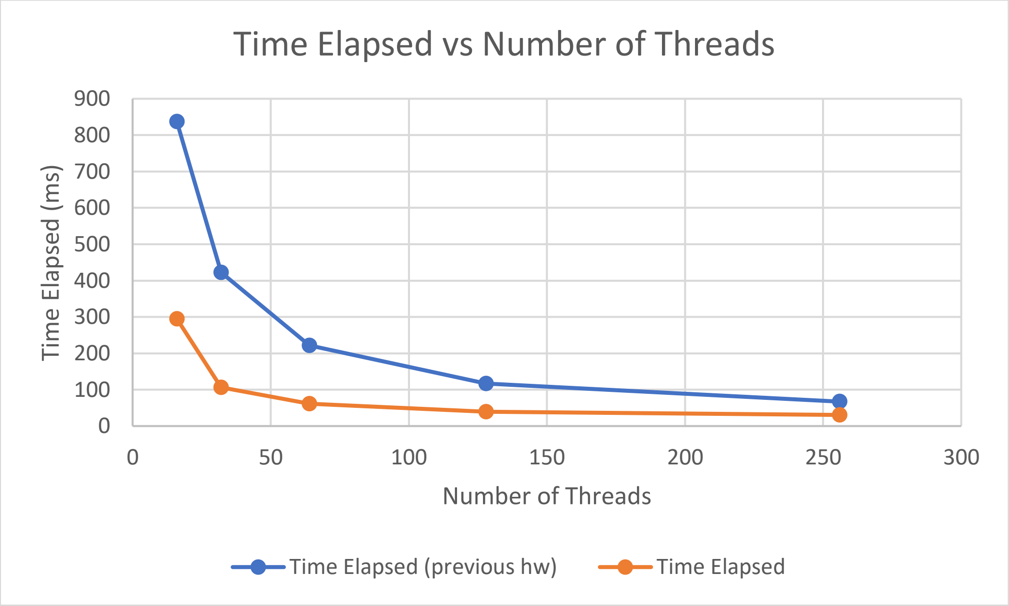
Homework 11 – CUDA Blur Image with Shared Memory

Jarod Klion

April 12th, 2022

1. Object of the project:
   a. Use CUDA and shared memory to redo the previous OpenMP project of blurring an image.
2. Details:
   a. Use CUDA functions and 2D blocks which use communication between CPU and GPU to parallelize the program with a different method. Start by letting the host read the image, then allocate needed memory in the device. Copy the colors from the host to the device, launch the kernel, and let each thread on the device convert one pixel. Afterwards, copy pixels back from device to host and write the file. I found the best times came at 256 blocks with 256 threads called for the kernel function.
3. Results (256 blocks only):
   a. Number of Threads: [16, 32, 64, 128, 256]
   b. Time Elapsed in last hw (ms): [837.4185, 422.2359, 221.9645, 116.476, 67.2156]
   c. Time Elapsed (ms): [295.0879, 106.1694, 61.0798, 39.2596, 30.5968]



4. Performance Improvements:
   a. I am sure using the ghosting method given in both the example code and slides for class could lead to even greater speedups in comparison to the previous code.
5. Bugs Encountered:
   a. For some combinations of blocks and threads, diagonal lines of black dots appear every so often periodically throughout the entire image. Additionally, this blur is not as strong as the previous assignment's blur because I don't think I go through all pixels in the image correctly using 2D shared memory.