

Advanced Topics in Digital Image Processing

Class 5 – Introduction to OpenCL Image Processing

Objectives:

The main objective of this session is to have a first contact with the OpenCL library, discover the OpenCL resources available on your personal computer and learn how to develop OpenCL kernels. The exercises are planned to provide a smooth introduction to the PyOpenCL library.

Exercises:

0 – Test OpenCL driver installation

To test whether you have OpenCL support drivers installed, start by installing the application (GPU_Caps_Viewer_Setup_v1.62.0.0.exe) and check the OpenCL panel to see if the driver is available. If you are not, you will need to download the corresponding driver from NVidia, AMD, or Intel.

0 – Prepare Python Environment for OpenCL

Add PyOpenCL to your python environment

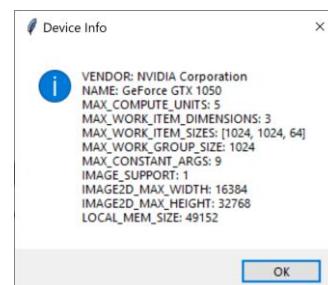
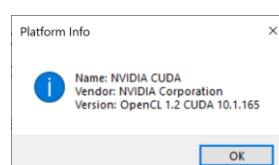
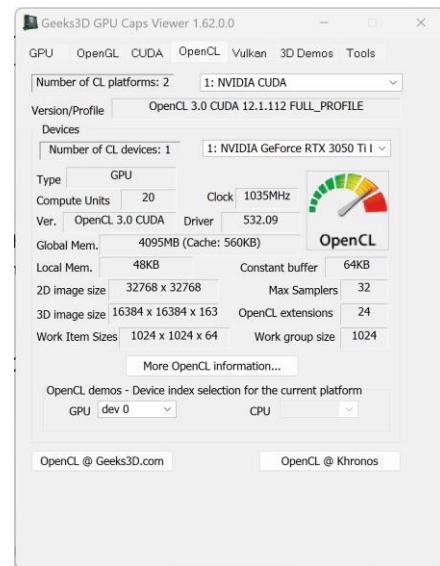
Note: for documentation please visit the website:

<https://documentation.tician.de/pyopencl/index.html>

1 – Show OpenCL Platform and Device details

Use the `platform.get_info` and `device.get_info` methods of the **OpenCL library** to show the properties of the OpenCL device on a MessageBox (*from imageForms*).

Note that your computer may have multiple platforms and devices (Intel, AMD and NVidia), in this case show all platforms and their devices.



2 – Multiply a vector by a constant and sum all elements

The objective of this exercise is to learn how to:

- create memory the device;
- exchange data between the Host and the Device and vice versa;
- pass parameters to kernels;
- execute kernels.

To this end, develop a kernel which multiplies by a constant an array of 25 integer elements and also returns the sum all elements, all variables (array, constant, and sum) are received as parameter (address or value).

The Host must:

1. initialize an array with values between 1 and 25
2. create a buffer that is initialized with a copy of the array
3. create a buffer with just one single sized array to store the sum
4. set kernel arguments (array Buffer, constant and sum output Buffer)
5. launch the kernel (use a work group size of 10)
6. copy the arrays from the device to the Host and show them on screen.