AdvancedHPC labwork 4

Hai Nguyen Ngoc

October 2024

1 Introduction

This project aims to compare the performance of 1D and 2D GPU implementations of RGB to grayscale image conversion using CUDA. The goal is to observe speed improvements over a baseline CPU implementation and investigate how different block sizes influence GPU performance.

2 Implementation

2.1 CUDA kernel

The 1D GPU Kernel: Explained in labwork 3.
The 2D GPU Kernel (rgb_to_grayscale_gpu_2d) organizes threads in a 2D grid structure that mirrors the 2D nature of the image data. Each thread processes a single pixel directly within the 2D grid, allowing it to access memory more efficiently

2.2 Block sizes

To maximize GPU performance, different block sizes were tested for both 1D and 2D implementations. Block sizes of 8×8, 16×16, and 32×32 were chosen, allowing sufficient threads per block while balancing memory constraints. Each block size was applied in both 1D and 2D GPU kernels to observe the effect on execution time. A larger block size allows for more work to be completed in parallel, improving GPU utilization.

3 Experiment

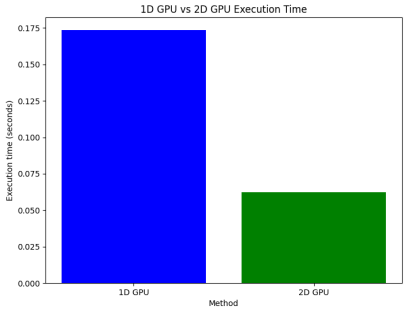

Figure 1: Execution time on laptop

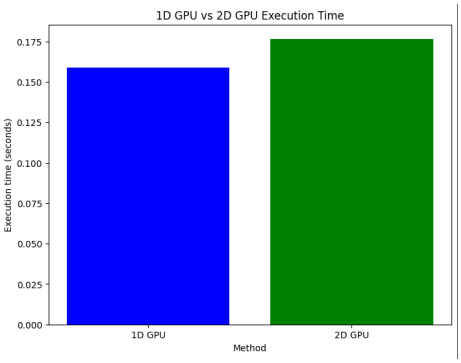

Figure 2: Execution time on colab

On my laptop, the execution time of 2D GPU is better than 1D, but worse on Google Colab.
Explanation: Laptop GPUs may have fewer but faster cores suited for tasks with high spatial locality, making the 2D kernel more efficient for certain image-processing workloads. Conversely, Google Colab's GPUs, typically designed for high-volume parallel processing (deep learning for example), may have higher thread counts but slightly slower individual core speeds.

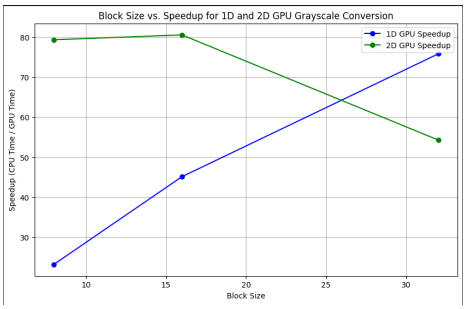

Figure 3: Block sizes versus speedup

For smaller block sizes (8 and 16), the 2D method (green line) is faster than the 1D method (blue line), meaning it uses the GPU more effectively. However, as the block size grows to 32, the speed of the 2D method decreases, while the 1D method keeps getting faster. It means larger block sizes work better for the 1D method but cause slowdowns in the 2D method, possibly due to extra memory or coordination work.

4 Conclusion

- 2D blocks are better than 1D because of memory access patterns, Parallel Processing(allowing more threads to run parallelly).
- The appropriate selection of block size is good for performance of GPU.