Lab 3 report

Marc Aguilar and Alejandro Fernández

Exercise 1

There's not much to comment here. Special attention has been put in the order of loops in order to take advantage of data locality. Furthermore, we have also used the directive `collapse(2)` in order to optimize the nested loops that appear when accessing all the pixels of a 2D image.

The last important thing to comment here is that the **time required to complete the filtering process is much higher than the one needed with the CPU** (in our test run, it takes `2.054103 s` with the GPU). We believe that this is due to the **PCIE bus bandwidth limitations**, which creates a huge overhead when compared to the simple computations we need to perform with the proposed filters.

Exercise 2

In this exercise we have implemented the same code but in an asynchronous manner. More specifically, each filter is computed in a separate stream in which each memory update is performed. The data region has been changed from structured to **unstructured**.

In this case the result is slightly worse than in the previous exercise (`2.149212 s`). This is probably caused by the fact that our code does not benefit from using an unstructured data region with `update` directives, since we only perform computations once for each image.

Exercise 3

In this exercise we have mapped two variables from the host `image_host` and `image_inverse_host` to two variables on the device: `image_device` and `image_inverse_device`. After having performed this mapping, we have only used the host variable, since the GPU operations are automatically linked to the CPU variable.

Regarding the execution time, once again we have evidence that we are limited by the PCIE bus, since the execution time is more or less the same as in the first exercise (although we are just using one filter now).