Computer Graphics IV Tutorial

– Assignment 2 –

**Computer Graphics and
Multimedia Systems Group**
Martin Lambers

Assignment 1 Simple Image Processing: Separable Filter

Extent the image processing program from the last assignment by implementing a different approach to the simple averaging filter, using two filter steps and an intermediate result (see below for details). Add CUDA kernel timing functions and use them to verify that this new version of the filter performs better than the original one.

The averaging filter, like many other image manipulation filters, is separable. That basically means that it can be computed using two separate steps to reduce the number of computations and memory accesses, at the cost of storing intermediate results.

For the 5 × 5 averaging filter we implemented in the first assignment, this separation is as follows:

1. For each pixel, sum up the 5 images values in the horizontal 5 × 1 neighborhood and store the result in a temporary image array.
2. For each pixel, sum up the 5 temporary image values in the vertical 1 × 5 neighborhood, divide by 25, and store the result in the output image.

Instead of having one kernel that performs 25 memory reads and additions, we now have two kernels that each perform 5 memory reads and additions, but we also need a temporary image array.

In your implementation, choose a suitable data type for the temporary array! (The sum of `unsigned char` values may not fit into an `unsigned char` variable!)