

# Concurrent Programming Assignment 4

Message Passing Interface (MPI) for Image Filtering Applications  
Distributed by Dr. Reza Shahidi on July 12, 2022

The Laplacian filter is used in image processing applications as an edge detector. It is a two-dimensional approximation of the second spatial derivative of an image. However, the Laplacian filter is known to amplify noise, and so often the image is first blurred with a Gaussian filter to reduce noise levels so that amplification with the Laplacian filter does not become a major issue.

Instead of first applying the Gaussian filter followed by application of the Laplacian filter, it is possible to combine these two steps into one and apply a single Laplacian-of-Gaussian filter.

The Laplacian-of-Gaussian filter is defined by the following expression:

$$\text{LoG}(x, y) = \frac{1}{\pi\sigma^4} \left[ 1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Depending on the extents of each of the two filters, different amounts of blurring and edge amplification will take place.

For the purposes of this assignment, we will assume the following discrete kernel with a value of  $\sigma = 1.4$ .

$$\begin{pmatrix} 0 & 0 & 3 & 2 & 2 & 2 & 3 & 0 & 0 \\ 0 & 2 & 3 & 5 & 5 & 5 & 3 & 2 & 0 \\ 3 & 3 & 5 & 3 & 0 & 3 & 5 & 3 & 3 \\ 2 & 5 & 3 & -12 & -23 & -12 & 3 & 5 & 2 \\ 2 & 5 & 0 & -23 & -40 & -23 & 0 & 5 & 2 \\ 2 & 5 & 3 & -12 & -23 & -12 & 3 & 5 & 2 \\ 3 & 3 & 5 & 3 & 0 & 3 & 5 & 3 & 3 \\ 0 & 2 & 3 & 5 & 5 & 5 & 3 & 2 & 0 \\ 0 & 0 & 3 & 2 & 2 & 2 & 3 & 0 & 0 \end{pmatrix}$$

Given the 256x256 pixel image in the assignment folder in PGM format (see <http://www0.cs.ucl.ac.uk/staff/Daniel.Alexander/D0b/pgmformat.txt> for more information on this format), you are to split the entire image into sixteen 64x64 non-overlapping square sub-images and apply the above filter by convolving it with the image pixel values. See <https://www.mathworks.com/help/images/what-is-image-filtering-in-the-spatial-domain.html> if you are not familiar with convolution in the context of image filtering. If any values are less than the minimum-allowed value in the image or greater than the maximum-allowed value in the image, according to the image's original bit depth, then the value should be clipped to the minimum or maximum respectively.

The actual image filtering/convolution should be implemented using the Message Passing Interface (MPI) with each node responsible for calculating the output for a single 64x64 subimage in the same PGM format as the input.

Note that near the boundaries of each subimage, pixel values from outside that subimage will be required to calculate the filter output. As a result, it will be necessary to send messages from the node responsible for one subimage to the nodes responsible for neighboring subimages in order to properly calculate the filter output. You may also assume that outside of the boundary of the entire image, all values are equal to zero (this will affect the quality of the filtering at the image boundaries, but simplifies the overall implementation).

The output of the entire MPI program should be a file containing the data in the same PGM format as the given input 256x256 image which is the result of filtering the image with the above kernel.