UNIVERSITY OF
LINCOLN

Lincoln School of Computer Science

Assessment Item Briefing Document

Title: CMP3752M Parallel Programming, Assessment Item One, Coursework	**Indicative Weighting: 40%**

Learning Outcomes:

On successful completion of this assessment item a student will have demonstrated competence in the following areas:

- [LO1] demonstrate practical skills in applying parallel algorithms for solving computational problems;
- [LO3] analyse parallel architectures as a means to provide solutions to complex computational problems.

Requirements

Your task is to develop a digital image enhancement tool which will perform contrast adjustment using the histogram equalisation algorithm. The algorithm uses a cumulative intensity histogram to back-project original image intensities resulting in an image of equalised intensities. The algorithm is described in detail in [1] with the main calculation steps summarised below and in Fig. 1:

- calculate an intensity histogram from the input image (Fig. 1b);
- calculate a cumulative histogram (Fig. 1c);
- normalise and scale the cumulative histogram (Fig. 1d): the cumulative frequencies are normalised and scaled to represent output image intensities (e.g. from 0-255 for an 8-bit image);
- back-projection: the normalised cumulative histogram is used as a look-up table (LUT) for mapping of the original intensities onto the output image. For each output pixel, the algorithm should use the original intensity level as an index into the LUT and assign the intensity value stored at that index.
- the output should be an intensity equalised image (Fig. 1e).

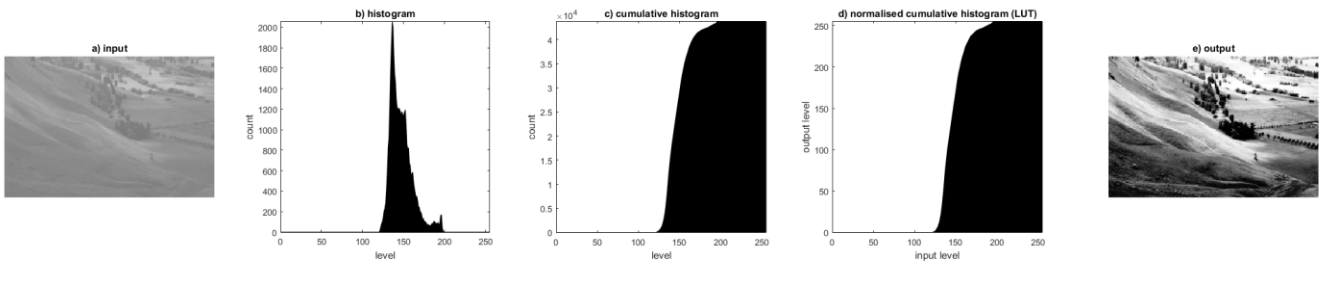

Fig. 1 The individual steps of the histogram equalisation procedure.

Due to a potentially large amount of data, all image processing shall be performed on parallel hardware and implemented by a parallel software component written in OpenCL. You should develop your own device code (i.e. kernels) which performs the main steps of the algorithm. The steps include several classic parallel patterns including scan, histogram and map. Your tool should also report memory transfer, kernel execution and total program execution times for performance assessment. All host-side image I/O operations (i.e. reading, displaying of images) are not assessed and can be based on the code provided during the workshop tutorials. You can use all the code and libraries provided and developed during the workshop sessions, but you must not use any additional and pre-existing libraries for this purpose.

The basic version of the tool can be based on purely atomic calculations of the intensity and cumulative histograms, work with 8-bit intensity monochrome images and a pre-determined number of bins (i.e. 256). Further credit will be given for additional functionality such as the efficient parallel implementation of the histogram based on local memory, variable bin number/size, different parallel variants of the scan and support for colour images. In such a case, your program should run and display execution times for different variants of your algorithms. The test images of different size are provided together with this assessment brief. During the demonstration, you might be asked to demonstrate your code working on an “unseen” image provided by your instructor. The main assessment criteria for this task are related to the correctness of the developed parallel algorithms and the effectiveness of optimisation strategies. The code should be well commented and clearly structured into functional blocks.

References:

[1] Rafael C. Gonzalez, R.C. and Woods, R.E (2018), Digital Image Processing. Pearson.

Useful Information

This assessment is an individually assessed component. You will be required to demonstrate your developed code and answer questions related to your submission during a dedicated workshop session in the computing labs. Only the submitted code will be considered during demonstrations so further developments occurring after the deadline will not be taken into account. Non-attendance at the scheduled demonstration session will automatically result in zero mark for this assessment item. The demonstration date and further details will be provided during lectures. Please make sure you have a clear understanding of the grading principles for this component as detailed in the accompanying Criterion Reference Grid. If you are unsure about any aspect of this assessment component, please seek the advice of a member of the delivery team.

Submission Instructions

The deadline for submission of this work is included in the School Submission dates on Blackboard.

You must make an electronic submission of your work by using the assessment link on Blackboard for this component. All developed code should be submitted as a single ZIP file. It is important that you check that your .ZIP file decompresses to a Visual Studio project that can be directly built and executed on the computers in the computing labs. This will be required during the final demonstration of your work. Do not include the test images in your submission as these are relatively large! There is no written report required for this submission.

DO NOT include this briefing document with your submission.