Request for Proposals

OpenVX Implementation on the Raspberry Pi Platform

November 2018

Notice

Trademarks

1 BACKGROUND

OpenVX is a state-of-the-art open API standard for accelerating applications using computer vision and machine learning. While there are numerous conformant OpenVX implementations on a wide variety of hardware, there is no open-source implementation of OpenVX on a widely-available, low-cost embedded platform for evaluation and teaching purposes. The Khronos Group seeks to promote industry and community involvement in the use and evolution of the OpenVX standard by providing an optimized and easily accessible accelerated implementation in open source.

Consequently, the goal of this project is to procure an implementation of OpenVX 1.2.1, optimized for the popular Raspberry Pi platform. This source code for this project will be uploaded by Khronos to GitHub.

2 REQUIREMENTS

2.1 GENERAL

The project will deliver a fully conformant implementation of the OpenVX 1.2.1 standard that is optimized for the Raspberry Pi 3 Model B+ (or similar) platform. The project will demonstrate the performance advantage of using the OpenVX API by implementing several optimizations that are enabled by OpenVX including:

- Automatic optimization of memory access patterns via tiling and chaining;
- Use of highly optimized kernels leveraging multimedia instruction sets;
- Automatic parallelization to utilize multiple compute resources, including multicore CPUs and GPUs;
- Automatic merging of common sequences of processing kernels into a single, higher-performance kernel.

2.2 IMPLEMENTATION

The implementation shall be written in C and use OpenCL. Some C++ may be used internally within the implementation, but the external API interfaces must be C. It is expected that there may be code incorporated from OpenCV, but there should not be a dependency on OpenCV as a whole. Optimized multimedia libraries for the CPU (e.g. ARM NEON) are also expected to be leveraged. The development may use the existing Khronos OpenVX sample implementation as a baseline, and any updates should be made directly in the Khronos git repository with history preserved. The cmake and make-based Concerto build systems currently supported on the Khronos OpenVX sample implementation should be supported.

The Khronos OpenVX sample implementation will be available on GitHub at the location below by December 1

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

Additional suggested resources are:

- The AMD Open-Source OpenVX implementation: https://gpuopen.com/compute-product/amd-openvx
- The ARM Compute Library: https://developer.arm.com/technologies/compute-library
- Open source OpenCL 1.2 project for Raspberry Pi: https://github.com/doe300/VC4CL
- A patch to the Khronos sample implementation that leverages the ARM Compute Library to accelerate many of the OpenVX functions: https://khronos.org/openvx/files/openvx-arm_compute.tar.gz

Dependencies on additional third-party components should be minimized, and responses are requested to identify those components. In any case, only open source components should be used, preferably under the Apache 2.0 license.

The OpenVX working group suggests the latest Raspberry Pi platform (currently Raspberry Pi 3 model B+) as the hardware platform, with support for the camera module. The implementation should build and run entirely on the Raspberry Pi platform running the Raspbian operating system. Alternate platforms with low cost, high availability, and comparable performance can be recommended by the implementor.

2.3 CONFORMANCE TESTING

The implementation developed for this project must be submitted for conformance through the usual OpenVX 1.2 Adopter's process, with the exception that the Adopter's Fee will be waived. The implementation must pass all conformance tests for the full base specification (excluding extensions) without modification of the tests. The implementor will provide instructions to build and run the implementation and conformance tests such that the results can be reproduced by members of the OpenVX working group.

2.4 PERFORMANCE REQUIREMENTS

The implementor will demonstrate the performance gain of each of the optimizations mentioned in section 2.1. Each of these optimizations must be enabled/disabled via a run-time or compile-time flag, with each optimization independently producing an increase in overall performance. This performance increase can be shown using the conformance tests, or the implementor can create application-level OpenVX programs to demonstrate each performance increase. Like the conformance tests, any provided sample application-level programs must be completely portable to other OpenVX implementations and may not use any hardware-specific features not exposed through the OpenVX API.

Suggested performance targets for each of the optimizations are as follows:

- Automatic optimization of memory access patterns via tiling and chaining: 2X speedup
- Use of highly optimized kernels leveraging multimedia instruction sets: 4X speedup
- Automatic parallelization of the processing to utilize multiple compute resources, including multicore CPUs and GPUs:
    - 90% of linear speedup for multicore CPUs
    - 4X speedup of a kernel running on GPU vs. a single CPU
- Automatic merging of common sequences of processing kernels into a single, higher-performance kernel
    - 50% speedup on merged kernels vs. independent kernels

The speedup targets should be met for at least one test case—not all tests or apps are expected to meet all these criteria. *Every* OpenVX function should show *some* performance improvement over the existing OpenVX sample implementation. Counterproposals to the above, with justification, will be considered.

3 DELIVERABLES AND ACCEPTANCE CRITERIA

3.1 DELIVERABLES

The scope of the OpenVX Implementation project will include the following deliverables:

- All source code for the implementation and tests with instructions for building and running
- An Implementation Notes Document summarizing implementation decisions made during the project
- The test log of running and passing the OpenVX conformance tests
- A performance report demonstrating achievement of the performance goals described in section 2.4

3.2 Review Period

1. The working group will have a review period of 4 weeks to review the implementation code. At the end, it will provide a list of issues to be fixed, ranging these issues as critical/high/medium/low.
2. All critical and high issues need to be fixed/addressed as part of the Acceptance Criteria.

3.3 Acceptance Criteria

1. 100% passing on the OpenVX 1.2 base specification conformance tests without test modification
2. Tests or sample applications demonstrating the performance goals in section 2.4
3. All issues found during the review period and classified as critical/high should be fixed

4 Project Scoping and Schedule

The OpenVX working group estimates that the project can achieve complete implementation, testing and documentation, in no more than 20 person weeks.

Below are the suggested project milestones. An overall goal is to complete work in time for release at the Embedded Vision Alliance (EVA) Summit on May 20, 2019, and use in the Khronos EVA Summit workshop. We will assess progress on a weekly basis, so the feature coverage timeline below is only a rough guideline to the order in which we expect to have validator and tests written. Please provide detailed milestone dates that you can commit to delivering:

Milestone	Date	Content	Notes
M1		Khronos releases RFQ	
M2	M1 + 4 weeks	RFQ responses received by Khronos	
M3	M2 + 2 weeks	Contractor selected and notified	
M4	M3 + 3 weeks	Contract executed and start of work	10% of money is provided
M5	M4 + 6 weeks	50% of the performance goals met	30% of money is provided
M6	M5 + 4 weeks	100% of implementation complete and 100% of the performance goals met	30% of money is provided
M7	M6 + 4 weeks	Review Period over	
M8	M7 + 3 weeks	Issues found during review period fixed, project is complete.	30% of money is provided

5 KHRONOS NDA, CONTRACTOR AND MEMBERSHIP AGREEMENT

The selected contractor will be required to execute the standard Khronos Contractors Agreement, available on request, with Milestones and Costs entered into Exhibit B and Contractor Disclosures entered into Exhibit C.

If the selected contractor is not a Khronos member, the contractor shall also be required to execute the standard Khronos membership agreement (with fees waived) for the duration of the project to gain access to confidential materials and meetings for the sole purpose completing deliverables in this RFQ.

No work shall begin, and Khronos shall be liable for no costs or expenses, until the selected contractor is in receipt of a mutually executed Contractor's Agreement.

It is important that contractors understand that Khronos will be assessing progress on a regular basis and reserve the right to terminate or renegotiate the contract in the event insufficient progress is being made.

6 RFP RESPONSES

The RFP response materials will form the basis for detailed milestone and cost negotiations for the final contract with the selected vendor or vendors. Please provide the following information in the format of your choice:

- Identification of deliverables on which you wish to bid;
- Justification of the alternate platform if you propose something other than the Raspberry Pi;
- Proposed schedule, highlighting any differences from the suggested milestones in Section 4;
- The hourly cost for engineering resources from your company, the minimum and maximum number of hours you can commit to this project on a weekly basis, and a description of the qualification of the engineering resource(s) which would be used;
- The total project cost to Khronos. We can accept time and material or fixed cost bids – but strongly prefer fixed cost proposals;
- An indication you are willing to work under the terms of the standard Khronos Contractor Agreement and execute the Khronos membership agreement if necessary;
- Any issues or risk factors that you wish to highlight;
- Supporting materials, including background materials about your company, highlighting other relevant experience and expertise for this project.

RFQ responses are requested by the close of business on January 15, 2019 and should be sent to openvx-rfq@khronos.org.

7 REFERENCES

[1] The OpenVX 1.2.1 Specification: https://www.khronos.org/openvx/

