Lab 4: Sobel Edge Detector – Hybrid Programming Model

EEL 6763 – Fall 2023

Part 1: Sobel Edge Detector in c

a) Given the black-and-white image in the file image.jpg (shown in Figure 1(a)), convert it into a corresponding input file using Matlab (see notes at the end of this lab) for your Sobel edge-detector application with the following specification: 5000X5000 array, 8-bit precision.

b) Implement a c program yourself or find an implementation of the Sobel Edge Detector algorithm in c.

c) Run the c code, debug if necessary, and obtain a baseline output of the edge detection.

d) Using Matlab (see notes at end of lab), convert the output file into an image (e.g., Figure 1(b)).

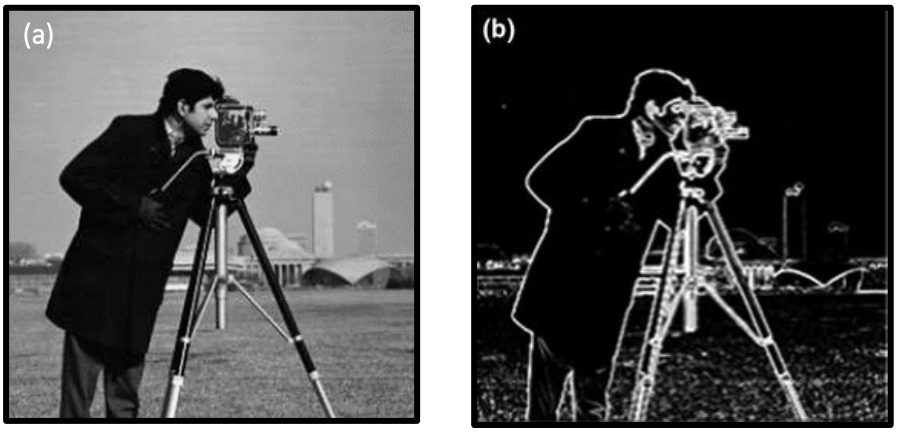

Figure 1

Part 2: Hybrid Programming Model: MPI+OpenMP

a) Modify the c code from Part 1 to extend it to use both MPI and OpenMP to obtain an output file of the edge detection.

b) Using Matlab, convert the output file into an image and compare with the baseline image in Part 1(d).

c) Perform design-space exploration (DSE) to identify the best-performing configuration. Timing should start right before the distribution of data to the ranks and ends right after the image is produced.

Use 4 ranks with 4 threads each, varying the number of nodes:

- 1 node, 4 ranks
- 2 nodes, 2 ranks each (also varying sockets assignments)
- 4 nodes, 1 rank each
- Any other MPI configurations that you think may help performance

You can find samples and explanation of SLURM scripts here for varying MPI configurations: https://help.rc.ufl.edu/doc/Sample_SLURM_Scripts#Threaded_or_multi-processor_job

Also vary OpenMP parameters to see if any improvements can be obtained (e.g., dynamic/static scheduling, chunk size, etc.)

SUBMISSION INSTRUCTIONS

Make sure you and your partner's names are at the top of every file, including the report.pdf; and only submit one set of files.

You are to submit 2 files on Canvas:

(1) Create a directory named **Lab4**. Use the following structure and zip the entire directory and submit the zip file on Canvas using the following name: lastNamesOfAllPartnersLab4.zip

Lab4/Part1

.c file
batch script
Output file from the c program

Lab4/Part2

.c file
batch script(s)
All output files used to produce the tables and figures in your report

- Name each output file appropriately.

(2) A pdf file (Lab 4 report) using the following name: lastNamesOfAllPartnersLab4.pdf

- Put your names at the top of the report and any information or explanation that you want to give me
- Part 1:
  - The c code that you implemented or found
  - An image produced from the output of the c program (i.e., using Matlab to convert the text file into an image.
- Part 2:
  - The modified MPI+OpenMP code
  - An image produced from the output of the MPI+OpenMP program (i.e., using Matlab to convert the text file into an image.
  - Explain clearly what you did to find the best performing configuration in Part 2(c).
  - Justify your selection of the best performing configuration with supporting data: graphs, tables, etc.; and a paragraph discussing the data.

Using Matlab for image/file conversion

Matlab to convert image to input text file:

```
img = imread("image.jpg");
gray = rgb2gray(img);
resized_gray = imresize(gray, [5000, 5000]);
writematrix(resized_gray, 'input.txt', 'Delimiter', 'tab');
```

Matlab to convert outfile text file to image:

```
i=dlmread("processed_matrix.txt");
i=uint8(i);
imshow(i)
```