OpenCL @ Adobe

Eric Berdahl
Adobe Imaging Foundation Engineering Manager
Adobe Systems Inc.
History of Heterogeneous Computing at Adobe

- **AIF Project**
  - Formed

- **Point product support for OpenGL**

- **Pre-2005**
- **2005**
- **2007**
- **2008**
- **2010**
- **2012**
Adobe Imaging Foundation

- **Image processing system**
  - Apply filters to images
  - Do for pixels what OpenGL does for triangles

- **Support for heterogeneous hardware**
  - GPU, (multi-core) CPU

- **Framework**

- **Programmable language (Pixel Bender) for kernels**
### History of Heterogeneous Computing at Adobe

<table>
<thead>
<tr>
<th>Year</th>
<th>Product/Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-2005</td>
<td>Point product support for OpenGL</td>
<td>AIF Project Formed</td>
</tr>
<tr>
<td>2005</td>
<td>CS3 ships with GPU-accelerated features</td>
<td>CS4 ships with GPU-accelerated features</td>
</tr>
<tr>
<td>2007</td>
<td>CS4 ships with GPU-accelerated features</td>
<td>CS5 ships with GPU-accelerated features</td>
</tr>
<tr>
<td>2008</td>
<td>CS5 ships with GPU-accelerated features</td>
<td>CS6 ships with GPU-accelerated features</td>
</tr>
<tr>
<td>2010</td>
<td>CS6 ships with GPU-accelerated features</td>
<td>CS7 ships with GPU-accelerated features</td>
</tr>
<tr>
<td>2012</td>
<td>CS7 ships with GPU-accelerated features</td>
<td>CS8 ships with GPU-accelerated features</td>
</tr>
</tbody>
</table>
History of Heterogeneous Computing at Adobe

- AIF Project Formed: Pre-2005
- Point product support for OpenGL: 2005
- CS3 ships with GPU-accelerated features: 2007
- CS4 ships with GPU-accelerated features: 2008
- CS5 ships with GPU-accelerated features: 2010
- CS6 ships with OpenCL-accelerated features: 2012
Adobe ♥ OpenCL

- Compute API supported across vendors
- Programming model familiar to C programmers
- Demonstrated performance
- Same compute kernels on CPU and GPU!

- Adobe is now active member of OpenCL working group
  - Contributing Adobe’s experience and minds to continue OpenCL evolution
OpenCL in Photoshop CS6

Sarah Kong
Photoshop Engineering Manager
Adobe Systems Inc.
Blur Gallery Demo
Why OpenCL

- Only cross-platform GPGPU solution
- Advantages over OpenGL
  - Learning curves; data formats; debugging
- Increasing maturity and ubiquity
How did we do it?

• OpenCL kernel was naïve port of SSE2 function (scheduled with TBB on CPU)
• Broken into 2K x 2K blocks for GPU
Challenges

- **Need good candidate algorithms**
  - Bandwidth, compute, parallel
Challenges

- Need good candidate algorithms
  - Bandwidth, compute, parallel
- Need debugged C algorithm first
Challenges

• Need good candidate algorithms
  - Bandwidth, compute, parallel
• Need debugged C algorithm first
• GPU win over CPU variable and unpredictable
Challenges

- Need good candidate algorithms
  - Bandwidth, compute, parallel
- Need debugged C algorithm first
- GPU win over CPU variable and unpredictable
- Resource limits
  - Win/Mac Timeout issues on low end cards.
Challenges

- Need good candidate algorithms
  - Bandwidth, compute, parallel
- Need debugged C algorithm first
- GPU win over CPU variable and unpredictable
- Resource limits
  - Win/Mac Timeout issues on low end cards.
  - Out of memory issues on low end cards.
Challenges

• Need good candidate algorithms
  - Bandwidth, compute, parallel

• Need debugged C algorithm first

• GPU win over CPU variable and unpredictable

• Resource limits
  - Win/Mac Timeout issues on low end cards.
  - Out of memory issues on low end cards.
  - Used bandwidth test to enable OpenCL acceleration
    - Affected by power-gating on mobile GPU
Challenges

- Need good candidate algorithms
  - Bandwidth, compute, parallel

- Need debugged C algorithm first

- GPU win over CPU variable and unpredictable

- Resource limits
  - Win/Mac Timeout issues on low end cards.
  - Out of memory issues on low end cards.
  - Used bandwidth test to enable OpenCL acceleration
    - Affected by power-gating on mobile GPU

- Platform variation
  - Driver Issues
  - Various compiler issues
Performance Comparison

- Systems of standard configuration show 4-8x gain for typical use-cases
- Gains improve with Blur radii (results from MacBookPro running 10.7.4 listed below)
  - General application processing accounts for majority of time in smaller workloads

<table>
<thead>
<tr>
<th>Radius in Pixels (13 Megapixel image)</th>
<th>Radeon 6750m</th>
<th>Core i7 (2.3 GHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>4.7s</td>
<td>16.2s</td>
</tr>
<tr>
<td>100</td>
<td>7.0s</td>
<td>31.9s</td>
</tr>
<tr>
<td>150</td>
<td>9.4s</td>
<td>47.9s</td>
</tr>
</tbody>
</table>
Future Plans

- More OpenCL in future versions
- Investigate OpenCL for CPU
OpenCL in Premiere Pro CS

David McGavran
Premiere Pro Engineering Manager
Adobe Systems Inc.
Demo
Pipeline

Disk IO → CPU Processing → Upload → GPU Processing → Display/Export
Accelerated Effects

Intrinsics
- Adjustment layers
- Color space conversion
- Deinterlacing
- Compositing
- Blending modes
- Nested Sequences
- Multicam
- Time remapping

Transitions
- Additive dissolve
- Cross dissolve
- Dip to black
- Dip to white
- Film Dissolve
- Push

Effects
- Alpha adjust
- Basic 3D
- Black & white
- Brightness & contrast
- Color balance
- Color pass
- Color replace
- Crop
- Directional blur
- Drop Shadow
- Extract
- Fast blur
- Fast color corrector
- Feather edges
- Gamma correction
- Garbage matte
- Gaussian blur
- Horizontal flip
- Invert
- Luma corrector
- Luma curve
- Noise
- Proc amp
- RGB color corrector
- RGB curves
- Sharpen
- Three-way color corrector
- Timecode
- Tint
- Track matte
- Ultra keyer
- Vertical flip
- Video limiter
- Warp Stabilizer
Implementation

- Pipeline entirely floating point
- 10-bit display supported
- Subtree rendering for non-accelerated effects
- Draw with OpenGL interop
Filter Concatenation

• Entirely manual
• Host kernel for successive pointwise effects
• Read once, apply multiple filters, write once
• Major bandwidth savings
Future

• Increase set of supported effects
• Supporting third party effects & codecs
• GPU encoding & decoding
• Multiple GPU support
• GPU Scopes