**Themed Project**

**Moca Emanuel & Maxim Georgiana**

## Common requirements

Each project will have 2 implementations: one with "regular" threads or tasks/futures, and one distributed (possibly, but not required, using MPI). A third implementation, using OpenCL or CUDA, can be made for a bonus.

The documentation will describe:

* the algorithms,
* the synchronization used in the parallelized variants,
* the performance measurements

**Chosen Theme**

Applying a filter (transformation) on an image. We choose the Mean Filter. Mean filtering is a intuitive method of IMG_256smoothing images, i.e. reducing the amount of intensity variation between one pixel and the next. It is often used to IMG_257reduce noise in images.

## **Computer Specification**

* CPU: Intel Core i7-8750H, 2.20GHz up to 4.10Ghz (turbo)
* RAM: 12 GB

**Algorithms**

The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a IMG_256[convolution filter](https://homepages.inf.ed.ac.uk/rbf/HIPR2/convolve.htm). Like other convolutions it is based around a [kernel](https://homepages.inf.ed.ac.uk/rbf/HIPR2/kernel.htm), which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3×3 square kernel is used, although larger kernels (e.g. 5×5 squares) can be used for more severe smoothing. (Note that a small kernel can be applied more than once in order to produce a similar but not identical effect as a single pass with a large kernel.)

**Performance Tests**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **225 x 225** | **1392 x 794** | **4000 x 3000** |
| Using **Threads** | 162ms | 538ms | 7s |
| Using **MPI** | 244ms | 5s | 58s |

**Parallelization**

* For the Threads approach, we use the Java Thread Pool with 4 threads.
* For the MPI approach we use one master and 2 workers

Each thread/worker computes equal numbers of operations. An operations is: applying 3x3 kernel over a zone from the image.