

COMP 605 - HW04

Akhil Perimbeti

May 16, 2022

RGB to Greyscale Image Conversion using CUDA & OpenCV

OpenCV libraries were installed and used for compilation on the DGX Server.

NOTE!!: Please **INCLUDE** the following line of code in the compiler to fully link the library:

```
$ export LD_LIBRARY_PATH=:/home/perimbeti/build/opencv/lib
```

(found when you type in **\$make link** using the Makefile)



Figure 1: Image in RGB (bear.jpg) and Image in Greyscale (greyscale_bear.jpg)

Figure 1 above shows the results of the image conversion from RGBA to Greyscale using a block size of 32. We can see the resolution of the output image (greyscale_bear.jpg) is clear and the greyscale conversion was successful taking a total of 18.099 ms for the kernel execution call to run (for 1000 iterations).

CODE COMPILATION & RESULTS

For an a single iteration of the greyscale conversion kernel call, the difference between different block and grid sizes was too small and too insignificant. So in order to be able to compare and draw conclusions, I iterated the kernel call (RGB2GREY<<>>) **1000 times** and measured the total execution time for each different block and grid size (I essentially **simulated the image being converted to greyscale 1000 times** and recorded execution times). This resulted in the discernible kernel runtime differences we see below:

Original Image Dimensions : [1125×2438]

Format: [ROWS \times COLUMNS]

BLOCK SIZE: 1×1 **GRID SIZE:** [1125×2438]

Kernel Execution Runtime (1000 iterations): 2088.786 ms

BLOCK SIZE: 2×2 **GRID SIZE:** [563×1218]

Kernel Execution Runtime (1000 iterations): 570.175 ms

BLOCK SIZE: 4×4 **GRID SIZE:** [282×609]

Kernel Execution Runtime (1000 iterations): 163.576 ms

BLOCK SIZE: 8×8 **GRID SIZE:** [141×305]

Kernel Execution Runtime (1000 iterations): 47.224 ms

BLOCK SIZE: 16×16 **GRID SIZE:** [71×153]

Kernel Execution Runtime (1000 iterations): 18.356 ms

BLOCK SIZE: 32×32 **GRID SIZE:** [36×77]

Kernel Execution Runtime (1000 iterations): 18.099 ms

As stated above, we can clearly see the differences between the execution times for the kernel (1000 iterations) using the various grid and block sizes. The smallest block size of 1 (grid size of [1125×2438]) had the longest runtime of ≈ 2.1 seconds, and the largest block size of 32 (grid size of [71×153]) had the fastest/shortest runtime of ≈ 19 milliseconds. It is interesting to note that if I chose to run the kernel on 10,000 or even 1,000,000 iterations - these differences in kernel execution times would be even more significant. Using the a block size of 16×16 seems to be where the tolerance for this particular image is capped, as we can see that the execution times for the block size of 16 and block size of 32 are very close to each other. (This result is similar regardless of iterations)