GPU Computing Report

Kaustav Vats (2016048)

Introduction

Convolution is a techniques used in Image Processing. Convolution let you do many things like detect edges, apply blur, apply filter etc. All of this is done using convolution **Kernel.** It's a small matrix, in which the center element is called anchor. Kernel slides over the image, anchor point is used to determine position of the kernel with respect to the image.

Strategy

To increase speedup of program, we can use shared memory to access, pixel at faster rate. Since multiple thread require a single pixel value, so we can store it in much faster and better memory called shared memory. But limitation is that, each block get some amount of shared memory. So you can only use some variable to improve computation time.
The main idea was to load some part of the image into shared memory and perform computation over that, then load again with new values and compute again. So we can create a shared memory 2d array of size block size + 2*kernelRadius -1. We will load all pixel values and run our kernel over block of a GPU. By this we will be able to do convolution of all pixels in the block/Tile.
In code, each thread load twice into the shared memory. To cover pixels that lie outside the block.

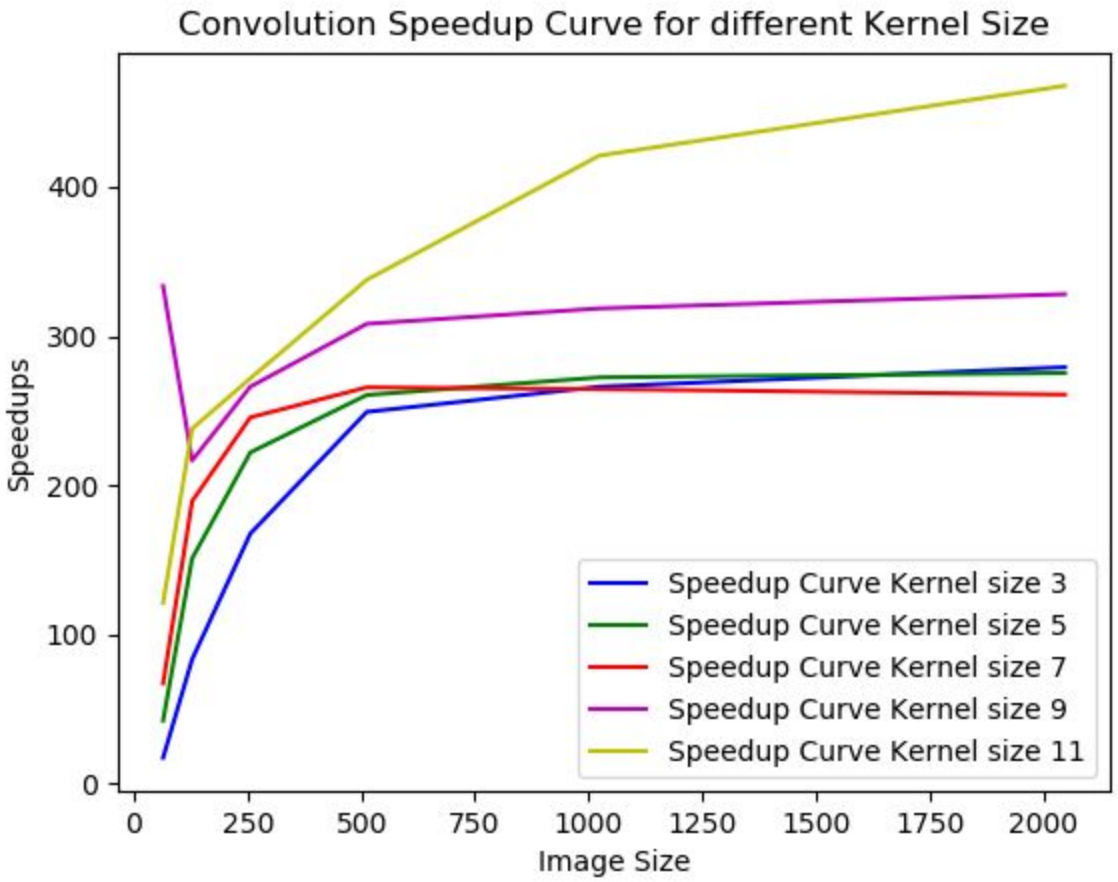
__restric__ is used to notify compiler that the data is read only. So that the compiler doesn't load the variable again.
Link Ref:- https://devblogs.nvidia.com/cuda-pro-tip-optimize-pointer-aliasing/

Problems Faced

1. Initially i wrote a cuda C++ program without shared memory. Later looked for parallelism that can be added to the code.
2. Then i tried to put whole image in shared memory, but later realized that shared memory is not enough for image of size 2048. So then i went for the above approach.

Result

Speedup x Image Size Curve



Actual Values

Kernel 3	`[17.48, 83.5008, 167.7, 249.205, 266.049, 279.15]`
Kernel 5	`[42.4391, 151.097, 222.049, 260.524, 272.326, 275.302]`
Kernel 7	`[67.2689, 189.466, 245.54, 265.623, 264.458, 260.811]`
Kernel 9	`[333.551, 216.654, 266.094, 308.191, 318.506, 328.045]`
Kernel 11	`[121.262, 238.01, 271.682, 337.681, 420.93, 467.673]`
Image Size	`[64, 128, 256, 512, 1024, 2048]`

Sobel Edge Detection

Taking magnitude of gradient in x and y direction.

Links ref: http://www.doc.ic.ac.uk/~ts2615/contribution.html

Result for Image512.png