The OpenVX Standard for Portable, Efficient Vision Processing

May 2017
Khronos Connects Software to Silicon

Industry Consortium creating OPEN STANDARD APIs for hardware acceleration
Any company is welcome - one company one vote

ROYALTY-FREE specifications
State-of-the art IP framework protects
members AND the standards

Software

Conformance Tests and Adopters Programs for specification integrity
and cross-vendor portability

Silicon

Low-level silicon APIs needed on
every platform: graphics, parallel
compute, vision, neural networks,
augmented and virtual reality...

International, non-profit organization
Membership and Adopters fees cover
operating and engineering expenses

Strong industry momentum
100s of man years invested by industry experts

Well over a BILLION people use Khronos APIs Every Day...
Khronos Open Standards

3D for the Web
- VR/AR and games in-browser
- Efficiently delivering runtime 3D assets

Vision and Neural Networks
- Tracking and odometry
- Scene analysis/understanding
- Neural Network inferencing

Real-time 2D/3D
- Virtual and Augmented Reality displays
- Cross-platform gaming and UI
- CAD and Product Design

Parallel Computation
- Machine Learning acceleration
- Embedded vision processing
- High Performance Computing (HPC)
OpenVX - Low Power Vision Acceleration

- Targeted at vision acceleration in real-time, mobile and embedded platforms
  - High performance AND low power consumption are key

- Higher abstraction than OpenCL for performance portability across diverse architectures
  - Multi-core CPUs, GPUs, DSPs and DSP arrays, ISPs, FPGAs, Dedicated hardware...

- Extends portable vision acceleration to very low power domains
  - Doesn’t require high-power CPU/GPU Complex or OpenCL precision
OpenVX Evolution

**Conformant Implementations**
- AMD
- Cadence
- CEVA
- E3LVEES
- Imagination
- Texas Instruments
- Intel
- NVIDIA
- Socionext
- Synopsys
- VeriSilicon

**OpenVX 1.0**
Spec released October 2014

**OpenVX 1.1**
Spec released May 2016

**New Functionality**
- Expanded Nodes Functionality
- Enhanced Graph Framework

**AMD OpenVX Tools**
- Open source, highly optimized for x86 CPU and OpenCL for GPU
- "Graph Optimizer" looks at entire processing pipeline and removes, replaces, merges functions to improve performance and bandwidth
- Scripting for rapid prototyping, without re-compiling, at production performance levels

**OpenVX 1.2**
Spec released May 2017

**OpenVX Roadmap**

**New Functionality Under Discussion**
- NNEF Import
- Programmable user kernels on accelerator

**Extensions**
- Neural Network Acceleration
- Graph Save and Restore
- 16-bit image operation

**Safety Critical**
OpenVX 1.1 SC for safety-certifiable systems
OpenCL - Low-level Parallel Programming

- Low level programming of heterogeneous parallel compute resources
  - One code tree can be executed on CPUs, GPUs, DSPs and FPGA
- OpenCL C language to write kernel programs to execute on any compute device
  - Platform Layer API - to query, select and initialize compute devices
  - Runtime API - to build and execute kernels programs on multiple devices
- New in OpenCL 2.2 - OpenCL C++ kernel language - a static subset of C++14
  - Adaptable and elegant sharable code - great for building libraries
  - Templates enable meta-programming for highly adaptive software
  - Lambdas used to implement nested/dynamic parallelism
Layered Vision Processing Ecosystem

Implementers may use OpenCL to *implement* OpenVX nodes on programmable processors.

And then developers can use OpenVX to enable a developer to easily *connect* those nodes into a graph.

OpenVX enables the graph to be *extended* to include hardware architectures that don’t support programmable APIs.

The OpenVX graph enables implementers to *optimize* execution across diverse hardware architectures and drive to lower power implementations.
<table>
<thead>
<tr>
<th>How OpenVX Compares to Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Governance</strong></td>
</tr>
<tr>
<td>Open standard API designed to be implemented and shipped by IHVs</td>
</tr>
<tr>
<td><strong>Programming Model</strong></td>
</tr>
<tr>
<td>Graph defined with C API and then compiled for run-time execution</td>
</tr>
<tr>
<td><strong>Built-in Vision Functionality</strong></td>
</tr>
<tr>
<td>Small but growing set of popular functions</td>
</tr>
<tr>
<td><strong>Target Hardware</strong></td>
</tr>
<tr>
<td>Any combination of processors or non-programmable hardware</td>
</tr>
<tr>
<td><strong>Optimization Opportunities</strong></td>
</tr>
<tr>
<td>Pre-declared graph enables significant optimizations</td>
</tr>
<tr>
<td><strong>Conformance</strong></td>
</tr>
<tr>
<td>Implementations must pass conformance to use trademark</td>
</tr>
<tr>
<td><strong>Consistency</strong></td>
</tr>
<tr>
<td>All core functions must be available in conformant implementations</td>
</tr>
</tbody>
</table>
Safety Critical APIs

OpenGL SC 1.0 - 2005
Fixed function graphics subset

OpenGL ES 1.0 - 2003
Fixed function graphics

OpenGL SC 2.0 - April 2016
Shader programmable pipeline subset

OpenGL ES 2.0 - 2007
Shader programmable pipeline

OpenVX SC 1.1 Released 1st May 2017
Restricted “deployment” implementation executes on the target hardware by reading the binary format and executing the pre-compiled graphs

Experience and Guidelines

Khronos SCAP
‘Safety Critical Advisory Panel’ Guidelines for designing APIs that ease system certification. Open to Khronos member AND industry experts. If interested to join contact ntrevett@nvidia.com

Vulkan SC being discussed
Small driver size
Advanced functionality
Graphics and compute

New Generation APIs for safety certifiable vision, graphics and compute
e.g. ISO 26262 and DO-178B/C

© Copyright Khronos Group 2017 - Page 10