**CSC8820**

**Advanced Graphics Algorithms**

Spring 2018

**Project 4**

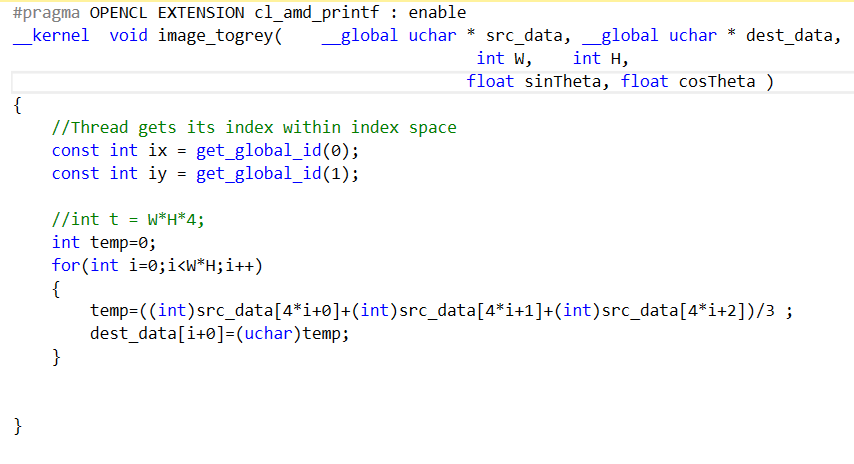
Due date: 11:59 pm **04/27/2018**

Requirements

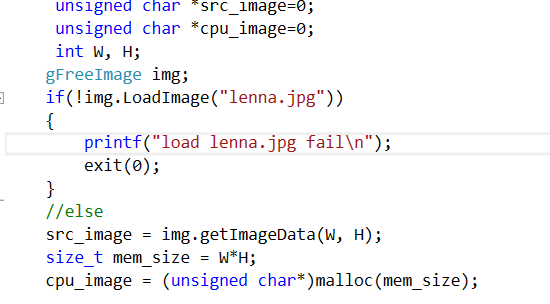
In this project you will learn how to write an OpenCL program for simple image processing.

Requirements:

1. Your program should read a color image, convert it to a grayscale image, and save the grayscale image to a file.
   1. The algorithms for converting color to grayscale can be found [here](http://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/) or [here](http://www.tannerhelland.com/3643/grayscale-image-algorithm-vb6/).



* 1. You must use one of the two image loading libraries: [CImg](https://github.com/dtschump/CImg) or [FreeImage](http://freeimage.sourceforge.net/). This will make it easier for my TA and me to build your program.



Source JPG:



GPU Des JPG:  


CPU Des JPG:

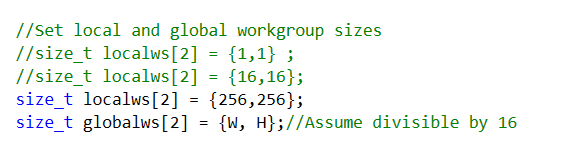


1. Implement both a parallel version of this algorithm in C/C++ and OpenCL, and also a sequential version in C or C++.
   1. Compare the performance of the parallel version and the sequential version. Is the parallel version faster than the sequential version?

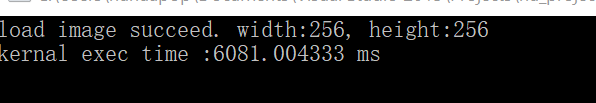
Yes, parallel version is faster than the sequential version

Because set size\_t localws[2] = {1,1} , this means the program is sequential to deal with every pixel. This cost 6081ms. When set size\_t localws[2] = {256,256} , this means the program is parallel to deal with every pixel. This cost 26.79ms.

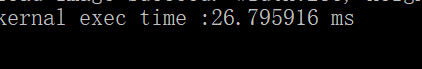
Set group size:



Group size = 1



Group size = 256



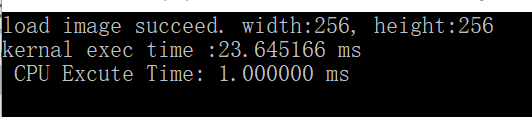
* 1. Vary the number of work items. Does it make a difference in performance (i.e. time)?

Yes, if the number of work items increases, the performance will be large.

Above is detail to this problem.

* 1. Compare the performance of the parallel version of your algorithm on CPU and GPU. Which one is faster?

CPU is faster. Because the GPU of my computer is Intel HD Graphics 520 which depends on CPU 6200U.



* 1. Include a brief report to describe your experiments and results.

My experiments contain the following functions:

<1>. Load image

<2>.Transform image to grey by GPU with different numbers of work items (max 256, min 1 which means sequential)

<3> Transform image to grey by CPU

1. Upload your source code and report in a ZIP file to iCollege under the folder Project 4. Do not submit via email.