially
- **To address this inconvenience**
  - We have posted all course material to a permanent web page
    - http://s08.idav.ucdavis.edu/
  - We have included in the published notes a number of related recently published articles that provide key background material for the course

Future interactive
rendering techniques
will be an inseparable mix of
data- and task-parallel algorithms
and graphics pipelines

How do we write new interactive 3D rendering algorithms?

Fixed-Function Graphics Pipeline

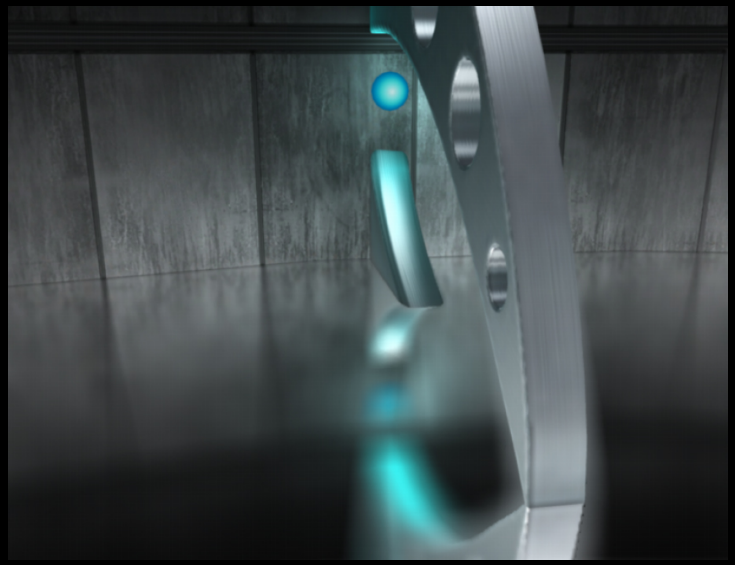
- **Writing new rendering algorithms means**
  - Tricks with stencil buffer, depth buffer, blending, …
  - Examples
    - Shadow volumes
    - Hidden line removal
    - …

Programmable Shading

- **Writing new rendering algorithms means**
  - Tricks with stencil buffer, depth buffer, blending, …
  - Plus: Writing shaders
  - Examples
    - Parallax mapping
    - Shadow-mapped spot light
    - …

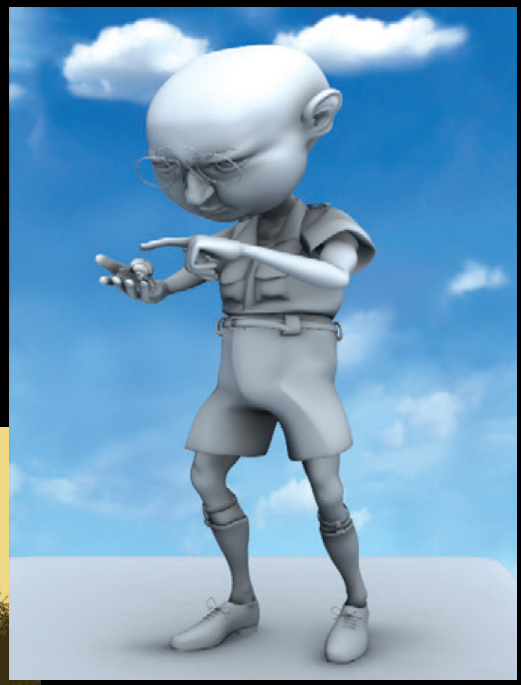
Beyond Programmable Shading

- **Writing new rendering algorithms means**
  - Tricks with stencil buffer, depth buffer, blending, …
  - Plus: Writing shaders
  - Plus: Writing data- and task-parallel algorithms
    - Analyze results of rendering pipeline
    - Create data structures used in rendering pipeline
  - Examples
    - Dynamic summed area table
    - Dynamic quadtree adaptive shadow map
    - Dynamic ambient occlusion
    - …



"Fast Summed-Area Table Generation and its Applications," Hensley et al., Eurographics 2005

"Resolution Matched Shadow Maps," Lefohn et al., ACM Transactions on Graphics 2007



"Dynamic Ambient Occlusion and Indirect Lighting," Bunnell, GPU Gems II, 2005

Beyond Programmable Shading

- **Writing new rendering algorithms means**
  - Tricks with stencil buffer, depth buffer, blending, …
  - Plus: Writing shaders
  - Plus: Writing data- and task-parallel algorithms
    - Analyze results of rendering pipeline
    - Create data structures used in rendering pipeline
  - Plus: Extending, modifying, or creating graphics pipelines
  - Examples
    - PlayStation 3 developers creating hybrid Cell/GPU graphics pipelines
      - See afternoon talk from Jon Olick (Id Software)
    - Active area of research

Why "Beyond Programmable Shading?"

- **Short answer:**
  - The parallel processors in your desktop machine or game console are now flexible and powerful enough to execute both
    - User-defined parallel programs and
    - Graphics pipelines
    - …All within 1/30th of a second

The Point Is…

- **Interactive graphics programming is changing**

- **This morning's course gives you:**
  - Introduction to the HW causing/enabling this change
  - Programming tools used to explore this new world
  - A little bit about what developers/researchers can do with these new capabilities

    - And the afternoon course…

This Afternoon Course

- **"Beyond Programmable Shading: In Action"**
  - Case studies from game developers, academics, and industry

This Afternoon Course

- **"Beyond Programmable Shading: In Action"**
  - Showcasing new interactive rendering algorithms that result in more realistic imagery than is possible using only the pre-defined D3D/OpenGL graphics pipeline by
    - Combining task-, data-, and/or graphics pipeline parallelism,
    - Analyzing intermediate data produced by graphics pipeline,
    - Building and using complex data structures every frame, or
    - Modifing/extending the graphics pipelines

Speakers (in order of appearance)

- Aaron Lefohn, Intel
- Kayvon Fatahalian, Stanford
- Dave Luebke, NVIDIA
- Mike Houston, AMD
- Tom Forsyth, Intel
- John Owens, UC Davis
- Chas Boyd, Microsoft
- Aaftab Munshi, Apple
- Fabio Pellacini, Dartmouth
- Jon Olick, Id Software
- Jeremy Shopf, AMD
- Matt Pharr, Intel

Morning Schedule

• **Intro**	8:30 – 8:40	Lefohn
• **GPU Architectures**		
– Overview	8:40 – 8:55	Fatahalian
– NVIDIA GPU Architecture	8:55 – 9:15	Luebke
– Anatomy of AMD's GPU	9:15 – 9:35	Houston
– Larrabee Graphics Architecture	9:35 – 9:55	Forsyth
• **GPU Programming Models**		
– Overview	9:55 – 10:15	Owens
<Break>	10:15 – 10:30	
– AMD Stream SDK	10:30 – 10:50	Houston
– CUDA Fundamentals	10:50 – 11:10	Luebke
– DirectX 11 ComputeShader	11:10 – 11:30	Boyd
– OpenCL	11:30 – 11:50	Munshi
– Programming Larrabee	11:50 – 12:10	Lefohn
• **Q & A**	12:10 – 12:15+	All

Afternoon Schedule

• **Introduction**	1:45 – 2:00	Houston
• **Research case study**		
– Interactive Cinematic Lighting	2:00 – 2:30	Pellacini
• **Game developer case study**		
– Current Generation Parallelism in Games	2:30 – 3:00	Olick
– Next Generation Parallelism in Games	3:00 – 3:30	Olick
<Break>	3:30 – 3:45	
• **Hardware vendor case studies**		
– NVIDIA Case Studies: Compute Enabled Graphics	3:45 – 4:15	Luebke
– AMD Case Study: March of the Froblins	4:15 – 4:45	Shopf
– Advanced Rendering on Larrabee	4:45 – 5:15	Pharr
• Q & A	5:15 – 5:30+	All

http://s08.idav.ucdavis.edu/