**Requirement**

Each student or team of 2 students will take one project. It is ok to take a theme that is not listed below (but check with the lab advisor before starting).

Each project will have 2 implementations: one with "regular" threads or tasks/futures, and one distributed (possibly, but not required, using MPI). A third implementation, using OpenCL or CUDA, can be made for a bonus.

**Algorithm: Hough transform**

The algorithm is composed of several stages.

1. **Preprocessing**: image is converted to grayscale and then Gaussian blur is applied which consists of convolving the grayscale image with the following filter:

2/159, 4/159, 5/159, 4/159, 2/159

4/159, 9/159, 12/159, 9/159, 4/159

5/159, 12/159, 15/159, 12/159, 5/159,

4/159, 9/159, 12/159, 9/159, 4/159

2/159, 4/159, 5/159, 4/159, 2/159

1. **Edge map**: to extract the edge map, the Sobel operator is used.

**Text

Description automatically generated with low confidence**

Then the gradients are computed in the following way:

**Text

Description automatically generated with medium confidence**

And if the gradients are greater than some preset threshold, then the pixel is an edge pixel.

1. **Mapping of edge points to the Hough space and storage in an accumulator**: For each edge point we convert it to polar coordinates and if it is within the image boundaries, the accumulator for that point is incremented.
2. **Interpretation of accumulator to generate infinite lines**: for each point in the accumulator, we check if it is larger then a preset threshold and if it is a peak in its neighborhood then the polar coordinates representing a line are converted back to cartesian coordinates.
3. **Drawing the lines**: for each point we check if it is on a detected line.

To perform all these steps, multiple transformers are implemented which receive an image and return a transformed image.

Transformers used:

- Convolution and GaussianBlur simply uses a convolution

- GrayScale

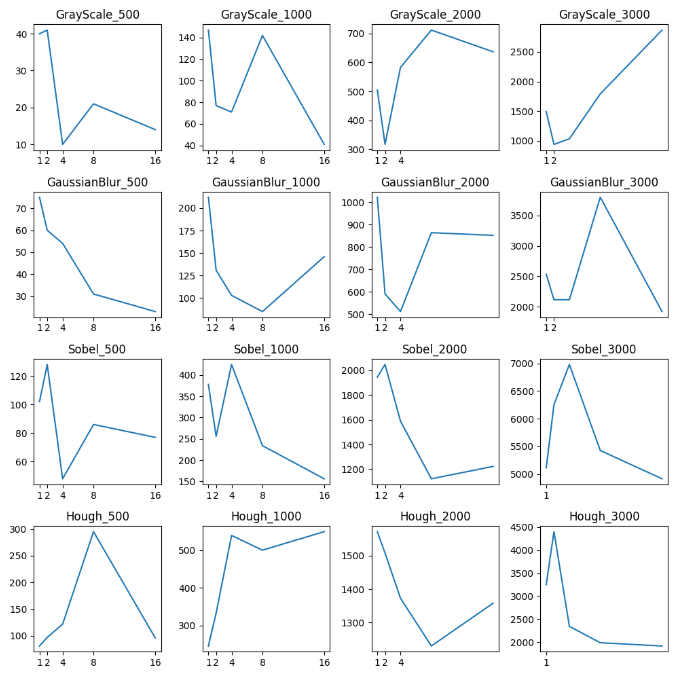
- Sobel – uses 2 convolutions (vertical and horizontal Sobel filters) then it computes their gradients – the output is an edge map

- Hough – this takes an edge map, and it outputs the edge map with the infinite lines

In the parallel versions, each thread receives a list of positions in the image, split sequentially and performs some action required by the transformer.

The MPI version operates in a similar manner, only that the input image must be sent to each process at the start.

**Performance**

****

