Université de Pau et des Pays de l'Adour
Département de Mathématiques
2022-2023

M2-BigData : GPGPU
Chapter 3 – Exercice 2

Objective

The purpose of this exercice is to convert an RGB image into a gray scale image. The input is an RGB triple of float values and the program will convert that triple to a single float grayscale intensity value. A pseudo-code version of the algorithm is shown bellow :

```
for i from 0 to height do
  for j from 0 to width do
    idx = i * width + j
    # here channels is 3
    r = input[3*idx]      g = input[3*idx + 1]    b = input[3*idx + 2]
    grayImage[idx] = (0.21*r + 0.71*g + 0.07*b)
  end
end
```

Instructions

Edit the given code `2-imgTograyscale.cu` to performs the followings steps :
— allocate device memory
— copy host memory to device
— initialize thread block and kernel grid dimensions
— invoke CUDA kernel
— copy results from device to host
— deallocate device memory
The execution is performed giving the 2 filenames as program arguments :
`./2-imgTograyscale inputImg.png outputImg.png`
The input image must be a colored image. The reading and writing is performed by two functions `read_image_asfloat` and `write_image_fromfloat`. In this exercise, images are represented as RGB or gray with float values.

The CUDA thread grid must be computed from the image dimensions. A common approach is to define squared blocks (called tile), and the grid size accordingly. Mind the arbitrary image dimensions which are generally not a multiple of tile dimensions. Use the given images `Lena.png` then the `Lena1080.png` or any RGB image you want.

Questions

1. How many floating points operations are being performed in your color conversion kernel ? EXPLAIN.
2. How many global memory reads are being performed by your kernel ? EXPLAIN.
3. How many global memory writes are being performed by your kernel ? EXPLAIN.
4. How do you choose the CUDA thread grid dimensions ? EXPLAIN.