

# Exercise Sheet 4 - OpenCL

Yan Xia, Siyuan Mei, Linda-Sophie Schneider, Andreas Maier

June 30, 2025

In this exercise, we will learn about OpenCL and parallel computation on the graphics card. First, we implement a summation of Grids on GPU and CPU and compare the runtime. Afterwards, we accelerate the reconstruction process by performing the parallel-beam backprojection on the GPU.

Hint: Look for the OpenCL Cheat-Sheet on the web.

1. **Setup:** We are using `pyopencl` for our implementation. You can install with:

```
pip install pyopencl==2021.2.2
```

If you encounter issues when installing `pyopencl` via `pip`, you might need to download the wheel suitable for your system and graphics card from [this website](#) (e.g. `cl12` means `opencl` version 1.2, `cp36` is Python version 3.6, `win_amd64` means windows 64bit system).

2. **OpenCL Kernels:** Implement an OpenCL kernel that adds two Grids together. Each grid has a size of  $1000 \times 500$  and the  $i, j$ -th pixel value is  $i \times j$ . Implement the host code in python and `.cl` kernels to test your kernel.
  - Use OpenCL texture buffers with `cl.image_from_array` to perform the grid addition;
  - Compare the runtime of the GPU approach with the runtime of a Grid addition using `numpy`'s `add` method or your own two for-loops for addition.
  - Print out the last element value to see whether the addition result is correct, e.g., `np_grid3[999, 499] = 997002.0`;
  - Change the grid size from  $1000 \times 500$  to  $10000 \times 5000$  to see the OpenCL acceleration better.
3. **OpenCL Back-Projection:** Implement a GPU Version of the parallel back-projection from Exercise Sheet 2. This means you have to implement the device code in a `.cl`-file and the host code in python:  
`backprojectOpenCL(sinogram, size_x, size_y, spacing)`  
The kernel function in the `.cl` file should be

```
kernel void backproject(read_only image2d_t sinogram, write_only image2d_t  
reco, int num_projections, int detector_size, float angular_increment_degree,  
float detector_spacing, float detector_origin, int reco_sizeX, int reco_sizeY,  
float reco_originX, float reco_originY, float reco_spacingX, float reco_spacingY)
```