CS 380 - GPU and GPGPU Programming
Lecture 2: Introduction; GPU Architecture 1

Markus Hadwiger, KAUST
Example: Fluid Simulation and Rendering

- Compute advection of fluid
  - (Incompressible) Navier-Stokes solvers
  - Lattice Boltzmann Method (LBM)

- Discretized domain; stored in 2D/3D textures
  - Velocity, pressure
  - Dye, smoke density, vorticity, ...

- Updates in multi-passes

- Render current frame

Courtesy Mark Harris
Example: Volumetric Special Effects

- NVIDIA Demos
  - Smoke, water
  - Collision detection with voxelized solid (Gargoyle)

- Ray-casting
  - Smoke: direct volume rendering
  - Water: level set / isosurface

Courtesy Keenan Crane
Example: Particle Simulation and Rendering

• NVIDIA Particle Demo
Example: Level-Set Computations

- Implicit surface represented by distance field
- The level-set PDE is solved to update the distance field
- Basic framework with a variety of applications
Example: Diffusion Filtering

De-noising

- Original
- Linear isotropic
- Non-linear isotropic
- Non-linear anisotropic
Example: Linear Algebra Operators

Vector and matrix representation and operators

- Early approach based on graphics primitives
- Now CUDA makes this much easier
- Linear systems solvers

Courtesy Krüger and Westermann
Glift: Generic, Efficient, Random-Access GPU Data Structures

- “STL“ for GPUs
- Virtual memory management

Courtesy Lefohn et al.
What’s in a GPU?

Lots of floating point processing power

- Stream processing cores
  - different names:
    - stream processors,
    - CUDA cores, ...
  - Was vector processing, now scalar cores!

Still lots of fixed graphics functionality

- Attribute interpolation (per-vertex -> per-fragment)
- Rasterization (turning triangles into fragments/pixels)
- Texture sampling and filtering
- Depth buffering (per-pixel visibility)
- Blending/compositing (semi-transparent geometry, ...)
- Frame buffers
What can the hardware do?

- **Rasterization**
  - Decomposition into fragments
  - Interpolation of color
  - Texturing
    - Interpolation/Filtering
    - Fragment Shading

- **Fragment Operations**
  - Depth Test (Z-Test)
  - Alpha Blending (Compositing)
Graphics Pipeline

Scene Description

Geometry Processing

Rasterization

Fragment Operations

Vertices → Primitives → Fragments → Pixels

Raster Image
Geometry Processing

- **Transformation**: Multiplication with Modelview and Projection Matrix
- **Per-Vertex Lighting**: Per-Vertex Local Illumination (Blinn/Phong)
- **Primitive Assembly**: Geometric Primitives (Points, Lines, Triangles)
- **Clipping, Perspect. Divide**: Clip Space To Screen Space

**Vertices** -> **Primitives**
Rasterization

**Geometry Processing**

**Rasterization**

**Fragment Operations**

**Polygon Rasterization**
- Decomposition of primitives into fragments

**Texture Fetch**
- Interpolation of texture coordinates
  - Filtering of texture color

**Texture Application**
- Combination of primary color with texture color

Primitives → Fragments
Combination of primary color with texture color

Fragment Operations

Geometry Processing ➔ Rasterization ➔ Fragment Operations

- **Alpha Test**: Discard all fragments within a certain alpha range
- **Stencil Test**: Discard a fragment if the stencil buffer is set
- **Depth Test**: Discard all occluded fragments
- **Alpha Blending**: Combination of primary color with texture color

Discard all fragments within a certain alpha range

Discard a fragment if the stencil buffer is set

Discard all occluded fragments

Combination of primary color with texture color
Direct3D 10 Pipeline (~OpenGL 3.2)

New geometry shader stage:
- Vertex -> geometry -> pixel shaders
- Stream output after geometry shader

Courtesy David Blythe, Microsoft
New tessellation stages

- Hull shader
  (OpenGL: *tessellation control*)

- Tessellator
  (OpenGL: *tessellation primitive generator*)

- Domain shader
  (OpenGL: *tessellation evaluation*)

- Trend of adding new stages likely to continue...

- ... or full flexibility such as in Intel MIC (Larrabee) architecture?
GPU Structure Before Unified Shaders

**Vertex Processors**

- Host
  - Cull/Clip/Setup
  - Z-Cull
  - Rasterization

**Fragment Processors**

- Texture Cache
  - Fragment Crossbar
  - Memory Access
    - Z-Compare and Blending

**Memory Access**

- Memory Partition
  - Memory Partition
  - Memory Partition
  - Memory Partition

Example NVIDIA GeForce 6/7, 2004, 2005
Thank you.