Beyond Programmable Shading

Why and How is Interactive Graphics Programming Changing

Aaron Lefohn
Intel Corporation
Disclaimer about the Course Notes

- The material in this course is bleeding edge
  - 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
  - All course material posted to a permanent web page
    - http://s09.idav.ucdavis.edu/
Interactive rendering techniques are created using 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
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 photon mapping
  - ...

Beyond Programmable Shading

http://s09.idav.ucdavis.edu/
“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

“Real-Time Approximate Sorting for Self Shadowing and Transparency in Hair Rendering,”
Sintorn et al., I3D 2008
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
  - PS3/XBox360 hybrid Cell/CPU + GPU graphics pipelines
  - Interactive ray tracing
  - Interactive micropolygon rendering
  - *Five talks this year in this course on this topic*
Braided Parallelism

- Intermixed task-, data-, and pipeline parallelism

"Braid: Integrating Task and Data Parallelism,”
West and Grimshaw, 1994

Image from Johan Andersson, DICE
This Course Offers...

- Introduction to the HW causing/enabling this change
- Parallel programming models for graphics
- Game developers, academic researchers, and IHVs:
  - Showcasing new interactive rendering algorithms that result in more realistic imagery than is possible using only the pre-defined D3D/OpenGL graphics pipeline
Morning Schedule

- Why and How is Interactive Graphics Programming Changing?
  8:30 – 8:50  Aaron Lefohn, Intel

- Beyond Programmable Shading Retrospective
  8:50 – 9:15  Mike Houston, AMD

- Running Code at a Teraflop: Overview of GPU Architecture
  9:15 – 10:00  Kayvon Fatahalian, Stanford

- Q&A
  10:00 – 10:15  All Speakers

- Break
  10:15 – 10:30

- Parallel Programming for Interactive Graphics
  10:30 – 11:10  Tim Foley, Stanford / Intel

- Parallel Graphics in Frostbite - Current & Future
  11:10 – 11:45  Johan Andersson, DICE

- id tech 5 Challenges
  11:45 – 12:15  J.M.P van Wavern, id software
Afternoon Schedule

- Intro / recap of morning course  
  1:45 – 1:55  
  Mike Houston, AMD

- GPU Primitives---Case Study: Hair Rendering  
  1:55 – 2:25  
  Ulf Assarsson, Chalmers University

- A Real-Time Micropolygon Rendering Pipeline Is Not Far Away  
  2:25 – 3:00  
  Kayvon Fatahalian, Stanford

- AMD Case Study  
  3:00 – 3:30  
  Justin Hensley, AMD

- Break  
  3:30 – 3:45

- Innovating in a Software Graphics Pipeline  
  3:45 – 4:15  
  Paul Lalonde, Intel

- NVIDIA Case Study  
  4:15 – 4:45  
  David Luebke, NVIDIA

- Panel: "What Next for 3D Graphics Programming Models?"  
  4:45 – 5:30  
  Moderator: Kurt Akeley
Additional Panelists

- Cass Everitt, Epic Games
- Jonathan Ragan-Kelley, MIT
Be on the Lookout For...

- What forms of parallelism are used?
- Which hardware is used and why?
- Why was the standard graphics pipeline not enough?
- What is the future of the existing graphics APIs?
Course Evaluations

- Please fill in the course evaluation at: http://www.siggraph.org/courses_evaluation/
- You could win a Siggraph’09 mug
- One winner per course, notified by email in the evening

http://s09.idav.ucdavis.edu