Vision and Inferencing Acceleration
OpenVX Performance, Portability and Memory Footprint Advantages
14th October 2020
Welcome!

This slide deck will be posted at www.khronos.org today

The webinar will be recorded and made freely available

Please take a moment to answer the polls at the end of the webinar

Ask any questions via the Q&A Panel at any time

We will answer questions at the end of the webinar
Introduction to Khronos and OpenVX
Neil Trevett
Khronos President

Cadence and OpenVX
Frank Brill
Cadence Systems
Design Engineering Director

AMD and OpenVX
Mike Schmit
AMD
Director of Software System
Design ML Computer Vision

Using OpenVX
Kiriti Nagesh Gowda
OpenVX Group Chair
Khronos Connects Software to Silicon

Open interoperability standards to enable software to effectively harness the power of 3D and multiprocessor acceleration

3D graphics, XR, parallel programming, vision acceleration and machine learning

Non-profit, member-driven standards-defining industry consortium

Open to any interested company

All Khronos standards are royalty-free

Well-defined IP Framework protects participant’s intellectual property

Founded in 2000

>150 Members ~ 40% US, 30% Europe, 30% Asia
Khronos Active Initiatives

3D Graphics
Desktop, Mobile, Web
Embedded and Safety Critical

3D Assets
Authoring and Delivery

Portable XR
Augmented and Virtual Reality

Parallel Computation
Vision, Inferencing, Machine Learning

3D Graphics
Vulkan®
OpenGL®
GLSL®
EGL®
WebGL®
Khronos®

3D Assets
GLTF
COLLADA®
3D Commerce™

Portable XR
OpenXR™

Parallel Computation
OpenVX™
OpenCL™
NNEF™
SYCL™
SPIR™
The Origin of OpenVX

**Engines and Applications**

**Vulkan**

3D Graphics API Driver

**GPU**

**Driver Model**

An open API standard enables multiple silicon vendors to ship drivers with their silicon.

Silicon vendors can aggressively optimize drivers for their own silicon architecture.

OpenVX is the industry’s only API standard enabling portable access to vendor-optimized vision drivers.

**Engines and Applications**

**OpenVX**

Vision API Driver

**GPU**

**DSP**

**HW**

**CPU**

**High-level Abstraction**

3D graphics is always accelerated by a GPU - so a low-level GPU-centric API still provides cross-vendor portability.

Vision processing can be accelerated by a wide variety of hardware architectures.

OpenVX needs a higher-level graph abstraction to enable optimized cross-vendor drivers.

**Vision Processing Graph**
OpenVX Cross-Vendor Vision and Inferencing

High-level graph-based abstraction for portable, efficient vision processing

- Optimized OpenVX drivers created, optimized and shipped by processor vendors
- Implementable on almost any hardware or processor with performance portability
- Graph can contain vision processing and NN nodes for global optimization
- Run-time graph execution need very little host CPU interaction

OpenVX Graph

Vision Node

CNN Nodes

Vision Node

Downstream Application Processing

Native Camera Control

Vision Node

Open Source Convertors

ONNX

TensorFlow

TensorFlowLite

Caffe

Caffe2

Stable Specification

NNEF Import converts a trained Neural Network into OpenVX Graph
Layer are represented as OpenVX nodes

https://github.com/KhronosGroup/NNEF-Tools

Open Source Projects

Source

https://github.com/KhronosGroup/NNEF-Tools

Stable Specification

NNEF Import converts a trained Neural Network into OpenVX Graph
Layers are represented as OpenVX nodes

Open Source Projects
OpenVX Efficiency through Graphs..

**Graph Scheduling**
- Split graph execution across the whole system: CPU / GPU / dedicated HW
- Faster execution or lower power consumption

**Memory Management**
- Reuse pre-allocated memory for multiple intermediate data
- Less allocation overhead, more memory for other applications

**Kernel Fusion**
- Replace a sub-graph with a single faster node
- Better memory locality, less kernel launch overhead

**Data Tiling**
- Execute a sub-graph at tile granularity instead of image granularity
- Better use of data cache and local memory

Performance comparable to hand-optimized, non-portable code
Real, complex applications on real-world hardware
Much lower development effort and higher portability than hand-optimized code
## OpenVX and OpenCV are Complementary

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Community driven open source library</th>
<th>API drivers implemented, optimized and shipped by hardware vendors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conformance</td>
<td>Extensive OpenCV Test Suite but no formal Conformance program</td>
<td>Implementations pass Khronos Conformance Testing to use trademark and ensure cross vendor consistency</td>
</tr>
<tr>
<td>Scope</td>
<td>100s of imaging and vision functions, Multiple camera APIs/interfaces</td>
<td>Tight focus on dozens of core hardware accelerated functions plus extensions and accelerated custom nodes. Uses external camera drivers</td>
</tr>
<tr>
<td>Inferencing</td>
<td>Deep Neural Network module to construct networks from layers for forward pass computations only. Import from ONNX, TensorFlow, Torch, Caffe</td>
<td>Neural Network layers and operations represented directly in the OpenVX Graph. NNEF direct import with multiple convertors</td>
</tr>
<tr>
<td>Acceleration</td>
<td>OpenCV 3.0 Transparent API (or T-API) enables function offload to OpenCL devices</td>
<td>Implementation free to use any underlying API such as OpenCL. OpenCL for accelerated Custom Nodes</td>
</tr>
<tr>
<td>Efficiency</td>
<td>OpenCV 4.0 G-API graph model for some filters, arithmetic/binary operations, and well-defined geometrical transformations</td>
<td>Graph-based execution of all Nodes. Optimizable computation and data transfer</td>
</tr>
<tr>
<td>IP Protection</td>
<td>None. Source code licensed under BSD. Some modules require royalties/licensing</td>
<td>Protected under Khronos IP Framework - Khronos members agree not to assert patents against API when used in Conformant implementations</td>
</tr>
</tbody>
</table>
OpenVX Extensibility

OpenVX core specification defines market-targeted feature sets
Baseline Graph Infrastructure (enables other Feature Sets)
Default Vision Functions
Enhanced Vision Functions
Neural Network Inferencing (including tensor objects)
NNEF Kernel import (including tensor objects)
Binary Images
Safety Critical (reduced features and graph import for easier safety certification)

OpenVX is Extensible
Fully accelerated custom nodes can be integrated into the OpenVX graph with OpenCL interop

OpenVX user-kernels can access command queue and cl_mem objects to asynchronously schedule OpenCL kernel execution

OpenVX data objects

OpenCL Command Queue

Application

Copy or export cl_mem buffers into OpenVX data objects

Map or copy OpenVX data objects into cl_mem buffers

OpenCL Runtime

cl_mem buffers

OpenVX/OpenCL Interop

Fully asynchronous host-device operations during data exchange
OpenVX Industry Deployment

OpenVX Working Group

OpenVX Specifications

Open source
Conformance Test Suite

Adopters
Program (fee)

Open source sample implementation

Vendors optimize and ship drivers for their platform
Full list of conformant products here: https://www.khronos.org/conformance/adopters/conformant-products/openvx

Conformant Implementations can use OpenVX logo

Porting and optimization

Developers can freely use any OpenVX Implementation

This work is licensed under a Creative Commons Attribution 4.0 International License
Using OpenVX on Cadence Vision DSPs
Efficient, Portable Code with the Cadence Application Programming Kit (APK) v4.1
Frank Brill, Design Engineering Director
14 October 2020
Vision DSPs + DNA for Computer Vision and AI Applications

#1 VISION + AI IP

VISION + AI CHOICE

50+ ECOSYSTEM PARTNERS

200+ VISION + AI SOFTWARE IN USE

80+ VISION + AI LICENSEES WORLDWIDE

Vision + DNA FAMILY Licensees

1.5B+ VISION + DNA SHIPPING WORLDWIDE ANNUALLY
Vision DSPs for Imaging, Vision and AI Applications

Scalable Performance
- 512 MACs, 64-way SIMD
- 512-bit memory, 2TOP/sec
- SuperGather - 32 accesses/cycle

Multiple Markets
- Handsets, ADAS
- AI, Robotics
- Surveillance, AR/VR

Licensable Vision DSP IP
- 3 of Top 5 Application Processors
- #1 Volume shipping Vision DSP

Vision Software Ecosystem
- Optimized SLAM, OpenCL, Halide, OpenVX, Neural Network (AI), CV, Libraries, and 50+ software partners...

FuSa Safety Documentation
- ISO 26262 Ready
Vision P6/Q7 Architecture

- VLIW SIMD Architecture for parallel processing
- 512-bit SIMD
- 5 VLIW Slots
- Dual Load Store, 1024-bit Memory interface
- Data Types: Fixed point (8/16/32 bit), FP16, FP32
- Scatter Gather
- Advanced iDMA
- AXi4 interface, 128/256-bit iDMA interface
- Vision Q7: Accelerators for SLAM
- AI Acceleration
  - Vision Q7: 512 8-bit MAC, 128 16-bit MAC
  - Vision P6: 256 8-bit MAC, 64 16-bit MAC
- FUSA (ISO 26262) ready IP

*Compared to previous-generation Vision Q6 DSP

Tile-based processing

- Most access is to local (fast) memory
- Data transfer (DMA) overlaps with compute
- Full intermediate images aren’t allocated

Coding this by hand is
- time consuming
- not portable
Cadence OpenVX block diagram (host + DSP)

Host
(e.g., Xtensa controller, x86, or Other CPU)

User Program

OpenVX API

vxVerifyGraph
vxProcessGraph

Graph Mapper

binary script

OS (e.g. Linux)

Host Runtime

IPC Driver

Vision DSP
(e.g. P6/Q7)

DSP Runtime

DMA Library

XI Library

Shared memory

Legend

APK Component
Cadence OpenVX case study

- Application: background subtraction for video security

Graph Speed-up

Higher memory access penalty → greater graph benefits
Computer Vision & Machine Learning with OpenVX™

Kiriti Nagesh Gowda
Staff Engineer | AMD
OpenVX™ Chair | The Khronos Group
Agenda

- OpenVX Overview
- OpenVX 1.3 - Highlights & New Features
- Common Questions About OpenVX
- Conformant OpenVX Implementations
- OpenVX Sample Implementation
- OpenVX for Raspberry Pi
- An OpenVX Cross-Platform Application: Case Study
- Conclusion
OpenVX Overview

OpenVX™ is an open, royalty-free API standard for a cross-platform acceleration of computer vision applications.

What is an API?

Application Programming Interface
OpenVX Overview

Why APIs are important

- Building Blocks
- Speeds up development
- Portability
- Innovation

OpenVX™ enables portable, performance, and power-optimized computer vision processing, especially important in embedded and real-time use cases.

OpenVX™ is unique in being the only vision API shipped as an optimized driver.
OpenVX Overview

OpenVX™ Graph Framework

- OpenVX allows graph-level processing **optimizations**, which lets implementations to **fuse nodes** when possible to achieve better overall **performance**
- The graph also allows for auto graph-level **memory optimizations** to achieve a **low memory footprint**
- OpenVX graph-optimized workloads can be **deployed** on a **wide range of computer hardware**, including small embedded CPUs, ASICs, APUs, discrete GPUs, and heterogeneous servers
OpenVX™ 1.3 – Released on October 22nd, 2019

• Enable deployment flexibility while avoiding fragmentation, OpenVX 1.3 defines several feature sets that are targeted at common embedded use cases

• Hardware vendors can include one or more complete feature sets in their implementations to meet the needs of their customers and be fully conformant

• The flexibility of OpenVX enables deployment on a diverse range of accelerator architectures, and feature sets are expected to dramatically increase the breadth and diversity of available OpenVX implementations
OpenVX 1.3 - Highlights & New Features

The defined OpenVX 1.3 feature sets include:

- **Graph Infrastructure** - baseline for other Feature Sets
- **Vision** - core vision functionality
- **Enhanced Vision** - functions introduced in OpenVX 1.2
- **Neural Network Inferencing** - including tensor objects
- **NNEF Kernel import** - including tensor objects
- **Binary Images** – one bit images
- **Safety Critical** - reduced features to enable easier safety certification
Common Questions About OpenVX

**Question:** Is OpenVX an Open-Sourced Library?

- Callable API implemented, optimized OpenVX drivers are created, optimized, and shipped by processor vendors

**Question:** Must I pay royalties and licensing fee to use OpenVX?

- Protected under Khronos IP Framework - Khronos members agree not to assert patents against API when used in Conformant implementations

**Question:** Must I be a Khronos member to use OpenVX?

- Khronos members and non-members develop conformant implementations to be used by all
Common Questions About OpenVX

**Question:** Is OpenVX Functions limiting?
- Tight focus on dozens of core hardware accelerated functions plus extensions and accelerated custom nodes
- Users can create custom nodes and vendors can create custom extensions, with some cost in terms of portability

**Question:** Are Implementations different in functionality?
- Core API provides identical functionality across platforms due to strict conformance testing
- Implementations must pass Khronos Conformance Test Suite to use trademark
- Conformance does not extend to vendor extensions and custom nodes created by users
- Popular vendor extensions adopted as Khronos extensions with conformance testing
Conformant OpenVX Implementations

• Conformant Implementations must pass exhaustive conformance test suite
• Hardware vendors provide optimized OpenVX drivers, architected to get the best performance from their silicon architecture and ready for developers to use

Conformant Implementations of OpenVX from the following vendors:
OpenVX Sample Implementation

Open-Sourced OpenVX Sample Implementation available on GitHub - https://github.com/KhronosGroup/OpenVX-sample-impl

The purpose of this software package is to provide a sample implementation of the OpenVX 1.3 Specification that passes the conformance test. It is NOT intended to be a reference implementation.

Optimized OpenVX libraries available from vendor implementations

IS:
• passing OpenVX 1.3 conformance tests

IS NOT:
• a reference implementation • optimized • production ready
OpenVX for Raspberry Pi

July 2020

• The Khronos Group and the Raspberry Pi Foundation have worked together to implement an open-source implementation of OpenVX™ 1.3, which passes the conformance on Raspberry Pi
• The open-source implementation passes the Vision, Enhanced Vision, Neural Net, & NNEF Kernel Import Conformance Profiles specified in OpenVX 1.3 on Raspberry Pi
• The Implementation is NEON optimized

Conformant hardware
• Raspberry Pi 3 Model B Rev 1.2
• Raspberry Pi 4 Model B Rev 1.2
OpenVX for Raspberry Pi

“Raspberry Pi is excited to bring the Khronos OpenVX 1.3 API to our line of single-board computers. Many of the most exciting commercial and hobbyist applications of our products involve computer vision, and we hope that the availability of OpenVX will help lower barriers to entry for newcomers to the field.”

Eben Upton
Chief Executive Raspberry Pi Trading
OpenVX - NNEF Import Conformance Feature Set

• Provides a minimum set of functions to import and execute neural networks described in the NNEF standard format.

• Applications using this feature set will use the `vxImportKernelFromURL` function to import an NNEF file at the location of the URL to create an OpenVX kernel representing the neural network.

• Sample Implementation Available on GitHub
An OpenVX Cross-Platform Application: A Case Study

Open-Source OpenVX Samples

Open-source OpenVX sample applications, to use with any conformant implementation of OpenVX available on GitHub
An OpenVX Cross-Platform Application: A Case Study

SkinTone Detector Sample

# extract R,G,B channels and compute R-G and R-B
node org.khronos.openvx.channel_extract input !CHANNEL_R R   # extract R channel
node org.khronos.openvx.channel_extract input !CHANNEL_G G   # extract G channel
node org.khronos.openvx.channel_extract input !CHANNEL_B B   # extract B channel
node org.khronos.openvx.subtract R   G   !SATURATE RmG    # compute R-G
node org.khronos.openvx.subtract R   B   !SATURATE RmB    # compute R-B

# compute threshold
node org.khronos.openvx.threshold R   thr95 R95          # compute R > 95
node org.khronos.openvx.threshold G   thr40 G40         # compute G > 40
node org.khronos.openvx.threshold B   thr20 B20         # compute B > 20
node org.khronos.openvx.threshold RmG thr15 RmG15        # compute RmG > 15
node org.khronos.openvx.threshold RmB thr0  RmB0         # compute RmB > 0

# aggregate all thresholded values to produce SKIN pixels
node org.khronos.openvx.and R95   G40   and1             # compute R95 & G40
node org.khronos.openvx.and and1  B20   and2           # compute B20 & and1
node org.khronos.openvx.and RmG15 RmB0  and3      # compute RmG15 & RmB0
node org.khronos.openvx.and and2 and3 output          # compute and2 & and3 as output
An OpenVX Cross-Platform Application: A Case Study

SkinTone Detector Sample - On Raspberry Pi 4 Model B Rev 1.2

* using open-source OpenVX Raspberry Pi Implementation for OpenVX Libraries

© 2020 The Khronos Group
An OpenVX Cross-Platform Application: A Case Study

SkinTone Detector Sample – On X86 Processor Windows

* using AMDs open-sourced MIVisionX for OpenVX Libraries
An OpenVX Cross-Platform Application: A Case Study

SkinTone Detector Sample – On X86 Processor Linux

* using Khronos OpenVX open-sourced Sample Implementation for OpenVX Libraries
An OpenVX Cross-Platform Application: A Case Study

SkinTone Detector Sample – *On MacOS*

* using AMDs open-sourced MIVisionX for OpenVX Libraries
An OpenVX Cross-Platform Application: A Case Study

SkinTone Detector Sample – Performance

![Performance Chart](chart.png)
An OpenVX Cross-Platform Application: A Case Study

SkinTone Detector Sample – Memory Footprint

![Bar chart showing memory footprints of OpenVX Graph Unoptimized and OpenVX Graph Optimized]
Conclusion

- OpenVX is unique in being the only vision API shipped as an optimized driver.
- OpenVX delivers performance comparable to hand-optimized, non-portable code.
- **Acceleration** on a wide range of vision hardware architectures.
- OpenVX provides a high-level **Graph-based abstraction**:
  - Enables Graph-level optimizations.
  - Can be implemented on almost any hardware or processor.
- **Portable, Efficient Vision Processing!**
Acknowledgement

Thanks To

- Mike Schmit - Director of Software Engineering, AMD
- AMDs MIVisionX Team
- OpenVX Working Group
- Neil Trevett – President, The Khronos Group
- Khronos Team
Resource Slide

Sample Implementation:
https://github.com/KhronosGroup/OpenVX-sample-impl

Sample Applications:
https://github.com/KhronosGroup/openvx-samples

Tutorial Material:
https://github.com/rgiduthuri/openvx_tutorial

OpenVX Programming Guide:
https://www.elsevier.com/books/openvx-programming-guide/brill/978-0-12-816425-9

Conformant Implementations
https://www.khronos.org/conformance/adopters/conformant-products/openvx

Khronos OpenVX API Registry
https://www.khronos.org/registry/OpenVX/

OpenVX for Raspberry Pi

AMD ROCm MIVisionX - OpenVX
https://gpuopen-professionalcompute-libraries.github.io/MIVisionX/
Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED ‘AS IS.’ AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

© 2020 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD Arrow logo, EPYC, Radeon, ROCm and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.
Khronos for Global Industry Collaboration

Khronos membership is open to any company
Influence the design and direction of key open standards that will drive your business
Accelerate time-to-market with early access to specification drafts
Provide industry thought leadership and gain insights into industry trends and directions
Benefit from Adopter discounts

www.khronos.org/members/