ENGI 4559 Digital Signal and Image Processing – Fall 2015
Project 1: Fun with Image Processing in Matlab (10%)

Spatial Filters

A spatial filter is composed of (1) a neighbourhood and (2) an operation. The neighbourhood is typically a square of pixels, and the operation is typically taking the median, mean, maximum, minimum, or weighted average of the pixels that neighbourhood. When a spatial filter is applied to an image, it creates a new output image where each pixel (x,y) is the result of applying the operation on the neighbourhood of pixel (x,y) in the original image. For example, if the neighbourhood is a 3x3 square with the target pixel in the middle and the operation is the median, then pixel (x,y) in the output image will be the median of the neighbourhood around (x,y) in the original image. The application of such a filter on a 4x4 image is illustrated here:

$$\begin{bmatrix} \mathbf{1} & \mathbf{5} & \mathbf{27} & \mathbf{32} \\ \mathbf{18} & \mathbf{48} & \mathbf{57} & \mathbf{6} \\ \mathbf{42} & \mathbf{19} & \mathbf{7} & \mathbf{31} \\ \mathbf{17} & \mathbf{45} & \mathbf{9} & \mathbf{22} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{1} & \mathbf{5} & \mathbf{27} & \mathbf{32} \\ \mathbf{18} & \mathbf{19} & \mathbf{27} & \mathbf{6} \\ \mathbf{42} & \mathbf{19} & \mathbf{22} & \mathbf{31} \\ \mathbf{17} & \mathbf{45} & \mathbf{9} & \mathbf{22} \end{bmatrix}$$

Notice that we cannot compute a 3x3 square around the edge pixels. For that reason, in this example, we simply exempted them of filtering and left them unchanged. You can use this simplification in this project as well.

Part 1:

Download the image “fields.png” from the course website. It is an 8-bit grayscale, 720x576 image of houses and a street overlooking some fields. Open the “fields” image in Matlab, and add 10% impulse noise to the image using the function seen in class. Then, filter the noisy image using the following filters:

- A 3x3 median filter
- A 3x3 mean filter
- A 3x3 minimum filter
- A 5x5 median filter

For each filtered image, compute the SNRI compared to the noisy image.

Discuss your results. Which filter offers the best improvement mathematically? Which filter offers the best improvement visually in your opinion? What differences do you see between the images, and how do they relate to the operation of the filter used to create each image?

Part 2:

Implement this weighted sum filter:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

This operation multiplies each pixel in the 3x3 neighbourhood with the weight in the corresponding entry of the filter, and the central pixel in the output image is set to the sum of the neighbourhood. For example, if the original image is this:

$$\begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix}$$

then in the central pixel in the output image will have the value "$20 + 40 + 60 + 80 - 4 * 50 = 0$".

After you've applied this filter to the entire image, subtract the filtered image you obtained from the original image, and look at the result. What is the effect of this filter on the image? Why? Can you think of a way to further accentuate this effect? How?

Submission:

Print out a report in which you present your results and your discussion on the questions asked in this project, and hand it to the professor in class on the due date. Do not hand in your Matlab code. Each student must hand in an individual report. Do not forget to write your name on your report.