CSci 364, Spring 2023

HW5 - CUDA Image Processing

The nature of the problem usually drives the decision of how to organize the threads in a grid.

If the picture color model is greyscale, the color is defined with one value from 0 (black) to 255 (white). Each location of the 2D array has a single value from 0 to 255 representing the lightness of the pixel.

If the color model is RGB, each location of the 2D array has three integer values representing the brightness of the respective red, green, and blue components of the pixel.

The provided rgb2grey.cpp program reads RGB data values from 3000 x 4499 image file into an array (rgb_data). The program then converts the red, green, and blue values to a greyscale value using the average method ((red + green + blue) / 3) and stores the grey scale value in another array (bw_data). Finally, the output array is written to a file. The provided tar file also includes a Makefile and Slurm script to compile and run the rgb2grey.cpp program.

Convert the CPU program to a CUDA program. Specifically, copy rgb2grey.cpp to rgb2grey-cuda.cu and do the following.

- Implement the convert_avg function as a CUDA kernel function. Your kernel function should replace the two for-loops with values for row and col based on blockDim, blockIdx, and threadIdx.
- Allocate arrays in device memory for RGB and BW data (cudaMalloc).
- Copy the host's RGB data array to the device's RGB data array (cudaMemcpy).
- Call the convert_avg kernel function with an execution configuration where the threads per block is a 2D dim3 structure (ex. dim3 threads(16, 16, 1)). The blocks per grid argument should also be a 2D dim3 structure.
- Free the device memory for both device arrays (cudaFree).

Compile and run your CUDA program using the same Makefile and slurm-cuda.sh Slurm script. Your output should be identical to the provided key-avg.txt file. You can use the diff tool on Talon to compare files.

```
diff key-avg.txt bw-black-lab-cuda.txt
```

Please include a comment block with your name at the top of your .cu file. Submit your .cu file to Blackboard.