

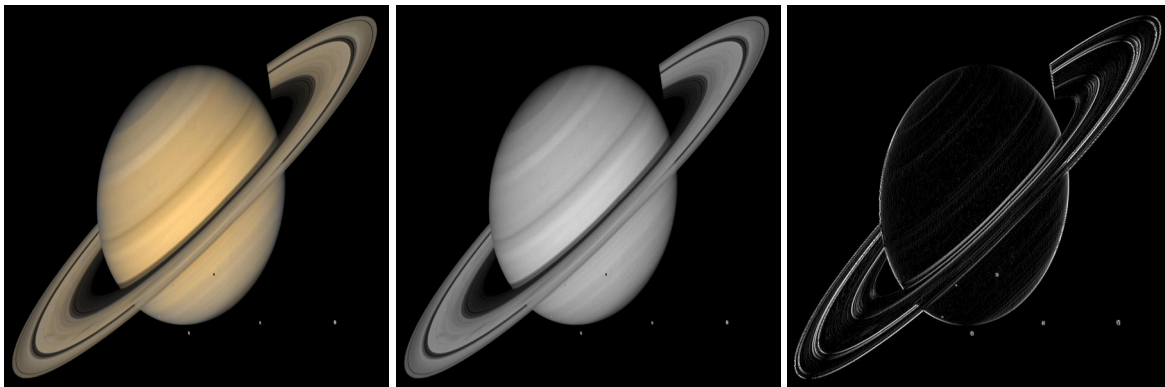
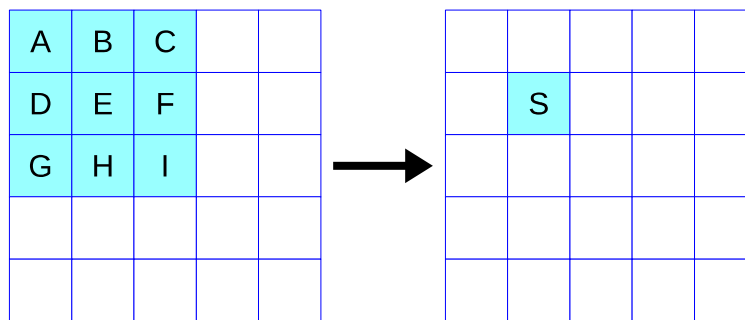
Homework #2 Assignment

Spring 2018 - ECE 2195 Parallel Computer Architecture

REMINDER: This is an INDIVIDUAL assignment. All deliverables for this assignment must be original (not borrowed or revamped from the work of others).

A. Sobel Filter

For this assignment, you will be implementing the Sobel filter for edge detection. Your MPI program will use the scatter-gather messaging pattern to parallelize this image processing application for gray-scale images. Sobel filter uses convolution and updates each pixel based on a weighted average across its neighboring pixels as shown below.



Provided

- PGM image reader and writer functions
- Template `sobel.c` and `Makefile`
- Sample 860x860 PGM image `input.pgm`
- *Hint:* You only need to worry about the distribution of data between nodes and the bounds when performing convolution

Rules

- Only rank 0 may use the PGM image reader and writer functions
- Use `MPI_Scatterv` and `MPI_Gatherv` for communication only
- Support arbitrary image dimensions and number of nodes
- Invocation: `mpirun -n $NODES ./sobel $IN $OUT $WIDTH $HEIGHT`
- This program should be able to build and run on the Center for Research Computing (CRC) cluster without any issues (e.g. compilation errors, segmentation faults, etc.)

B. Performance

Using the program from **Part A**, you will measure the speedup of the program by plotting the execution times when varying the number of nodes (\$NODES) and data size (\$WIDTH and \$HEIGHT) on the CRC cluster.

Provide the following in a report:

- Plot #1: Use values of 1, 2, 4, 8, 16 for \$NODES scaling the provide image by **x8**
- Plot #2: Use scales of 1x, 2x, 4x, 8x, and 16x for the provided image width and heights, and keep \$NODES fixed at **8 nodes**
- Describe the speedup observed when varying processing power and data size

Hint: Imagemagick can resize images: `convert -resize WxH! input.pgm input2.pgm`

Deliverables

- Completed `sobel.c`
- Plots and description in `report.pdf`