KHRONOS API ECOSYSTEM

Neil Trevett, GTC, 8th May 2017
Khronos is an International Industry Consortium of over 100 companies creating royalty-free, open standard APIs to enable software to access hardware acceleration for 3D graphics, Virtual and Augmented Reality, Parallel Computing, Neural Networks and Vision Processing.
Khronos Standards Ecosystem

**3D for the Web**
- Real-time apps and games in-browser
- Efficiently delivering runtime 3D assets

**Collada**

**WebGL**

**GLTF**

**3D Portability**

**Real-time 2D/3D**
- Cross-platform gaming and UI
- VR and AR Displays
- CAD and Product Design
- Safety-critical displays

**OpenGL ES**

**Vulkan**

**OpenGL SC**

**OpenVG**

**Parallel Computation**
- Machine Learning acceleration
- Embedded vision processing
- High Performance Computing (HPC)

**VR, Vision, Neural Networks**
- VR/AR system portability
- Tracking and odometry
- Scene analysis/understanding
- Neural Network inferencing

**NNEF**

**OpenVX**

**OpenXR**

**3D Portability**

**SPIR**

**SYCL**

**OpenCL**

© Copyright Khronos Group 2017 - Page 3
Vulkan and New Generation 3D APIs

DirectX 12
Only Windows 10

Only Apple

Cross Platform
Vulkan

SteamOS
Ubuntu
Red Hat
Tizen
Nintendo Switch
Android 7

© Copyright Khronos Group 2017 - Page 4
Vulkan 1.0 provides access to OpenGL ES 3.1 / OpenGL 4.X-class GPU functionality but with increased performance and flexibility.
Vulkan Games and Game Engines

Dota 2 on Vulkan port of Source 2

‘ProtoStar’ demo on Vulkan port of Unreal Engine 4

Talos Principle on Vulkan port of Serious Engine

Doom’s Vulkan patch is a PC performance game-changer

DOOM on Vulkan port of id Tech 6

Vulkan support since V1.8

Vulkan support in Unity 5.6

Vulkan support in V5.4 - Late May 2017
Vulkan Momentum

Games Studios publicly confirming that work is ongoing on Vulkan Titles

In first 12 months:
#Vulkan Games on PC = 11
In first 18 months:
#DX12 Games on PC = 19


All Major GPU Companies shipping Vulkan Drivers - for Desktop and Mobile Platforms

Mobile, Embedded and Console Platforms Supporting Vulkan

Android 7.0  Nintendo Switch  Android TV  Embedded Linux
Latest Vulkan Functionality

Vulkan 1.0.42 with new extension sets

Explicit Building Blocks for VR

Multiview
Efficiently render geometry to multiple surfaces
Each with its own viewing parameters
E.g. render stereo pairs or environment maps

Resource Sharing
Share memory and synchronization primitives
Works across process and instance boundaries

Descriptor Updates
Update resource references between
draw or compute dispatch calls
Efficiently repeatedly update fixed set of resources

Explicit Building Blocks for Multi-GPU

Works with NVIDIA SLI and AMD Crossfire
Does NOT support dGPU/iGPU

“Device Group” is set of physical devices
Acts as single logical device

Access separate GPUs only for explicit control
Memory allocation and binding resources
Command Buffer Recording/Submission
Synchronization

Enables a variety of operating modes
AFR (alternate frame), SFR (Sequential frame)
VR SLI Stereo view rendering

Shipped 1.0.42 beta drivers on the day the specifications were released PLUS building block Vulkan extensions for VRWorks on Maxwell and Pascal
Khronos APIs for Virtual Reality

Khronos 3D APIs already powering VR rendering

- **OpenGL.** The standard for desktop VR today
- **OpenGL ES.** On millions of mobile VR devices
- **WebGL.** Powering WebVR in browsers
- **Vulkan.** The future of explicit, low-latency VR

But What About INPUT Hardware for VR..

- Device discovery
- Multiple sensor tracking
- Device Events
- Haptics
- Parameters for optics corrections etc. etc...

OpenXR!

Cross-Platform, portable Virtual Reality
OpenXR - Solving VR Fragmentation

Before OpenXR
VR Market Fragmentation

After OpenXR
Wide interoperability of VR apps and devices
OpenXR Working Group Members

Design work started in December 2016
Typically 12-18 months to develop a V1.0 specification
## OpenXR and VR Run-times

<table>
<thead>
<tr>
<th>Proprietary APIs</th>
<th>Proprietary Driver Interfaces</th>
<th>OpenXR Application Interface</th>
<th>OpenXR Device Layer</th>
<th>OpenXR Extensions</th>
</tr>
</thead>
</table>
| VR Run-time      |                               | Access to any OpenXR Application | Access to any OpenXR Device | Any successful standard encourages and enables healthy industry competition

OpenXR will not replace VR run-times - or outlaw existing interfaces

OpenXR will simply provide cross-vendor APIs that can be exposed by a runtime to access more apps and devices
OpenVX - Efficient Vision Acceleration

- Vision acceleration for 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 CUDA/OpenCL precision

OpenVX provides ‘Performance Portable’ vision processing. Write once, run (efficiently) everywhere
OpenVX - Graph-Level Abstraction

- OpenVX developers express a graph of image operations (‘Nodes’)
  - Using a C API
- Nodes can be executed on any hardware or processor coded in any language
  - Implementers can optimize under the high-level graph abstraction
- Graphs are the key to run-time power and performance optimizations

OpenVX Graph

Feature Extraction Example Graph
Simple Edge Detector in OpenVX

vx_image input = vxCreateImage(1920, 1080);
vx_image output = vxCreateImage(0, 0);
vx_image horiz = vxCreateVirtualImage();
vx_image vert = vxCreateVirtualImage();
vx_image mag = vxCreateVirtualImage();

vx_graph g = vxCreateGraph();
vxSobel3x3Node(g, input, horiz, vert);
vxMagnitudeNode(g, horiz, vert, mag);
vxThresholdNode(g, mag, THRESH, output);

status = vxVerifyGraph(g);
status = vxProcessGraph(g);
OpenVX - Efficiency AND Rapid Development

- Graphs enable automatic optimizations
  - Especially tiling and kernel fusion
- Automatic selection of kernels
  - Can be optimized by data type at graph verification time
- User does NOT need to handle:
  - Hardware features: like scatter-gather, custom ISPs, hardware blocks
  - Custom kernels: intrinsics, assembly code
  - Custom data movement: tiling, local memory management, DMA, memory hierarchy
- The OpenVX framework does all of this for you
## OpenVX Evolution

<table>
<thead>
<tr>
<th>Version</th>
<th>Spec Released</th>
<th>New Functionality</th>
<th>Extensions</th>
<th>Safety Critical</th>
<th>OpenVX Roadmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenVX 1.0</td>
<td>October 2014</td>
<td>Conformant Implementations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenVX 1.1</td>
<td>May 2016</td>
<td>New Functionality</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenVX 1.2</td>
<td>May 2017</td>
<td>New Functionality</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Conformant Implementations
- AMD
- cadence
- CEVA
- imagination
- Texas Instruments
- 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
- NNEF Import (Neural Net Exchange Format)
- Programmable user kernels with accelerator offload

### Under Discussion
- NNEF Import
- (Neural Net Exchange Format)
- Programmable user kernels with accelerator offload

### New Functionality
- Conditional node execution
- Feature detection
- Classification operators
- Expanded imaging operations
- Neural Network Acceleration
- Graph Save and Restore
- 16-bit image operation
- OpenVX 1.1 SC for safety-certifiable systems

### Extensions
- OpenVX 1.2 Spec released May 2017
New OpenVX 1.2 Functions

- **Feature detection**: find features useful for object detection and recognition
  - Histogram of gradients - HOG
  - Local binary patterns - LBP
- **Classification**: detect and recognize objects in an image based on a set of features
  - Import a classifier model trained offline
  - Classify objects based on a set of input features
- **Image Processing**: transform an image
  - Generalized nonlinear filter: Dilate, erode, median with arbitrary kernel shapes
  - Non maximum suppression: Find local maximum values in an image
  - Edge-preserving noise reduction
- **Conditional execution & node predication**
  - Selectively execute portions of a graph based on a true/false predicate
- **Many, many minor improvements**
- **New Extensions**
  - **Import/export**: compile a graph; save and run later
  - **16-bit support**: signed 16-bit image data
  - **Neural networks**: Layers are represented as OpenVX nodes

\[
\text{If A then } S \leftarrow B \text{ else } S \leftarrow C
\]
## OpenCL 2.2 - Top to Bottom C++

**Single Source C++ Programming**
Full support for features in C++14-based Kernel Language

**API and Language Specs**
Brings C++14-based Kernel Language into core specification

**Portable Kernel Intermediate Language**
Support for C++14-based kernel language e.g. constructors/destructors

<table>
<thead>
<tr>
<th>Time</th>
<th>OpenCL Version</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec08</td>
<td>OpenCL 1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18 months</td>
<td></td>
</tr>
<tr>
<td>Jun10</td>
<td>OpenCL 1.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18 months</td>
<td></td>
</tr>
<tr>
<td>Nov11</td>
<td>OpenCL 1.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24 months</td>
<td></td>
</tr>
<tr>
<td>Nov13</td>
<td>OpenCL 2.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24 months</td>
<td></td>
</tr>
<tr>
<td>Nov15</td>
<td>OpenCL 2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18 months</td>
<td></td>
</tr>
<tr>
<td>May17</td>
<td>OpenCL 2.2</td>
<td></td>
</tr>
</tbody>
</table>

- SYCL
- SPIR-V

- 3-component vectors
- Additional image formats
- Multiple hosts and devices
- Buffer region operations
- Enhanced event-driven execution
- Additional OpenCL C built-ins
- Improved OpenGL data/event interop

- Device partitioning
- Separate compilation and linking
- Enhanced image support
- Built-in kernels / custom devices
- Enhanced DX and OpenGL Interop

- Shared Virtual Memory
- On-device dispatch
- Generic Address Space
- Enhanced Image Support
- C11 Atomics
- Pipes
- Android ICD

- SPIR-V in Core
- Subgroups into core
- Subgroup query operations
- clCloneKernel
- Low-latency device
  - timer queries

- OpenCL C++ Kernel Language
  - SPIR-V 1.2 with C++ support
  - SYCL 2.2 for single source C++
OpenCL Evolution Discussions

Single source C++ programming. Great for supporting C++ apps, libraries and frameworks

2011

OpenCL 1.2

SYCL 1.2
C++11 Single source programming

2015

OpenCL 2.1
SPIR-V in Core

SYCL 2.2
C++14 Single source programming

2017

OpenCL 2.2
C++ in Core

SYCL 2.2

Industry working to bring Heterogeneous compute to standard ISO C++

C++17 Parallel STL hosted by Khronos
Executors - for scheduling work
“Managed pointers” or “channels” - for sharing data
Hoping to target C++ 20 but timescales are tight

OpenCL 1.2++?

Focus on embedded imaging, vision and inferencing
Make FP32 optional for DSPs and general power efficiency

‘OpenCL-V’
Converge Vulkan and OpenCL
‘OpenCL-V’ - OpenCL and Vulkan Convergence

- Converge over time with OpenCL roadmap with Vulkan API and run-time
  - Expand Vulkan to include OpenCL-class compute
  - Support more processor types, e.g. DSPs and FPGAs (graphics optional)

- Layered ecosystem for backwards-compatibility and market flexibility
  - Feature sets for target market agility

- Single runtime stack for graphics and compute
  - Streamline development, adoption and deployment for the entire industry

Applications

Vendor-supplied and open source middleware

OpenCL 1.X/2.X Compatibility

Math Libraries

Language Front-ends

Tool Layers

Vulkan API

Thin, explicit Vulkan run-time with rigorous memory/execution model. Low-latency and predictable

Feature Sets can be enabled for particular target markets

Vendor-supplied and open source middleware

SYCL

Installable tool & validation layers

Dialable types and precision

Real-time Pre-emption and QoS scheduling

Explicit Asynch DMA

Self-synchronized, self-scheduled graphs

Stream Processing
NVIDIA and Khronos Standards

Actively helping to define and drive OpenXR to enable and complement VRWorks

Shipping extended OpenCL 1.2 on Linux and Windows desktop. Actively exploring roadmap convergence with Vulkan

First to ship production OpenVX 1.0. Extended OpenVX used in VisionWorks for Tegra/Jetson

NVIDIA shipping latest specification, driving experimental extensions (e.g. for VR), adding Vulkan support to NSIGHT
GPU Portability – Call For Participation

‘WebGL Next’
- Lift ‘Portability API’ to JavaScript and use in WebAssembly native code
- Nextgen graphics and GPU compute for the Web

API Overlap Analysis

No single API on all systems

Vulkan Portability Solution
C/C++ Portability API Library
+ Shading Language tools
All open source

Use Feature Sets to remove non-portable functionality

Use Extensions to add functionality e.g. security and robustness for the Web

Portable ‘Vulkan Subset’ API Specification

Open source compilers/translators for shading and intermediate languages

Vulkan is non-proprietary and is already designed to be portable

A Portability Solution needs to address APIs and shading languages

Vulkan

‘WebGL’

No single API on all systems
WebGL, WebVR and glTF Meetup at GTC

- **Audio**: MP3
- **Video**: H.264
- **Images**: JPEG
- **3D**: glTF

**glTF** - OpenGL Transmission Format
Efficient transmission of 3D assets

Now glTF 2.0 has PBR!
- Cool, portable materials
- Rendering API independence
- Coming to Microsoft Office!

Laugh Engine running on Vulkan
https://github.com/jian-ru/laugh_engine
Thank You!

• NVIDIA resources
  - https://developer.nvidia.com/opencl

• Any company or organization is welcome to join Khronos for a voice and a vote in any of these standards - membership starts at $3,500
  - www.khronos.org

• Neil Trevett
  - ntrevett@nvidia.com | @neilt3d