Lab Assignment 7

Title: Image Filter

Learning Objective:

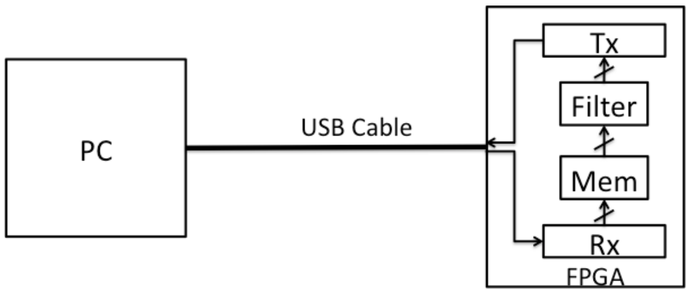
Learn how to design a circuit to perform simple computation over data stored in memory. Utilize the serial receiver and transmitter of the previous assignment to bring the data into memory and save the computed result.

Specification:

Design an image filter that uses a 3x3 sliding window. The serial receiver downloads the image to be filtered from a disk file into memory. The serial transmitter uploads the filtered image into another disk file. Filter coefficients are kept in another small memory. Make provision for at least two sets of filter coefficients.

Details:

This assignment builds over the circuit of Assignment 6. The image filter is introduced between the memory and the serial transmitter as shown below.



Filtering Operation

Let X be the m x n matrix denoting the image to be filtered.
Let Y be the m-2 x n-2 matrix denoting the filtered image.
Let C be the 3 x 3 filter coefficient matrix.

The filtering operation is defined as follows.

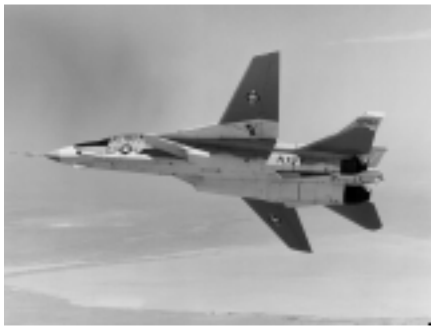
$$Y[I,J] = \sum_{i=-1}^{1} \sum_{j=-1}^{1} X[I+i, J+j] * C[i,j]$$
$$1 \le I \le m\text{-}2,\ 1 \le J \le n\text{-}2$$

The elements of X are read from the memory one by one, multiplied by coefficients and summed. This memory is filled with the data received by the receiver. The sum forms one element of Y and is passed on to the transmitter. Another smaller memory is used to store multiple instances of C matrix.

Filtering examples

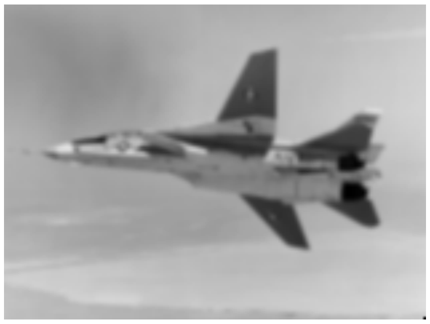
Two examples of coefficient matrices are shown below and effect of filtering with these is illustrated. Sum of the coefficients is usually kept as 1.

Original 120 x 160 image
Pixel values: 0 to 255



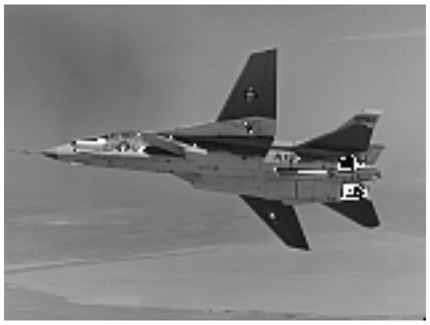
Smoothening filter :

$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$



Sharpening filter :

$$\begin{bmatrix} -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \\ -\frac{1}{9} & 1\frac{8}{9} & -\frac{1}{9} \\ -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \end{bmatrix}$$



Handling fractions

The coefficients can be scaled up by a suitable power of 2 and rounded off to integers. The final results could be scaled down by the same factor, simply by doing a right shift. Suppose the pixel values in X are 8 bit unsigned numbers in the range 0 to 255. These can be treated as 9 bit signed integers by prefixing a 0. Keeping the above two filter examples in mind, let the scaling factor be $2^7 = 128$. Then coefficient values can also be treated as

9 bit signed integers. Use 18 bits for holding the products and carrying out summation. In the final result, replace negative values with 0 and perform down scaling, that is, right shift by 7 bits (in other words, discard the rightmost 7 bits). In the above examples, the result will not exceed 9 bits (in other words, the leftmost 2 bits can be discarded). Note that scaling up of the coefficients is done off-line (these are constants that can be pre-computed) whereas scaling down of the result is done in the circuit.

Representing image

There are numerous image formats available. One simple format is pgm (portable gray map), suitable for gray scale images. See https://en.wikipedia.org/wiki/Netpbm_format for details. The file used in the above illustration is uploaded on moodle. You can download pgm file viewer from https://inkscape.org .