APPLIED PARALLEL PROGRAMMING ON GPU

Assignment 3

thrust::Automatic Contrast Enhancement

Deadline: 14 May 2019

1. Prerequisites:

1. You may refer to your lecture notes and the textbook for the GPU programming concepts and optimization issues.

2. Assignment Requirements:

In this assignment, you are expected to implement automatic contrast enhancement for 8-bit grayscale images as in Assignment 2, this time you are expected to use the thrust library for implementation and the code must work for any image size up to HD.

1. The algorithm works as follows:
   - Go through all the pixels in the image and find the minimum value (*nMax*).
   - Go through all the pixels in the image and find the maximum value (*nMin*).
   - Apply the following operation to all the pixels (*i,j*) for contrast enhancement *pDst*: output image, *pSrc* = input image

$$pDst(i,j) = \frac{pSrc(i,j) - nMin}{nMax - nMin} x\ 255$$

As you can see the operation could be divided into 3 parts:

(i) Finding *nMin* and *nMax*

(ii) Subtracting *nMin* from the pixels of the source image *pSrc*

(iii) Scaling the pixels of source image *pSrc* by a factor of $\frac{255}{nMax-nMin}$

2. Have a look at the code supplied with assignment 2, this code has been implemented using NVIDIA Performance Primitives (NPP) library. In the main.cpp you'll see that these operations have been done using the NPP library. You can run this code and observe the results. The code reads "lena_before.pgm" file and writes the output into "lena_after.pgm" file. You can install and view pgm files using IrfanView: http://www.irfanview.com/

3. `Npp8u` type is basically 8-bit unsigned chars used in NPP library (`typedef unsigned char Npp8u`).

4. Create a new file by copying the main.cpp file and change these operations to run using **thrust library**:
   - Replace the `nppiMinMax_8u_C1R()` operation and find the minimum and maximum values in the image.
   - Replace the `nppiSubC_8u_C1RSfs()` operation and subtract the minimum value that you found in the previous step from all the pixels in the image: $I_{in} - nMin$.
     Replace the `nppiMulC_8u_C1IRSfs()` operation multiplying all the pixel values with the calculated constant $\frac{255}{nMax-nMin}$.
     Hint:You can do this by integer arithmetic, please have a look at GTC 2013 Lab NPP.pptx You can use the calculation already in the code. You need to multiply by *nConstant* and then divide by *nScaleFactor*-1.
   - Make sure that you have the same results in lena_after.pgm with the original implementation.

5. Now you are expected to do benchmarking to compare the original code and your implementation. You need to run 10 iterations and take the average for the final result. You should use the provided 512x512 Lena image (lena_before.pgm). You can use Lena image for benchmarking but your code should work for any image size up to HD.
   Hint: Use QueryPerformanceCounter function in windows for high precision timing for CPU, see the following link for an example:
   http://stackoverflow.com/questions/1739259/how-to-use-queryperformancecounter

Important**:** *Please write what you observed in your implementations and the experiments,* ***not just the information written in the books or the lecture notes.*** *If your observation results are different from the information in the books or in the other published documents, please write them and try to analyze this difference.*

3. Documentation and Submission Requirements:

While preparing your assignment documents, please consider the following issues:

1. The files that include the implementation codes, must be explained briefly. For example, briefly explain the functionalities of the files that will run on GPU and/or CPU.

2. Explain your implementation methodologies briefly, such as the management of the memory operations and the thread operations, the determination of the block numbers (if you use the block-matrix operations), etc. You should consider the architectural constraints of GPU.

Note that, if you just write "*I did and it works*", you will **NOT** get any points. You should explain your reasons (I mean you should support your hypotheses).

Please introduce your solutions either with tables or figures that are the visualizations of the tables. Moreover, you are required to explain each of the tables and the figures in your document.

Remember that there are no strict rules or algorithms in order to write an academic document. You should use your imagination and academic skills.

Hint: You may consider writing an assignment document or any other academic paper as talking with the reader or presenting your studies in a conference to the reader. But you should remember that the reader can not ask questions to you, if he/she does not understand some of the concepts (or points) in your study. Therefore, you should help to him/her.

4. If your implementation requires specific configuration settings (such as the configuration of path settings), please indicate them in your document.

4. Submission and Grading Policy

Assignments will be submitted via ODTUClass online .You are required to submit the code and a report.

Please submit codes as plain text files or ".cu" files. Do not submit the whole project. Codes will be tested on Visual Studio 2017, Windows 10.

Submit the report in pdf format.

You are expected to work individually NOT in groups. You will also be expected to follow the academic integrity rules.

Policy for Copying: ***Passing the work of others (either from another student or a code on internet etc.) off as your own is a breach of academic ethics and also of the University's disciplinary rules. When you submit a work it automatically implies that you claim the ownership of the work.***

Note that METU is subscribed to some tools which allow cross checking of submitted works as well as checking with any work on internet or any university subscribed to the system. No exceptions will be allowed and any work found to be copied will result in failing the course.

Please send your questions about the assignment to congun@metu.edu.tr or you may post a message to the forum on ODTUClass.