Abstraction

Image convolution is the process of applying a small mask or convolution matrix to an image to obtain a desired effect–i.e., sharpening, smoothing, blurring or edge detection. Depending on the size of the image, convolving can be a high-compute intensive task as it requires multiple operations for every pixel of the image. This process is intrinsically well-suited for parallelization, since each operation is independent from the rest. This assignment involves leveraging the raw power of GPGPUs using NVIDIA’s CUDA framework to optimize the image convolution process, and analyze the performance gains with various GPU optimization.

Introduction

The purpose of this assignment is to translate the serial code that was provided to us using the CUDA framework. The CUDA programming model created by NVIDIA provides a relatively easy way to program for GPU architectures so that we can leverage the massive computing power that GPUs have to offer.

The serial code is from program-2 that computes the horizontal and vertical gradients. The objective of program-5 is to translate the serial code to run on the GPU architecture via CUDA with various levels of optimization.

Implementation

Optimization 1

The first optimization involves solely the GPU kernel. We perform the convolution on the GPU and transfers the image into the global memory of GPU device. Each block has 16 threads and the number of thread blocks depends on the size of the image. Thus, insuring that each thread act on a single pixel.

Optimization