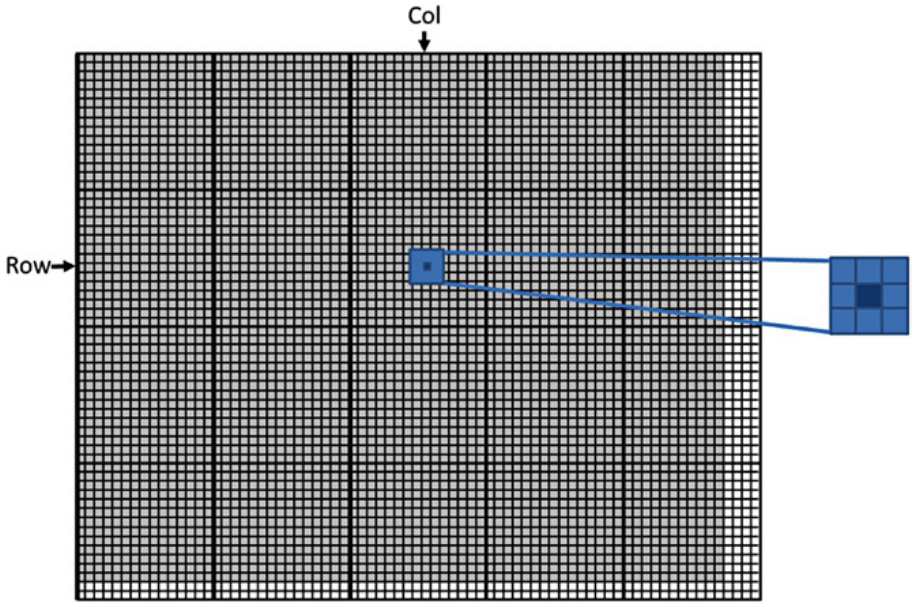
CENG443 **Fall 2018**

PROGRAMMING ASSIGNMENT 2

Due date: 10.12.2018 10:00 am

You are required to implement an image blurring kernel using CUDA. In this kernel, you are required to use the **shared memory** efficiently (by tiling) to increase the performance.

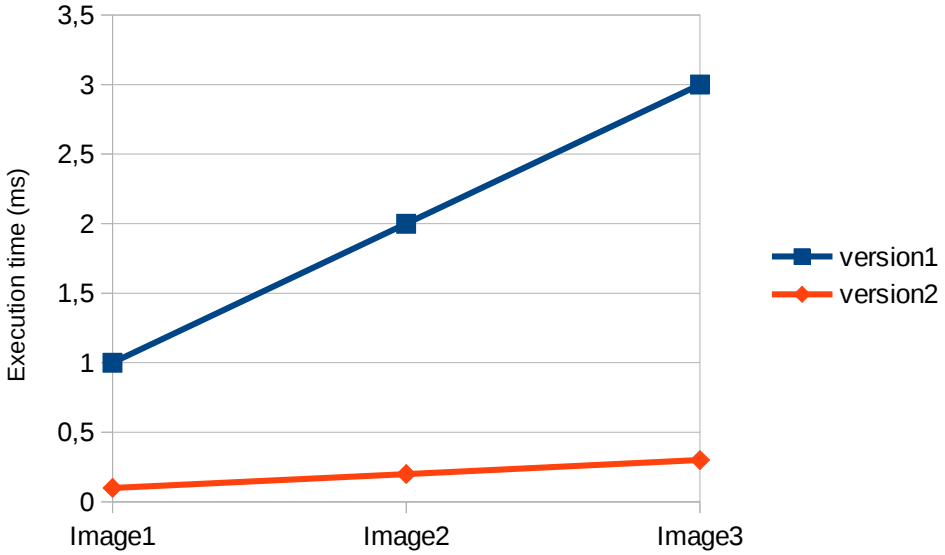
Your kernel will use a 3 × 3 patch (as shown in the figure below). When calculating an output pixel value at the (Row, Col) position, we see that the patch is centered at the input pixel located at the (Row, Col) position. The 3 × 3 patch spans three rows (Row-1, Row, Row+1) and three columns (Col-1, Col, Col+1). To illustrate, the coordinates of the nine pixels for calculating the output pixel at (25, 50) are (24, 49), (24, 50), (24, 51), (25, 49), (25, 50), (25, 51), (26, 49), (26, 50), and (26, 51).



You **can** start with the image blur kernel version provided in the lecture, also in the recommended textbook (the version without shared memory usage). You can also write your own code from scratch (no extra credit for this).

As part of the assignment (a very important part!), you need to run a set of experiments and evaluate the performance of your implementation. You need to compare the base implementation (without shared memory usage) with your improved one. To measure the timing of your kernel execution, you **must** use CUDA events (For more information, and an example: https://devblogs.nvidia.com/how-implement-performance-metrics-cuda-cc/). You **can** find/use the related timing functions in **wb.h** file provided in the lab, and use them directly.

In the experiment part, you will first execute the version without shared memory usage (version1). Then you will execute the improved version (version2). By collecting the execution times of those executions, you are required to draw a graph presenting the performance differences for each input image given as part of the assignment (similar to the following graph). If you propose different improvements (or parameters related to your improvement such as tile size), also demonstrate those versions as part of the graph.



Notes:

- You need to check boundary conditions to support different image sizes.
- You are required to run your programs on UheM machines **or** on your machine with a GPU. At the beginning of your execution, make sure that you provide the specifications of your GPU as in the first programming assignment. We will run your code on UheM if you don't.
- You are required to write a **report** which includes implementation details, graphs, your observations about the performance of your implementation, how you interpret the results. The report is required to be a substantial part of your submission.
- You are given a source file as a template (**test.cu**, using wb.h file provided in the lab for the file operations) and a set of images to be tested for your code. You can (don't have to) start with this template, and make sure your experiments will include the results for the given images (you can put additional images if you think it is necessary).

Submission: You are required to submit your **commented source code** and **report** to CMS. Please create a compressed file including all source files and report; and name it as yourstudentnumber_P2.zip (e.g. If your student number is 201812345678, the file name must be 201812345678_P2.zip).