Assignment 4: Image Processing

Sarah Sterchele

CMPT 431

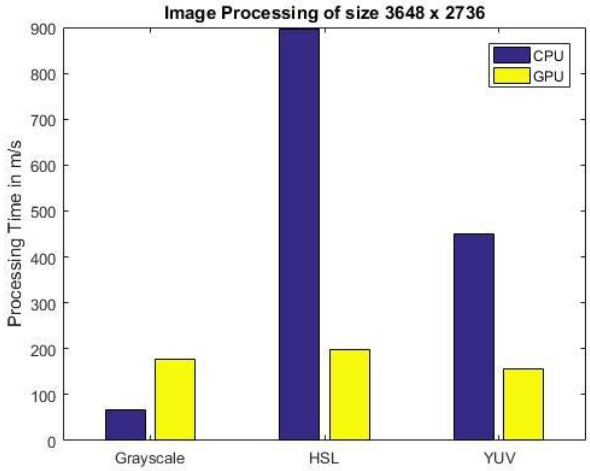
November 30, 2016

Implementation

The histogram creation and the histogram equalization for the GPU utilized a shared memory, which is faster than using a global memory. For the histogram creation, each block kept track of its own histogram. When all threads were finished, the block updated the global histogram. For the histogram equalization, each block actually created its own look-up table (it's pretty minimal calculations, only 256). Then, each block adjusts the image according to its section, using its own table. The reason for doing this is that allocating space for a global memory look-up table and passing it back to the kernel seemed to actually cause worse performance measures.

For the color images, there were many opportunities for parallelism on the GPU. All the color conversion functions were implemented as GPU kernels. These were the most performance heavy operations and really benefitted from the parallelism. A shared or global memory was not necessary for these operations as they were pure conversion calculations.

Performance



The grayscale image processing did not benefit from running on the GPU. The histogram operations, while non-trivial, did not outweigh the cost of CUDA memory allocation and memory copy to the GPU. In fact, sequential grayscale CPU image processing was faster than the grayscale GPU parallelization. I suspect that, in addition to overhead costs, this is due to CPU caching (GPU's do not have caches, and we are basically repeating the same lookups over and over.) Conversely, the color image processing benefitted significantly from the GPU parallelism. The processing time for grayscale and color images were all roughly the same, but the color images had far more floating point operations than grayscale, having to do 3 conversions per pixel. The CPU struggled performance-wise during this process. Image size seemed to not affect the GPU (the smaller image showed slightly better performance), but image size did make a noticeable difference in CPU processing time.

It is interesting to see that GPU's are an asset when doing very heavy graphical computations, and this assignment highlights this. One can see that they can handle **very** large images and processing while almost having a logarithmic running time behavior (at-a-glance observation). CPU's are fast

for smaller things, but the time grows exponentially as the amount of exploitable parallelism increases.