Final details for Assignment 2

NOTE 1: deadline postponed to Jan, *20th*, 2021

NOTE 2: *BUG* fixed in the `read_and_write_pgm_image.c`

Due to the limited computational resources, the deadline is postponed to **Jan, 20th**.

Test files

In the G. drive repository where the lectures' videos are stored you find an additional folder `Materials for Assignment2` (https://drive.google.com/drive/u/0/folders/1iDXlDuOTxzCcaJAPoHIJ0IYV4rMXt8fs) where there are the test files to be used for automatic validation of your work.

- `test_picture.pgm` is the original picture that you have to blur; it is a small one (~20MB) because not all of you may have a broadband connection during the vacations.
- `test_picture.b_0_101x101.pgm` is the blur with an Average Kernel of size 101 (i.e. radius 50; that is the first kernel mentioned in the Assignment presentation).
- `test_picture.b_1_101x101_02.pgm` is the blur with a Weight Kernel of size 101 and central value of 0.2 (that is the second kernel mentioned).

As noted by a student, there was an inconsistency in the uploaded test picture blurred with the Weight Kernel. The correct version has been uploaded

The **naming convention for the output files** that you must respect is:

```
output file name = original_file_name.b_#TYPE_#XSIZEx#YSIZE<_#CENTRALVALUE>.pgm
```

where:

- the `original_file_name` is the file name of the original file without the `.pgm` (i.e. "`test_picture`" for our example)
- `#TYPE` is an integer that corresponds to the kernel type: Average = 0, Weight = 1, Gaussian = 2.
- `#XSIZE` and `#YSIZE` are the *x*- and *y*- sizes of the kernel (in your case you are allowed to use square kernels, so `#XSIZE` = `#YSIZE`; however, you are free to generalize).
- `#CENTRALVALUE` is meaningful only for the Weight Kernel, and so it will be present in the name only for that case. In the test check that you are given the `_02` is the central value 0.2; since we are considering only values in the range `[0:1]` follow the same convention (challenge not required: can you correctly manage to have central value above 1.0, i.e. to enhance the luminosity ?)

In the case (not required) that you want to challenge yourself and avoid the loss of luminosity at the borders, you find the corresponding results in the sub-folder `border_effect_accounted`.

How the blur result will be checked

For the purpose of acceptance of your results, your final blurred image will be contrasted with the uploaded reference R image.
In order to your image V be considered valid, the following must hold

$$|V_{i,j} - R_{i,j}| < 0.05\ \times \text{max\_value} \tag{1}$$

for each pixel (i, j), where `max_value` is the maximum value possible for a pixel (i.e. 65535) (note that is a **very** generous approach).

The details about your code will be discussed at the oral examination.
Please, report the absolute running time for your result (i.e. the configuration that you have used - how many MPI tasks, how many threads and the affinity)

File for scalability study

In order to have your scalability study, use the slightly more larger file (21600x21600 pixels) `earth-large.pgm` that you find already on the Orfeo cluster in my scratch area `/storage/dssc/tornatore/earth-large.pgm` which should be readable by all of you.
For those that do not have access to the Orfeo cluster, the same file is available in the G. drive under the subfolder `Materials for Assignment2/Image for scalability study` (https://drive.google.com/drive/u/0/folders/1Ea5N1CUbvVNl-8o4fu6tF2VkhF3XHJ_C).

Assess the scalability of your codes using this image and:

- Weight Kernels of size 11, 101 [the prevsiously required 501x501 kernel is now dropped; use only 11x11 and 101x101 ones]
- from 1 to the maximum number of physical cores on a Orfeo node for the omp version.
- using 1 node that you are allowed to use on Orfeo [perviously, it was requested to use more than 1 nodes for the MPI version; due to the limited computational resources, this requirement is dropped and you are entitled to use only 1 node].
- please report the absolute timings (of course, the run-time for 1 MPI task / 1 thread is sufficient).

Of course, **in addition** you may want to generate whatever extremely large image that you want (for instance with just white noise) to perform a better scalability study.

Bug in `read_write_pgm_image.c`

I realized only now that in the provided example file `read_write_pgm_image.c` there was a bug in the `swap_image()` routine:

```
for ( int i = 0; i < size; i+= 2 )
```

must be replaced by

```
for ( int i = 0; i < size; i++ )
```

I'm sorry for that. Due to this bug, which may have delayed some of you, the deadline is postponed by 1 week.