UNIVERSITÄT TÜBINGEN
PROF. DR.-ING. HENDRIK P.A. LENSCH
LEHRSTUHL FÜR COMPUTERGRAFIK
LUKAS RUPPERT (LUKAS.RUPPERT@UNI-TUEBINGEN.DE)
FAEZEH ZAKERI (FAEZEH-SADAT.ZAKERI@UNI-TUEBINGEN.DE)

MARCH 6, 2024

MASSIVELY PARALLEL COMPUTING
ASSIGNMENT 2

Instructions

Download the framework `exercise02.zip` from the ILIAS course web page.
Present your results to the exercise instructors to get a grading on this exercise sheet.

2.1 Asynchronous Memory Transfer (30)

- Start with the `upload` template.
- You might have noticed that the upload in the testDotProduct.cu was slowing down the overall performance. Here comes the solution:
- Transfer data to the GPU in three different modes
  - 0: Using simple memcpy
  - 1: Using memcpy from non-pageable memory
  - 2: Using asynchronous memcpy with two streams
- Use the provided skeleton and fill in the missing gaps
- Use the profiler (NVIDIA Nsight Systems) to create an application timeline for all three modes
  - Identify the memory transfer speed for mode 0 and mode 1
  - Identify the difference between mode 1 and mode 2
    You may notice that the overall difference is runtime is small since this problem is memory bound. However, you should see the benefit for more compute-intensive applications.

2.2 Reverse Array (30)

- Given an input array $\{a_0, a_1, ..., a_{n-1}\}$ in pointer d_a, store the reversed array $\{a_{n-1}, a_{n-2}, ..., a_0\}$ in pointer d_b
- Start from the `reverseArray` template
- Part 1: Compute the number of blocks to launch
- Part 2: Implement the kernel reverseArrayBlock()

2.3 Cross Correlation (40)

- Start from the `crossCorrelation` framework.
- Compute the normalized cross correlation between two images $f, g : \mathbb{R}^2 \to \mathbb{R}$, component-wise for RGB

$$\begin{aligned} \bar{f} &:= \sum_{x,y} f(x,y)/(M \cdot N) \\ \bar{g} &:= \sum_{x,y} g(x,y)/(M \cdot N) \\ f'(x,y) &:= f(x,y) - \bar{f} \\ g'(x,y) &:= g(x,y) - \bar{g} \\ (f' * g')(X,Y) &= \sum_{x,y} f'(x,y) \cdot g'(X+x, Y+y) \end{aligned}$$

- Use the provided skeleton and fill in the missing gaps
- In folder `images/` you will find example input images
- In folder `referenceImages/` we have pre-computed the solution for different combinations of input images
- Use these pre-computed solutions to check that your code works correctly!