Programming Assignment VI:OpenCL Programming

The purpose of this assignment is to familiarize yourself with OpenCL programming.

1 Problem Statement

A histogram is a statistic that shows frequency of a certain occurrence within a data set. The histogram of an image provides a frequency distribution of pixel values in the image. If the image is a color image, the pixel value can be the luminosity value of each pixel or the individual red (R), green (G), and blue (B). More about image histogram can be found at http://en.wikipedia.org/wiki/Image_histogram.

In this problem, you need to use OpenCL to parallelize the implementation of the image histogram. The input/output is a bitmap image format, which stores R/G/B values of each pixel. See the reference code for how to read/write a bitmap file.

Below shows a serial implementation of the image histogram. The complete implementation can be downloaded at http://people.cs.nctu.edu.tw/~ypyou/courses/PP-s19/assignments/HW5/image-histogram.cpp.

```
void histogram(Image *img,uint32_t R[256],uint32_t G[256],uint32_t B
    [256]){
    std::fill(R, R+256, 0);
    std::fill(G, G+256, 0);
    std::fill(B, B+256, 0);

    for (int i = 0; i < img->size; i++){
        RGB &pixel = img->data[i];
        R[pixel.R]++;
        G[pixel.G]++;
        B[pixel.B]++;
    }
}
```

2 Input/Output

2.1 Input

The program takes at least one command argument, which indicates the input file name.

```
./histogram test1.bmp test2.bmp test3.bmp
```

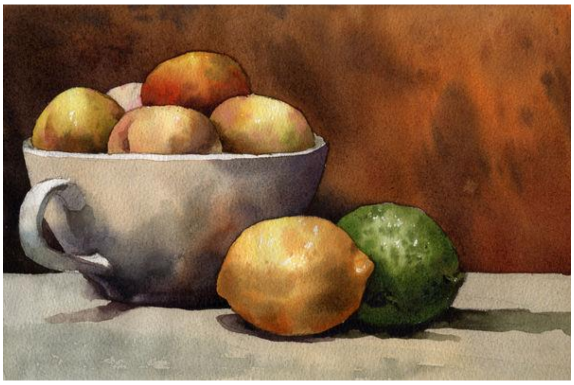

Figure 1: A sample image

2.2 Output

You will need to output a processed file, named `hist_<input_filename>.bmp`, for each input file. You can use the reference code to output histogram file(s) directly.

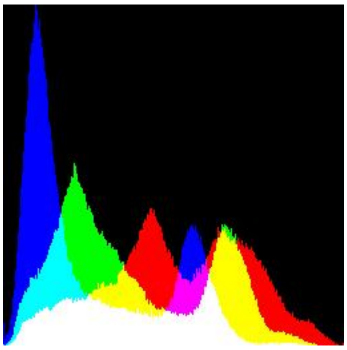

Figure 2: the histogram of sample image

3 Requirements

- Your submitted solution contains two source files, which are named `histogram.cpp and histogram.cl`.
- The `histogram.cl` must be the OpenCL kernel that you wrote.

4 Development Environment

4.1 Building the OpenCL environment on your own computer

If you have a nVidia or AMD GPU, you can build your own development environment by installing the corresponding SDK.
https://developer.nvidia.com/cuda-downloads
http://developer.amd.com/tools-and-sdks/heterogeneous-computing/amd-accelerated-parallel-processing-app-sdk/downloads/

4.2 Using CUDA/OpenCL Server in SSLAB

We have set up four servers (the same servers as described in the last assignment) for this assignment. You can login to one of the servers to work on your assignment. Each server contains a GeForce GTX 1060 GPU. TAs will grade your implementation on these servers, so please make sure your implementation works on the provided servers.

4.2.1 Login

Server IP: 140.113.215.195
Port: 37106, 37107, 37108, 37109
Account: `pp<student ID>`
Default Password: `pp<student ID>`

4.2.2 Compilation

```
g++ histogram.cpp -o histogram -lOpenCL
```

If the message below shows, do not bother with it.The OpenCL environment works as expected.
"/opt/cuda/lib64/libOpenCL.so.1: no version information available"

5 Submission

Be sure to upload your zipped source codes, which includes no folder, to e-Campus system by the due date and name your file as ”HW5_xxxxxxx.zip”, where xxxxxxx is your student ID.

Due Date: 23:59, June 10, 2019

6 References

- http://www.kimicat.com/opencl-1/opencl-jiao-xue-yi
- http://www.khronos.org/opencl/