Lab 7 - Reduction

Nguyen Duc Tung

This lab objective is to stretch the greyscale of an image. This includes two steps:

1. Find max/min intensity
2. Recalculate the intensity for each pixel

Step 2 can be easily done by inplement Map using a formula. But step 1 will be done by a Reduction to find the max/min. The reduction is done and optimized by following the instructions on the lecture. However, 1 block can only processed a small part of the images, we can not do 1 reduction and get the global result. Therefore, we need a blocks synchronization method call the reduction kernel again and again, until we reach only 1 output.

My idea is to used two pointers, 1 for the input, 1 for the output. After each kernel run finished, the pointers is swapped, so that output of this run will be the input of the next run. And it will continually invoking the kernel until we get the desired max/min. Following snippet of code explains my approach:

```
1 while (reduceGridSize > 1) {
2     getMaxIntensity<<<reduceGridSize, blockSize,
    threadsPerBlock>>>(maxArrayPointer[swap], maxArrayPointer
    [!swap]);
3
4     reduceGridSize = (reduceGridSize + threadsPerBlock - 1) /
     threadsPerBlock;
5     swap = !swap;
6 }
```

Here is an image with greyscale stretched:

Time elapsed: 29.9 ms

(a) Original image

(b) Greyscale stretched