Overview of 3D Graphics Processing

연세대학교 이용석 교수 연구실 박사과정 김현필
E-mail: yonglee@yonsei.ac.kr
Homepage: http://mpu.yonsei.ac.kr
전화: 02-2123-2872

Reference

[1] 이병래 외, "컴퓨터 그래픽스", 홍릉과학 출판사
[3] 이병래 외, "3D 그래픽 칩 관련 기술 및 산업 동향, 기업소개" ITSoC Magazine
[5] The OpenGL Shading Language
[9] 박재형 외, "모바일 3D 기술", ETRI 표준 기술동향

Agenda

• Introduction
• Graphic Processing
• GPU Evolving
• The Newest GPU Technology
• API for 3D Graphics Accelerating
• GPU for Mobile Device
INTRODUCTION

1. GPU (Graphic Processing Unit)
2. Motivation
   - Computational Power
   - Flexible and Precise
   - The Potential of GPGPU
3. Problem

Motivation: Computational Power

- GPUs are fast.
  - 3 GHz Pentium 4: 6 GFLOPS, 5.96 GB/sec peak
  - GeForceFX 5900: 20 GFLOPs, 25.3 GB/sec peak
- GPUs are getting faster, faster.
  - CPUs: annual growth $\times 1.5 \rightarrow$ decade growth $\times 60$
  - GPUs: annual growth $\rightarrow 2.0 \rightarrow$ decade growth $\rightarrow 1000$
- Why are GPUs getting faster?
  - Arithmetic intensity
  - Economics

Motivation: Flexibility and Precision

- Modern GPUs are deeply programmable
  - Programmable pixel, vertex, video engines
  - Solidifying high-level language support
- Modern GPUs support high precision
  - 32 bit floating point throughout the pipeline
  - High enough for many (not all) applications

Motivation: Potential of GPGPU

- The power and flexibility of GPUs makes them an attractive platform for general-purpose computation.
- Example applications (from GPGPU.org)
  - Advanced Rendering: Global Illumination, Image Image-based Modeling
  - Computational Geometry
  - Computer Vision
  - Image and Volume Processing
  - Scientific Computing: physically physically-based simulation, linear system solution, PDEs
  - Database queries
  - Monte Carlo Methods

Problem: Difficult To Use

- GPUs designed for and driven by video games
  - Programming model is unusual
  - Programming environment is tightly constrained
- Underlying architectures are:
  - Inherently parallel
  - Rapidly evolving (even in basic feature set!)
  - Largely secret
- Can’t simply "port" code written for the CPU!

What's GPU?

- The GPU (Graphic Processing Unit) on commodity video cards has evolved into an extremely flexible and powerful processor
  - Programmability
  - Precision
  - Power

(a) Animation Movie
(b) 3D Game
GPU Application Market\[3][4]\n
<table>
<thead>
<tr>
<th></th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
<th>CAGR(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop PC Discrete</td>
<td>82.1</td>
<td>82.3</td>
<td>81.6</td>
<td>80.4</td>
<td>-0.7</td>
</tr>
<tr>
<td>Notebook PC Discrete</td>
<td>21.2</td>
<td>21.3</td>
<td>25.6</td>
<td>28.6</td>
<td>10.5</td>
</tr>
<tr>
<td>Integrated Chips</td>
<td>162.2</td>
<td>183.5</td>
<td>195.2</td>
<td>210.2</td>
<td>9.2</td>
</tr>
<tr>
<td>Handheld</td>
<td>817.7</td>
<td>922.9</td>
<td>994.4</td>
<td>1071.9</td>
<td>9.4</td>
</tr>
<tr>
<td>Digital TV</td>
<td>73.7</td>
<td>98.3</td>
<td>114.9</td>
<td>134.5</td>
<td>22.2</td>
</tr>
<tr>
<td>Game Consoles</td>
<td>17</td>
<td>26.5</td>
<td>39.7</td>
<td>42.3</td>
<td>35.5</td>
</tr>
</tbody>
</table>

3D GRAPHICS PROCESSING
Graphic hardware pipeline
Vertex transformation & Lighting
Primitive assembly & Rasterization
Fragment texturing & Coloring
Raster operations

Graphic Hardware Pipeline

Vertex Translation

Primitive Assembly & Rasterization

Primitive Assembly
**Fragment Texturing & Coloring**

- **Application Stage**
  - User Input & Collision Detection
  - Vertex Connectivity

- **Geometry Stage**
  - Texture Transform & Lighting
  - Transformed Vertices

- **Rasterization Stage**
  - Fragment Texturing & Coloring
  - Colored Fragments

**Raster Operations (1)**

- **Application Stage**
  - User Input & Collision Detection
  - Vertex Connectivity

- **Geometry Stage**
  - Texture Transform & Lighting
  - Transformed Vertices

- **Rasterization Stage**
  - Fragment Texturing & Coloring
  - Colored Fragments

**Raster Operations (2)**

- **Data Flow of Graphic Pipeline**
  - Vertex Transformation & Lighting
  - Rasterization
  - Texturing & Coloring

**1st Generation GPU**

- **GPU EVOLVING**
  - 3D Graphics Process
  - 1st Generation
  - 2nd Generation
  - 3rd Generation
**2nd Generation GPU**

- CPU
  - Application Stage
  - Halstages
  - System Memory

- GPU
  - Geometry Stage
  - Rasterization Stage
  - Texture Unit
  - Register Combiner
  - Rasterizer
  - Raster Operation Unit
  - Video Memory

**3rd Generation GPU**

- CPU
  - Application Stage
  - Transform & Lighting
  - Textures
  - System Memory

- GPU
  - Geometry Stage
  - Rasterization Stage
  - Texture Unit
  - Pixel Shaders
  - Rasterizer
  - Raster Operation Unit
  - Video Memory

---

**Programmable Vertex Shader**

1. Load vertex attribute into temporary registers
2. Read vertex attribute
3. Apply transformation
4. Write temporary registers

**Programmable Pixel Shader**

1. Load pixel attribute into temporary registers
2. Read pixel attribute
3. Apply transformation
4. Write temporary registers

---

**THE NEWEST GPU TECHNOLOGY**

1. Unified Shader
2. GPGPU

---

**Unified Shader**

- Unified Shaders
- Frame Buffer
- Texture Unit
- Vertex Fetcher
- Graphic Memory
GPGPU (General Purpose GPU)

- Computational resource
  - Programmable processors
    - Vertex, primitive and fragment pipelines allow programmer to perform kernel on streams of data
  - Rasterizer
    - Creates fragments and interpolates per-vertex constants such as texture coordinates and color
  - Texture Unit: read only memory interface
  - Frame buffer: write only memory interface
- Texture as stream
- Kernel
- Flow control

API FOR 3D GRAPHICS ACCELERATING

1. OpenGL
2. DirectX Direct 3D

OpenGL

- API for 3D graphic application
- Open standard
- Verified on variable OS
  - Windows 95, NT, X Windows, OS/2
- Mainly used to PC or workstation
- Ver. 2.1 has vertex and fragment shader.
- OpenGL|ES is API for embedded system.

DirectX Direct 3D

- API for Windows’s multimedia application (2D, 3D, video, audio)
- Like as OpenGL
- Shader model
  - Vertex, pixel, geometry shader
- Mobile purpose
  - Direct 3D mobile
  - Not support vertex and pixel shader

GPU FOR MOBILE DEVICE

1. Khronos Group
2. JCP
Khronos Group

- Standardization of Audio, Video, 2D, 3D Graphic API and Development Environment for Embedded or Mobile
  - OpenGL|ES (OpenGL for Embedded Systems)
  - OpenML (Open Media Library)
  - OpenVG (Open Vector Graphics)
  - OpenMAX (Open Media Acceleration Primitives)

JCP (Java Community Process)

- Java Platform Standardization of Java Spec., Reference Implementation, Compatibility Tool Development
  - JSR184 (Mobile 3D Graphics API for J2ME)
  - JSR239 (Java Bindings for OpenGL|ES)

THANK YOU !!!