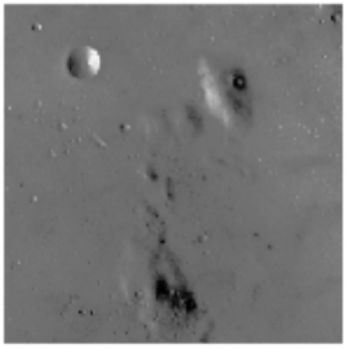
CSE 560 - Winter 2022
Assignment 1: Adaptive Histogram Equalization
Due date: 23:59:59, 9th February 2022

In image processing, histogram equalization is used to automatically enhance contrast of images. See: https://en.wikipedia.org/wiki/Histogram_equalization

Global histogram equalisation has a tendency to overexpose certain regions of image. Adaptive Histogram Equalization (AHE) performs a local contrast enhancement. See https://en.wikipedia.org/wiki/Adaptive_histogram_equalization



Original image

Histogram equalized

Adaptive histogram equalized

Go through the above links to understand the method and read the code to understand the provided CPU implementation of AHE. Compile (use CMAKE) and run the given code on two images. Note the time taken by the CPU implementation.

1. Write a CUDA version of the AHE computation (implement both steps; see CPU code). Use global memory for storing all images and arrays. [25 marks]
2. Change your code to use constant memory to store computed mappings. [10 marks]
3. Submit a report with the following (all parts below are mandatory):
   (a) Document your approach to the problem in both case (1) and (2) above. [5 marks]
   (b) Add output images
   (c) Vary the TILE_SIZE (both in X and Y together) to be (256, 512, 1024, 2048, and 4096) and present speedup analysis:
      (i) Report CPU vs GPU timing results for both images in tabular format. Discuss if using constant memory improved performance of your code or not. [5 marks]
      (ii) Plot the graph for speedup *S*, where
      $S = \frac{Time\_CPU}{Time\_GPU}$. [5 marks]

Total marks for this assignment: 50 marks

Bonus marks up to a maximum of 10 can be awarded for the following (this part is completely optional):

4. Use texture memory to read the image. Report CPU vs GPU (kernel and overall) timing results for size variations mentioned in question 3(c). [5 marks]
5. In question (1) above, what would be effect on performance (i.e, speedup) when TILE_SIZE_X and TILE_SIZE_Y are varied independently? You may use the sizes mentioned in question 3(c) and vary tile sizes in X and Y independently (i.e, vary X size while keeping Y size fixed) thus producing a 5 x 5 matrix of speedups. Analyse your results and discuss how 2D spatial locality plays a role here (or not). [5 marks]

Note: A report is mandatory along with code submission to receive any credit.

Disclaimer: Your code should be written by you and be easy to read. You are NOT permitted to use any code that is not written by you. (Any code provided by the TA/ instructor can be used with proper credits within your program).