<http://docs.mitk.org/nightly/MitkOpenCL_Overview.html>

OpenCL Module

The MITK OpenCL Module provides a basic class structure to allow usage of OpenCL-accelerated parallel computing.

**Build Instructions**

The MITK OpenCL module needs an OpenCL Device (most likely a graphics card but works also on some CPUs) and the corresponding driver. Furthermore, the OpenCL library and the headers are needed. A list of supported hardware is provided by [Khronos.org](http://www.khronos.org/conformance/adopters/conformant-products" \l "topencl). For the driver and the libraries please look at the support pages of the hardware manufacturer. Quick Links to the most likely of them:

* [NVIDIA OpenCL Page](https://developer.nvidia.com/opencl)
* [AMD/ATI OpenCL Page](http://www.amd.com/de/products/technologies/stream-technology/opencl/pages/opencl.aspx)
* [Intel OpenCL SDK](http://software.intel.com/en-us/vcsource/tools/opencl-sdk)

To activate the module, you have to activate the [CMake](http://docs.mitk.org/nightly/namespaceCMake.html) option:

MITK\_USE\_OpenCL

The build system tries to find the OpenCL library and the include path automatically. If this attempt fails, you will be prompted to manually specify the following variables:

OPENCL\_INCLUDE\_DIRS

OPENCL\_LIBRARIES

**Detailed description**

For own implementations, the OpenCL Module allows for building up a filtering pipeline in MITK Style ( see more in [Pipelining Concept](http://docs.mitk.org/nightly/PipelineingConceptPage.html) ). The OpenCL filter can also be simply connected to an existing MITK image filtering pipeline since the [mitk::OclImageToImageFilter](http://docs.mitk.org/nightly/classmitk_1_1OclImageToImageFilter.html) provides a SetInput and GetOutput method expecting an [mitk::Image](http://docs.mitk.org/nightly/classmitk_1_1Image.html) and returning it respectively.

**Examples**

Here is a list of [examples](http://docs.mitk.org/nightly/MitkOpenCL_Example.html):

* [Implementing Own OpenCL-based Image Filter](http://docs.mitk.org/nightly/MitkOpenCL_BinaryThresholdFilter.html)