# Lab 3 Report

How many floating operations are being performed in your color conversion kernel? EXPLAIN.

Calculating the greyscale intensity uses the following formula: Intensity = 0.21f \* r + 0.71f \* g + 0.07f \* b; each multiplication is one flop and each addition is one flop so each kernel performs 5 floating point operations per pixel in an image. There are also 2 integer operations to calculate the gray offset and then the rgb offset.

How many global memory reads are being performed by your kernel? EXPLAIN.

3 Global memory reads are being performed per pixel. One read for the Red Green Blue channels on the pixels.

How many global memory writes are being performed by your kernel? EXPLAIN.

3 Global memory writes are being performed to set the Red Green Blue channels to the calculated greyscale intensity.

Describe what possible optimizations can be implemented to your kernel to achieve a performance speedup.

Modify the Kernel so that each kernel only handles one channel of one pixel. That way each thread would perform one global read, one floating point operation, one global write.

Name three applications for color conversion.

1. As a pre-processing step before running an edge detection algorithm
2. Compression of data to speed up other algorithms like face recognition
3. A pre-processing step before running optical character recognition algorithms

How many floating operations are being performed in your blur conversion kernel? EXPLAIN.

None, there are however 5 per pixel that’s being averaged and another 5 to averaged them. In a 3x3 grid, assuming the edge of the grid is inside the bounds of the image, there will be 50 integer operations per kernel.

How many global memory reads are being performed by your kernel? EXPLAIN.

Each pixel that needs to be blurred reads it’s own value and the surrounding 8 from memory. For each pixel we need to read 3 rgb channels so 27 total memory reads per kernel.

How many global memory writes are being performed by your kernel? EXPLAIN.

3 memory writes per kernel because it only writes 3 rgb channels to the pixel in the centre of the 3x3 blur grid.

Describe what possible optimizations can be implemented to your kernel to achieve a performance speedup.

Instead of looping in the kernel, use one thread per surrounding pixel and one thread per channel for a total of 27 threads working on one pixel in the source image. This could be done by having every thread simply add to the total value in the middle pixel, then using \_\_syncthreads(); and having one of the divide the rgb channels by 9. Or having another kernel run after that just divides all channel values by 9.