Patrick Hanrahan
Benjamin Hobbs
Assignment 5
GPU

Part 1: Blur Filter

For the blur filter we averaged 5x5 blocks of an input image using CUDA GPU threading commands. The average of the 5x5 block was placed in the center of the equivalent 5x5 output block. To insure we did not write to an out of bounds address, we checked boundary conditions with an "if" statement.

```
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/filter$ ./lab5
FPS = 29.3334
FPS = 41.7086
FPS = 61.9868
FPS = 41.7087
```

Part II: Sobel Filter

USING OPEN CV:

```
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/sobel$ ./lab5 1024 1024 0
Using OpenCV
FPS = 15.7864
FPS = 16.1703
FPS = 17.3225
FPS = 17.2385
```

USING CPU:

```
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/sobel$ ./lab5 1024 1024 1
Using CPU
FPS = 18.5704
FPS = 18.0074
FPS = 18.2068
FPS = 19.8103
```

USING GPU:

```
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/sobel$ ./lab5 1024 1024
Using GPU
FPS = 34.9697
FPS = 34.5561
FPS = 22.9989
FPS = 24.9088
```

nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/sobel$ ./lab5 4096 4096
Using GPU
FPS = 17.0605
FPS = 21.3192
FPS = 21.2824
FPS = 21.3917
FPS = 21.5907

nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/sobel$ ./lab5 512 512
Using GPU
FPS = 39.7147
FPS = 57.5451
FPS = 71.7398
FPS = 64.8412

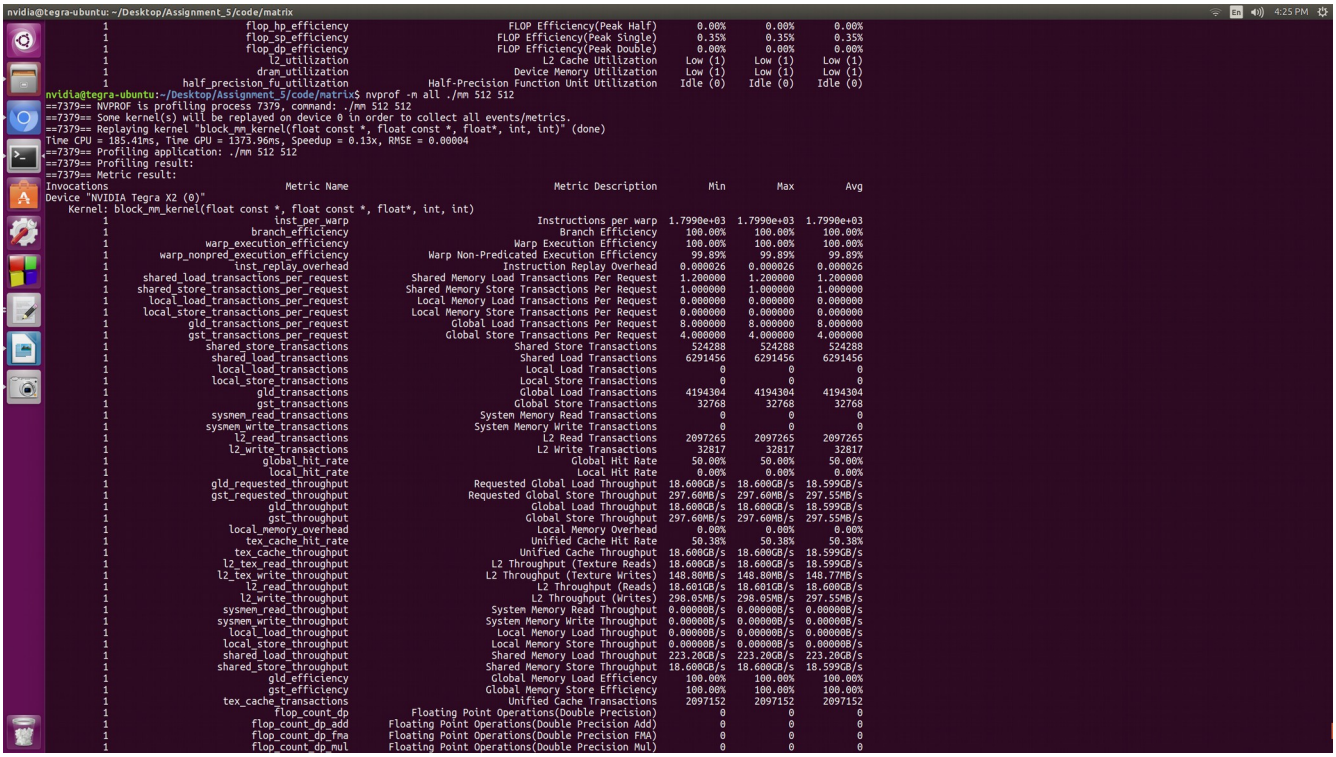
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/sobel$ ./lab5 512 1024
Using GPU
FPS = 37.2168
FPS = 33.358
FPS = 34.6448
FPS = 36.1099

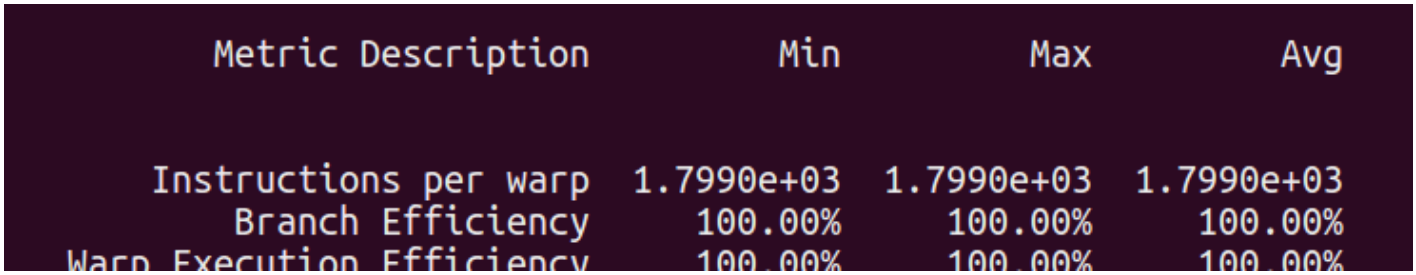
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/sobel$ ./lab5 1024 512
Using GPU
FPS = 30.0601
FPS = 42.2698
FPS = 47.3036
FPS = 46.5513

Part III

nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/matrix$ ./mm 16 16
Time CPU = 0.09ms, Time GPU = 0.11ms, Speedup = 0.81x, RMSE = 0.00000
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/matrix$ ./mm 1024 1024
Time CPU = 1535.44ms, Time GPU = 27.69ms, Speedup = 55.45x, RMSE = 0.00011
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/matrix$ ./mm 16 1024
Time CPU = 20.82ms, Time GPU = 0.53ms, Speedup = 39.63x, RMSE = 0.00000
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/matrix$ ./mm 1024 16
Time CPU = 0.69ms, Time GPU = 0.19ms, Speedup = 3.60x, RMSE = 0.00011
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/matrix$ ./mm 64 64
Time CPU = 0.59ms, Time GPU = 0.13ms, Speedup = 4.65x, RMSE = 0.00000
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/matrix$ ./mm 128 128

Time CPU = 3.43ms, Time GPU = 0.17ms, Speedup = 20.66x, RMSE = 0.00001
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/matrix$ ./mm 128 64
Time CPU = 0.87ms, Time GPU = 0.07ms, Speedup = 12.42x, RMSE = 0.00001
nvidia@tegra-ubuntu:~/Desktop/Assignment_5/code/matrix$ ./mm 64 128
Time CPU = 1.42ms, Time GPU = 0.15ms, Speedup = 9.44x, RMSE = 0.00000





We investigated the difference in warp Execution Efficiency with different block sizes. After changing the block size to 7, one can see the efficiency was reduced from 100 percent to 76.5 percent (below).

```
                    Metric Description         Min         Max         Avg

nt, int)
                  Instructions per warp  2.9470e+03  2.9470e+03  2.9470e+03
                      Branch Efficiency      100.00%     100.00%     100.00%
                Warp Execution Efficiency      76.56%      76.56%      76.56%
Warp Non-Predicated Execution Efficiency      76.51%      76.51%      76.51%
               Instruction Replay Overhead    0.000012    0.000012    0.000012
```