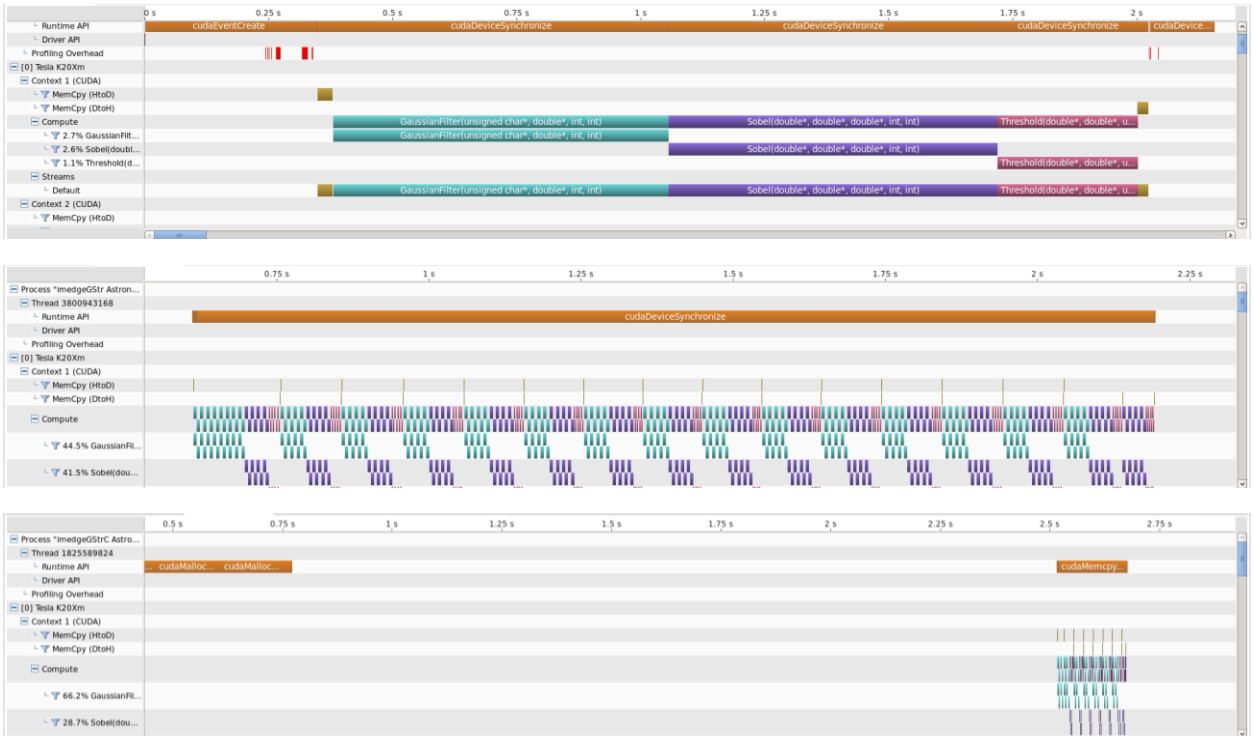
You Wu

ECE 206

Assignment 5 Report

In this assignment, the edge detection C program was adapted to function on the GPU using CUDA. As a result, large chunks of the programs were taken in order to complete the assignment. The first part involved just the unoptimized execution of the edge detection c program on the GPU. Part b involves streaming the execution process which allows for greater concurrency and finer granularity. In the last part, the CPU was involved in some of the execution whereas in the previous two parts, the CPU was merely used as a container to transfer and receive data, contributing nothing to the actual execution. I had decided that the easiest way of transitioning from parts B and C was to allow the CPU to compute the last stream's worth of task. I had significant difficulty with accommodating the CPU to do computation and it appears that the work the CPU does in the sobel kernel is lost.

Below are the NVIDIA visual profiler diagrams for all three parts in descending order.



From the timing diagram of part A, it can be observed that the Gaussian filter and Sobel filter kernels take significantly more time than the threshold kernel. This is due to the recursive nature of computing angles and gradients using 2D matricies. In the timing diagram of part B, it can be seen that streams (and levels) allow for the task to be split up into even more fine grained pieces. One last aspect to notice is that in part C, there exists a significant gap between memory allocation and start of execution. This might have been a point of improvement but I was unable to take advantage of this while doing this assignment.

The following execution times (in ms) were observed when running the three edge detection methods on the same Astronaut.bmp.

	Part A	Part B	Part C
Tfr C->G	489.14	21428.88	2336.62
GPU Exec	23931.05	2216.56	115.96
Tfr G -> C	319.12	6.71	43.51
Total Exec	24739.32	23652.14	2496.08

The table shows that even though the CPU is generally considered inferior to the GPU for highly parallizable tasks such as image processing, even allowing the CPU to do a little bit of work goes a long way in improving performance. The speedup from part A to part C is 10x!