Frameworks for ComputeCpp

While SYCL and ComputeCpp offer developers the ability to take advantage of heterogeneous programming and hardware, they also give developers access to a range of specialized frameworks. These frameworks have been integrated with ComputeCpp to enable them to take advantage of the parallel capabilities of OpenCL hardware.

TensorFlow™

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. Find out how to use TensorFlow with SYCL in our guide.

Eigen

The Eigen library is probably the most popular C++-based, high-performance dense linear algebra library. It implements matrix and vector operations, geometric transformations, numerical solvers, and related algorithms. The Eigen tensor (n-dimensional array) operations are used as part of the TensorFlow framework along with other Eigen operations. The library benefits from acceleration using heterogeneous hardware, since matrix and vector operations require many parallel calculations and are thus suited to parallelization on GPUs.

Follow the Eigen Guide article to find out how to use the SYCL accelerated Eigen operations. There is also a blog article that explains how the SYCL implementation of Eigen was implemented.

VisionCpp

VisionCpp is a lightweight header-only library for computer vision and image processing. The aim of the library is to provide a toolbox that enables performance portability for heterogeneous platforms using modern C++. VisionCpp is written using SYCL 1.2.1 and compiled/tested with ComputeCpp to accelerate vision code using OpenCL devices.

SYCLParallelSTL

Parallel STL is an implementation of the Technical Specification for C Extensions for Parallelism, current document number http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4507.pdf[N4507]. This technical specification describes a set of requirements for implementations of an interface that computer programs written in the C programming language may use to invoke algorithms with parallel execution. In practice, this specification is aimed at the next C standard (C 17) and offers the
opportunity to users to specify execution policies to traditional STL algorithms which will enable the execution of those algorithms in parallel.

**SyclParallelSTL** exposes a SYCL policy in the experimental::parallel namespace that can be passed to standard STL algorithms for them to run on SYCL. Currently, only some STL algorithms are implemented as part of the project and work is ongoing.

### SYCL BLAS

The **SYCL BLAS** project implements BLAS (**Basic Linear Algebra Subroutines**) using SYCL. BLAS defines a set of low-level routines for performing common linear algebra operations such as vector addition, scalar multiplication, dot products, linear combinations, and matrix multiplication. They are the de facto standard low-level routines for linear algebra libraries.

SYCL BLAS is a current work in progress research project from an ongoing collaboration with the High Performance Computing & Architectures (HPCA) group from the **Universitat Jaume I**.