

CS 677: Homework Assignment 4
Due: April 3, 6:00pm

Philippos Mordohai, Department of Computer Science
Philippos.Mordohai@stevens.edu

Collaboration Policy. Homeworks will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in understanding the material but not in solving the problems. Use of the Internet is allowed, but should not include searching for previous solutions or answers to the specific questions of the assignment. I will assume that, as participants in a graduate course, you will be taking the responsibility of making sure that you personally understand the solution to any work arising from collaboration.

Late Policy. No late submissions will be allowed without consent from the instructor. If urgent or unusual circumstances prohibit you from submitting a homework assignment in time, please e-mail me explaining the situation.

Submission Format. Electronic submission on Canvas is mandatory. **Please submit a zip file containing a pdf file and your code.**

Problem 1. (100 points) For this problem, follow the steps below.

1. Download the zip file and extract its contents into your SDK projects directory.
2. Edit the source file `sobel.cu` file to implement the Sobel filter as shown in the slides of Lecture 7. The input image can be of any size in binary PPM format. Only the red channel will be used.
3. There are several modes of operation for the application.
 - No arguments: The application will read the default image and use the default threshold value specified in `sobel.cu`.
 - One argument: The application will read the default image and read the threshold value from the command line (as the first argument).
 - Two arguments: The application will read the user specified file and threshold.
4. In all cases, the program will apply the Sobel filter on the image and generate an output binary image by thresholding the magnitude of the filter's response. The computation will be performed on both the CPU (code already provided) and the GPU and the results will be compared using the CPU as gold standard. A one-pixel wide border around the image can be ignored and set to black in the output.

In your report:

1. Explain how you increased memory bandwidth over the naive GPU version that only uses global memory.
2. Explain why you selected 1D or 2D structure for your blocks.
3. Describe your implementation for fetching pixels at the boundaries of a block that are needed for computing the filter for interior pixels.

Note 1: Solutions that require padding the image with zeros will not receive full credit, but will not be severely penalized (10%).

Note 2: Solutions using global memory only will receive little credit (30%).