OpenGL ES 3.0
Features and Benefits

FTF-SDS-F0045

Rick Tewell  |  Senior Graphics Architect
A P R . 2 0 1 4
Session Introduction

• **OpenGL ES 3.0** has a large number of significant enhancements over OpenGL ES 2.0.

• In this session we will describe some of the most interesting of these enhancements and how you can benefit by using them.

• I am the Senior Graphics Architect and lead of the Freescale Graphics Technology Engineering Center (GTEC).

• This session should run around one hour – hope to have time for a few questions at the end.
AN ORGANIZED TEAM OF CUSTOMER CENTRIC GRAPHICS TECHNOLOGY EXPERTS FOCUSED ON STRATEGY, ARCHITECTURE, SOFTWARE AND HARDWARE SOLUTIONS AT THE IC AND SYSTEM LEVEL FOR BOTH TRADITIONAL GRAPHICS AND GPU COMPUTE APPLICATIONS
Functional Teams

**GPU South**
- Kernel Level Drivers
- API Level Libraries (GLES, VG, etc.)
- Validate New SW from GPU IP Provider
- Debugging / Test
- Customer Support

**Graphics**
- Graphics “Above” the API Level
- Benchmarking
- Testing
- Demonstration Applications
- Customer Support / Education

**Compute**
- OpenCL
- Compute Shaders
- Benchmarking
- Testing
- Customer Support / Education
- Demonstration Applications
Functional Teams

Studio Solutions
- Conceptual Art
- 2D / 3D artwork creation
- Artwork Optimization
- Technical Art
- Tutorials
- Demos
- Benchmarks

3rd Party Ecosystem
- Graphics Engines (2D / 3D)
- Debuggers
- Performance Analysis
- HMI Tooling
- OS Vendors
Session Objectives

• After completing this session you will be able to:
  - **Understand** the practical use and application of OpenGL ES 3.0
  - **Know** how to leverage Freescale graphics SDK and third party tools to use OpenGL ES 3.0
  - **Find** additional resources to go deeper and further with GPU compute
Agenda

- A Brief Overview of OpenGL ES (levels and features)
- What does OpenGL ES 3.0 bring to the party? (new features)
- Significant OpenGL ES 3.0 features (details)
- OpenGL ES 3.0 in Action - tools and demo
- Additional Resources
Agenda

• A Brief Overview of OpenGL ES (levels and features)
• What does OpenGL ES 3.0 bring to the party? (new features)
• Significant OpenGL ES 3.0 features (details)
• OpenGL ES 3.0 in Action - tools and demo
• Additional Resources
A Brief Overview of OpenGL ES (Levels and Features)

• OpenGL – Desktop

• OpenGL ES 1.0, 1.1 (i.MX35, MPC5121, etc)
  - Fixed function graphics pipeline
  - Not compatible with ES 2.0

• OpenGL ES 2.0, 3.0 (i.MX5x, 6x)
  - Programmable pipeline (shader programs)
    ▪ Requires shader compiler (uses GLSL)
  - Compatible with ES 1.1 by a shader program
A Brief Overview of OpenGL ES (Levels and Features)

• 3D Graphics API – based on OpenGL 1.5

• Subset of desktop OpenGL for “embedded systems”

• Fixed point data types introduced for vertex coordinates

• Designed for “fixed function” graphics hardware

• Changes from desktop OpenGL are for performance and simplification:
  – Removed specifying primitives directly to the API (glBegin/glEnd)
  – Attributes and vertex arrays used to draw
  – Only triangles are used to draw
A Brief Overview of OpenGL ES (Levels and Features)

Fixed Function Graphics Pipeline (1.0, 1.1)
A Brief Overview of OpenGL ES (Levels and Features)

- **OpenGL ES 2.0**
  - 3D Graphics API – based on OpenGL 2.0
  - Emphasizes a programmable 3D graphics pipeline
  - Is NOT compatible with OpenGL ES 1.0 / 1.1
  - Requires shader programs based on OpenGL ES Shading Language (GLSL)
  - Adds frame buffer objects
A Brief Overview of OpenGL ES (Levels and Features)

Programmable Graphics Pipeline
Vertex Shader Functions

Receives vertex data as input (position, colors, normals…)

• The vertex shader can do:
  − Transformation of position using model-view and projection matrices
  − Transformation of normals, including renormalization
  − Texture coordinate generation and transformation
  − Per-vertex lighting
  − Color computation

• The vertex shader cannot:
  − Do anything that requires information from more than one vertex
  − Do anything that depends on connectivity
  − Do any triangle operations (e.g. clipping, culling)
  − Access color buffer

Responsible for writing AT LEAST `gl_Position`
A Brief Overview of OpenGL ES (Levels and Features)

**Fragment Shader Functions**

Receives rasterizer output as input and varying parameters from the vertex shader.

- The fragment shader can do:
  - Texture blending
  - Fog
  - Alpha testing
  - Dependent textures
  - Pixel discarding
  - Bump and environment mapping

- The fragment shader cannot do:
  - Blending with color buffer
  - ROP operations
  - Depth or stencil tests
  - Write depth

  Responsible for writing `gl_FragColor` (fragment color)
• A Brief Overview of OpenGL ES (levels and features)
• What does OpenGL ES 3.0 bring to the party? (new features)
• Significant OpenGL ES 3.0 features (details)
• OpenGL ES 3.0 in Action - tools and demo
• Additional Resources
What does OpenGL ES 3.0 bring to the party? (new features)

One of the biggest leaps in OpenGL history

• Over 50 new features

• Focus on creating application compatibility on divergent HW platforms

• Targeted significant accelerated performance

• Greatly enhanced texturing functionality including vast set of formats

• Fully backward compatible to OpenGL ES 2.0

• Supported on i.MX6
What does OpenGL ES 3.0 bring to the party? (new features)

- OpenGL Shading Language ES 3.00
- transform feedback 1 and 2 (with restrictions)
- uniform buffer objects including block arrays
- vertex array objects
- sampler objects
- sync objects and fences
- pixel buffer objects
- buffer subrange mapping
- buffer object to buffer object copies
- boolean occlusion queries, including conservative mode
- instanced rendering, via shader variable and/or vertex attribute divisor
- multiple render targets
- 2D array and 3D textures
- simplified texture storage specification
- R and RG textures
- texture swizzles
- seamless cube maps
- non-power-of-two textures with full wrap mode support and mipmapping
- texture LOD clamps and mipmap level base offset and max clamp
- at least 32 textures, at least 16 each for fragment and vertex shaders
- 16-bit (with filtering) and 32-bit (without filtering) floating-point textures
- 32-bit, 16-bit, and 8-bit signed and unsigned integer renderbuffers, textures, and vertex attributes
- 8-bit sRGB textures and framebuffers (without mixed RGB/sRGB rendering)
- 11/11/10 floating-point RGB textures
- shared exponent RGB 9/9/5 textures

- 10/10/2 unsigned normalized and unnormalized integer textures
- 10/10/2 signed and unsigned normalized vertex attributes
- 16-bit floating-point vertex attributes
- 8-bit-per-component signed normalized textures
- ETC2/EAC texture compression formats
- sized internal texture formats with minimum precision guarantees
- multisample renderbuffers
- 8-bit unsigned normalized renderbuffers
- depth textures and shadow comparison
- 24-bit depth renderbuffers and textures
- 24/8 depth/stencil renderbuffers and textures
- 32-bit depth and 32F/8 depth/stencil renderbuffers and textures
- stretch blits (with restrictions)
- framebuffer invalidation hints
- primitive restart with fixed index
- unsigned integer element indices with at least 24 usable bits
- draw command allowing specification of range of accessed elements
- ability to attach any mipmap level to a framebuffer object
- minimum/maximum blend equations
- program binaries, including querying binaries from linked GLSL programs
- mandatory online compiler
- non-square and transposable uniform matrices
- additional pixel store state
- indexed extension string queries
What does OpenGL ES 3.0 bring to the party? (new features)

- Occlusion queries
- Transform feedback
- Instanced rendering
- Multiple render targets
- ETC2 / EAC texture compression
- GLSL fully supports integer and 32-bit floating point operations
- Greatly enhanced texturing functionality
- Explicit texture and render buffer formats
Agenda

• A Brief Overview of OpenGL ES (levels and features)
• What does OpenGL ES 3.0 bring to the party? (new features)
• Significant OpenGL ES 3.0 features (details)
• OpenGL ES 3.0 in Action - tools and demo
• Additional Resources
Significant OpenGL ES 3.0 features (details)

Occlusion Queries

- Tessellated objects (bounding boxes) may retrieve number of pixels that passed depth test
- Occlusion Culling allows to discard occluded objects before processing through render pipeline in full resolution
- Further, allows to control light and lens flare intensities

Advantage:

- Performance leap and less bandwidth and power consumption with occlusion culling for scenes with multiple obscured objects (e.g. building blocks, car)
- Visual stunning bloom and lens flare effects (e.g. sun, head lights)
Significant OpenGL ES 3.0 features (details)

- **Geometry Instancing**
  - Renders multiple copies of same mesh with one single draw call. Geometry is shared, appearance like material, and texture may differ.
  - Accelerated rendering of clones

- **Advantage**
  - Performance improvement due to less state changes and batched rendering for scenes with multiple repetitive elements (e.g. particles, buildings, tires, any repeated geometry)
  - Visual improvement due to immersive scenes (e.g. weather effects - rain, haze, and rich environment)
Significant OpenGL ES 3.0 features (details)

- **Texture Compression**
  - ETC2 and EAC texture compression
  - Compressed textures are stored and accessed directly in VRAM
  - Reduces memory footprint and usage of texture lookup bandwidth

- **Advantage**
  - Significantly reduced memory consumption
  - Performance Improvement due to reduced bandwidth consumption
  - Better start-up times, texture upload is faster
  - Visual improvement as compressed textures allow bigger dimension at same memory size compared to uncompressed textures
Significant OpenGL ES 3.0 features (details)

- Self-shadowing using depth textures with shadow sampling
- HDR using RGB10_A2 render-to-texture and tone mapping
- Multi-layered track texture using 2D texture arrays
- ETC2 / EAC texture compression used throughout

Content: ARM "Timbuktu 2" tech demo
Significant OpenGL ES 3.0 features (details)

1. Occlusion query used to determine light visibility
2. Instanced drawing used for vehicles and particles
3. Deferred rendering using multiple render targets (MRT) and depth textures
Agenda

• A Brief Overview of OpenGL ES (levels and features)
• What does OpenGL ES 3.0 bring to the party? (new features)
• Significant OpenGL ES 3.0 features (details)
• OpenGL ES 3.0 in Action - tools and demo
• Additional Resources
OpenGL ES 3.0 in Action - Tools and Demo

https://unity3d.com/
OpenGL ES 3.0 in Action - Tools and Demo

https://unity3d.com/
OpenGL ES 3.0 in Action - Tools and Demo
OpenGL ES 3.0 in Action - Tools and Demo

2D&3D Candera Render Engines

Scene Composer

Player

Analyzer

Courier Interaction Frame

Asset Tool Suite

State Machine Connector

Multi Language Support

Photoshop Importer

HTML 5 Module

Memory Manager

Additional modules: Warping, Stereoscopy, HarfBuzz

ASIL Module
OpenGL ES 3.0 in Action - Tools and Demo

Professional Artist Tooling for 2D and 3D Graphics

- FBX, PNG, PSD, ...

Graphic Designs

- Scene Composer
- Player
- Courier
- Analyzer
- ...

Export of Asset Library

GUI Application

- Assets – Widgets - Animations

Candera 2D/3D Engines

Application State Machine
Business Logic
(Matlab Simulink, IAR Visualstate, …)

PC Development Environment

- PC Hardware
- Platform Abstraction

Target Environment (Embedded Hardware)

- Target Hardware
- Platform Abstraction

Real-time Operating System and Framework

(Linux, Windows, Integrity, Android, T-Kernel, Tizen, …)
Agenda

• A Brief Overview of OpenGL ES (levels and features)
• What does OpenGL ES 3.0 bring to the party? (new features)
• Significant OpenGL ES 3.0 features (details)
• OpenGL ES 3.0 in Action - tools and demo
• Additional Resources
Additional Resources

i.MX6 Community
- https://community.freescale.com/community/imx

i.MX6 Graphics SDK
- http://freescale.com/imx6
Click on i.MX6D
Software & Tools tabs
Software Development Tools-> Snippets, Boot Code, Headers, Monitors, etc.- IMX6_GPU_SDK

i.MX6 Graphics Vivante Tools
- http://freescale.com/imx6
Click on i.MX6D
Software & Tools tabs
Software Development Tools->IDE – Debug, Compile, and Build Tools -> IMX_6D_Q_VIVANTE_VDK_145_TOOLS
Additional Resources

Khronos OpenGL ES
OpenGL ES 3.0 Specification
- http://www.khronos.org/opengles/3_X/

OpenGL ES Registry
- http://www.khronos.org/registry/gles/

OpenGL Developer Forums
- http://www.khronos.org/message_boards/

OpenGL ES 3.0 Quick Reference Card

OpenGL ES 3.0 Online Man pages
Now Available...

**OpenGL ES 3.0 Programming Guide**

By Dan Ginsburg & Budirijanto Purnomo


Publishing Date: March 2014

Addison-Wesley Professional

http://www.informit.com/title/0321933885

Available in Print, major eBook formats, and Safari Books Online.

ORDER NOW from InformIT or your local book store or online reseller.

Discount code: OPENGLES35