CSE 560 - Winter 2022
Assignment 2: Signed Distance Transform
Due date: 23:59:59, 18th Feb. 2022

Signed distance transform (SDT) of a shape $\mathcal{O}$ is a scalar field function $f : \mathbb{R}^2 \mapsto \mathbb{R}$ such that any point $f(x, y)$ represents the signed shortest distance from the point (*x, y*) to the boundary $\partial\mathcal{O}$. Points inside the shape have a positive distance value and vice versa. An SDT is quite useful in scientific computing problems such as skeleton generation, robot motion planning, and path finding. It can be seen that computing *f* at a point requires knowing the entire boundary information (thus it is a global operation). The attached code computes *f* in a brute-force fashion by first calculating distance values for a pixel from all boundary pixels and then keeping the minimum value (of distance magnitudes). Understand the provided CPU code and realizing that the given problem is embarrassingly parallel in nature (and painfully slow in a serial implementation), speedup computing *f* using CUDA. The provided CPU code has already been made parallel on the host with OpenMP.

1. SDT computation on GPU
   (a) Write a CUDA version of the SDT computation using shared memory. [15 marks]
   (b) Document your approach to the problem. [5 marks]
   (c) Perform computations on CPU and GPU with image of sizes (256, 512, 1024, and 2048). Tabulate CPU and GPU (kernel and overall) timing results, plot speedups (kernel and overall), and report the MSE error in each case. [5 marks]
2. How will you modify your approach to use constant memory instead of shared memory? Explain why using constant memory instead of shared memory is a good/bad choice in this case. [5 marks]
3. Kernel analysis
   (a) Analyze your CUDA kernel in terms of efficiency using nvprof/nvvp tool. [2 mark]
   (b) Identify bottlenecks in your kernel. [3 marks]
4. Kernel optimisation
   (a) Rewrite a better version of your kernel based on your analysis (for example: improve occupancy, math performance, register/shmem usage, bank conflicts, coalescing, etc.). [10 marks]
   (b) Document your optimization strategies. Compare and plot new speedups (kernel and overall). [5 marks]

Total marks for this assignment: 50 marks

Bonus marks to a maximum of 5 can be awarded for the following:
5. (a) Use texture memory as an alternative to shared memory. [3 marks]
   (b) Document your observations and compare performance with your optimized shared memory version of the kernel. [2 mark]

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).