Assignment 4
Introduction to GPGPU with Nvidia Warp
Weight: 15% of the final grade
Due date: Sunday, December 4, 11:59pm.

Assignment description

In this assignment you will need to implement two basic image-processing algorithms - one for sharpening and one for noise removal - using the Nvidia Warp API. Each algorithm will be written in a dedicated kernel.

For sharpening, use unsharp marking described in class. For noise removal, you can pick one of the low-pass filtering algorithms described in Lecture 12.

Your executable will run as follows:
`python3 a4.py algType kernSize param inFileName outFileName`
where:
- `algType` is either `-s` (sharpen) or `-n` (noise removal)
- `kernSize` is the kernel size - e.g. `3` for 3x3, `5` for 5x5, etc.. It must always be positive and odd.
- `param` is the additional numerical parameter that the algorithm needs - e.g. the scaling value `k` for unsharp masking or sigma for the gaussian. If your algorithm doesn't need any additional parameters once it knows the kernel size, just pass some dummy value here (e.g. 0)
- `inFileName` is the name of the input image file
- `outFileName` is the name of the output image file

Note: do not use the simple averaging filter for noise removal - use one of the more sophisticated algorithms discussed in Lecture 12. The averaging filter is already part of the unsharp masking implementation, so you will not get any marks for it if you also use it for noise removal.

Your algorithm must work with both greyscale and colour images. You will use the PIL image library. A stub that loads a PIL image and runs a simple kernel on an image has been provided for you.

In addition to the implementation, provide a document in which you:
- Provide any information that we might need when running your code
- Describe which algorithms you have implemented
- Explain how you have implemented the kernel for this algorithm using the Warp API. You can paste the code into the document so you can more clearly explain it.

- Make sure your explanation includes a description of how you handle the borders. Note that you cannot leave them unprocessed. You must implement an approach that computes new values for the entire image, including the border - i.e. Option 2 or 3 from Lecture 11

Grace period

The grace period for this algorithm is 48 hours. There will be no penalties applied during the grace period. No submissions will be accepted after the end of the grace period.

Grading

The grading scheme for this assignment will be released shortly.

Please note that, since this assignment evaluates your knowledge of GPGPU, **no marks** will be given for a solution that is not parallelized using the Warp API.

Submission and Evaluation

Submit the assignment using Moodle. Submit only the source code and the makefile. Bundle the code in a zip file.

The assignment will be marked using the standard CIS*3090 Docker image provided on the course website and discussed in class. We will download and run a fresh image, create a container, and use Docker to run and grade your code. Make sure you test your code accordingly.

Academic Misconduct
<u>This assignment is individual work and is subject to the University Academic Misconduct Policy</u>. See course outline for details.