

MultiCore Computing

Lecturer: Hajar Falahati

Spring Semester 1400-1401



Project

Deadline 1401/05/03

1 Who are you?

Please write down the name and the student id of each team member in the following format:

- Team member No.1: <Name>, <Student id>
- Team member No.2: <Name>, <Student id>

Your group should submit one of the followings ...

1.a Edge Ditection Filter

The **Sobel filter**, is used in image processing and computer vision, particularly within edge detection algorithms where it creates an image emphasising edges. You can learn about how it works here. Your task is to implement the sobel edge detection algorithm using CUDA.

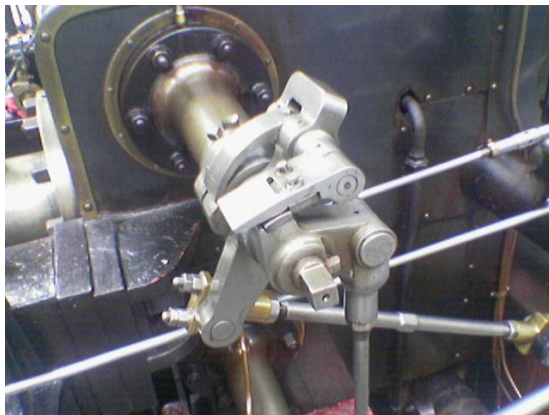


Figure 1: Original Image

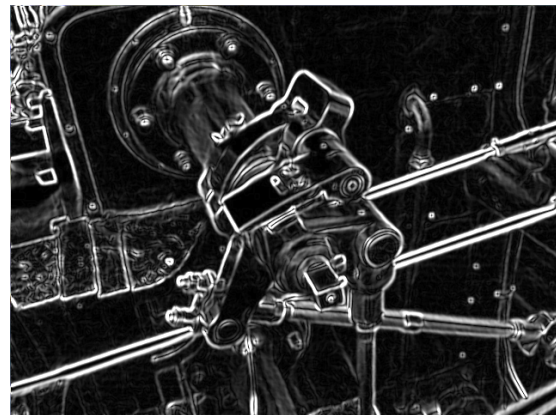


Figure 2: After Sobel Filter

- Your implementation should be able to return an edge detected image, using sobel algorithm, only on monochrome images. You are free to use any image format (e.g. *.pbm, *.pgm, *.jpg, *.png).

- Your implementation should also be able to change brightness of the input before executing the algorithm. The output image should differ depending on the brightness of the input. In case of CLI implementation, modified input should also be presented alongside its edge-detected result.
- Your code Should work for any size of image (extra points for 8K). Demonstrate this capability using a variety of inputs.
- Extra points are given, should you implement threshold for sobel algorithm.
- Extra points are given for interactive GUI, over plain CLI.
- Your code should run faster than single-threaded implementations (you should demonstrate speedup).
- Attach a report which reports all your work and results.
- You can't use computer Vision libraries such as **OpenCV** or **OpenCL** except for image file reading and writing.
- Extra points for any improvement in CUDA implementations (e.g. techniques are mentioned in **CUDA programming** slide like removing divergent branch, removing bank conflicts, and unrolling the warp). You should mention all your work in the document.

1.b PageRank

PageRank (PR) is an algorithm used by Google search to rank web pages in their search engine results. It's a way of measuring the importance of website pages. There are multiple algorithms and formulas describing it. Your task is to implement PageRank in CUDA.

- Your implementation should handle directed page graphs with **spider traps** and **dead ends**. We suggest implementing the **Random Teleport** solution explained in here and here.
- Your final grade depends on the performance of your implementation.
- You are free to choose any dataset but your code should work for different dataset sizes.
- Your code should run faster than CPU-based implementations (you should demonstrate the speedup).
- You have to validate your calculated page ranks.
- Attach a report which reports all your work and results.
- You can't use libraries, such as **cuGraph** or **cub**.
- Extra points for implementing a crawler and generating your own dataset.
- Extra points for any improvement in page rank implementation, such as Matrix-Sparse Vector Multiplication.
- Extra points for any improvement in CUDA implementations (e.g. techniques are mentioned in **CUDA programming** slide like removing divergent branch, removing bank conflicts, and unrolling the warp). You should mention all your work in the document.

1.c Graphics

Using **OpenGL** (i.e., you can find more about in [here](#)), implement a program to obtain the result below.



- Your score depends on the similarity between your output and the given picture.
- You should be able to change camera's position by mouse drag.
- Attach a report which reports all your work and results.
- Extra points for implementation of zoom, walk around, light reflection and **Raytracing**.

Required Document

Please upload a zip file in courseware website.

General Rules

Please consider the course rules.

- Deadlines are tight.
- No extra time for this assignment.

Deadline

Monday 23:59. 1401/05/03.

Contact Information

Ask your questions via the courseware website or send an email to:

`hfalahati@ipm.ir`

Good Luck