Assessment Item 1

Module Code & Title: CMP3752M Parallel Programming

Contribution to Final Module Mark: 40%

Description of Assessment Task and Purpose:

Your task is to develop a digital image enhancement program that 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).

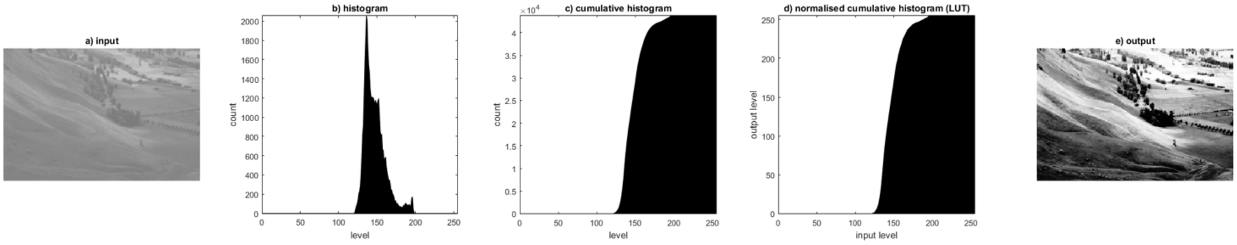

Fig. 1 – *The individual steps of the histogram equalisation procedure.*

Due the large amount of data, all image processing must be performed on parallel hardware and implemented by parallel software written in OpenCL with C++. You should develop your own device code (i.e. kernels) that perform the main steps of the algorithm. The steps include several classic parallel patterns including scan, histogram, and map. Your program 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 may use any code and libraries provided/developed during the workshop sessions, but you must not use any additional pre-existing libraries for this purpose.

The basic version of your program can be based on purely atomic calculations of the intensity and cumulative histograms, work with 8-bit intensity monochrome images, and use a pre-determined number of bins (i.e. 256). Further credit will be given for additional functionality such as efficient parallel implementation of the histogram based on local memory, variable bin number/size, different parallel variants of the scan pattern, and support for colour images. In such a case, your program should run and display execution times for different variants of your algorithms.

Two test images of different sizes are provided together with this assessment brief. Your code will also be tested on "unseen" images, such as colour and 16-bit depth examples to test the extra functionality of your program. 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.

You are also required to provide a short summary of your implementation (max. 300 words), which should be included as a code comment section above the main function in your host code. This should briefly describe the implemented functionality, clearly indicate any original developments in the code (beyond what was provided in the workshops), cite any external sources used, and detail any optimisation strategies employed. Note that your grade will solely depend on the quality of your code, so keep this section concise and do not exceed the word count.

References:

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

Learning Outcomes Assessed:

- **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

Knowledge & Skills Assessed:

Subject-specific Knowledge, Skills, and Understanding:

- Understanding of parallel programming patterns
- OpenCL parallel programming skills

Professional graduate skills:

- Analytical skills
- Critical thinking
- Problem solving
- Time management

Emotional intelligence skills:

- Self-management
- Motivation

Assessment Submission Instructions:

The deadline for submission of this work is included in the school submission dates on Blackboard. All developed code should be submitted as a single ZIP file via the upload area for this assessment item. Please remove all compiled binary and temporary files to reduce the size of the overall submission (folders such as x64, .vs, debug, release, etc).

Format for Assessment:

The submitted work should comprise of OpenCL code written in C++.
DO NOT include this briefing document with your submission.

Feedback Format:

Written feedback will be provided via Blackboard.

Additional Information for Completion of Assessment:

This assessment is an individually assessed component. Your work must be presented according to the School of Computer Science guidelines for the presentation of assessed written work.

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 advice from a member of the delivery team.

Assessment Support Information:

- Staff are available during their office hours and can provide feedback during this time outside of module hours.

Important Information on Dishonesty & Plagiarism:

University of Lincoln Regulations define plagiarism as 'the passing off of another person's thoughts, ideas, writings or images as one's own...Examples of plagiarism include the unacknowledged use of another person's material whether in original or summary form. Plagiarism also includes the copying of another student's work'.

Plagiarism is a serious offence and is treated by the University as a form of academic dishonesty. Students are directed to the University Regulations for details of the procedures and penalties involved.

For further information, see www.plagiarism.org