CPSC-6040 Computer Graphics Images
Homework 4 Convolution

Submitted by: Shashi Kumar Honnahalli Shivaraju
[XID: C88650674]

Objective: To filter images using convolution and display the results.

Implementation Notes:

a) Kernel Scale Factor:
The scale factor was selected as the maximum magnitude of either the sum of the positive weights or the sum of the negative weights in the kernel.

b) Image Normalization:
After the convolution of the input image with the filterkernel,normalization of the pixel values was performed so that the pixel values are with in the range of unsigned char [0,255].This was achieved using the below equation for each pixel of the convoluted image:

Normalized_Pixel[R][C] = (255 x Pixel[R][C] – minPixel) / (maxPixel – minPixel).

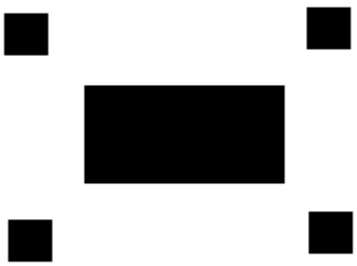
Here:
minPixel = minimum pixel value present in the convoluted image.
maxPixel = maximum pixel value present in the convoluted image.
Pixel[R][C] = current pixel value of the convoluted image at row R and coloum C in raster.
Normalized_Pixel [R][C] = Corresponding normalized pixel value for current pixel.

c)Boundary Mechanism:
I implemented a boundary mechanism in which during the process of convolution, if the index of the pixel is ouside the input image (at the image borders) then that index is skipped for processing.This prevented the formation of a black border around the convoluted image and instead replaces with a partially filtered output.But on repeated usage of the convolution filter,the borders of the convoluted image will differs from the center of the image.This difference at border also depends on the type of filter being used and its partial output.

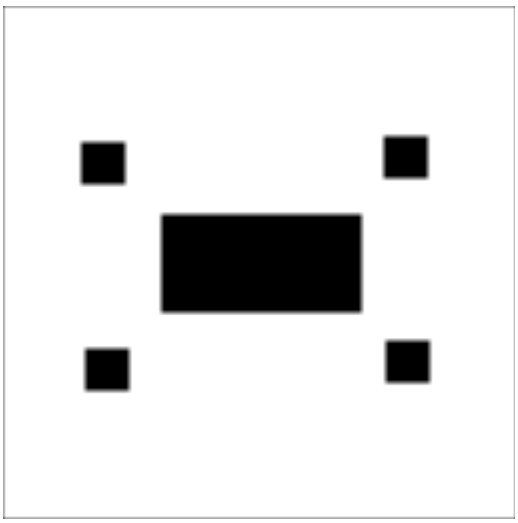
Results of Basic Requirements:

1) Pulse Filter

Input Image: Squares.png



Convoluted Image with Kernel pulse.filt:

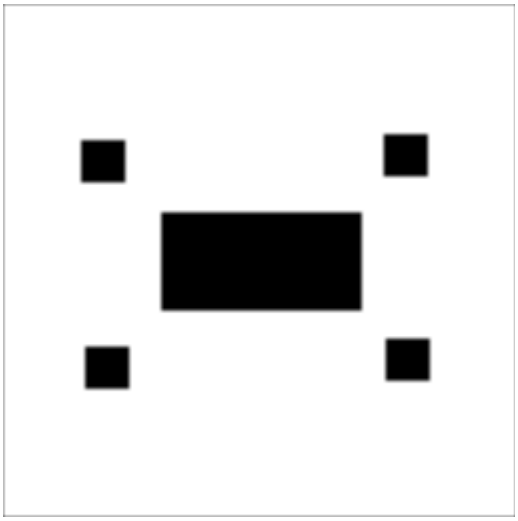


Observation : The Kernel pulse.filt blurs the input image.On applying the same filter repeatedly the blurness increases.Theoritically it is a low pass filter used for smoothing the images.

2) Tent Filter

Input Image: Squares.png

Convoluted Image with Kernel tent.filt:

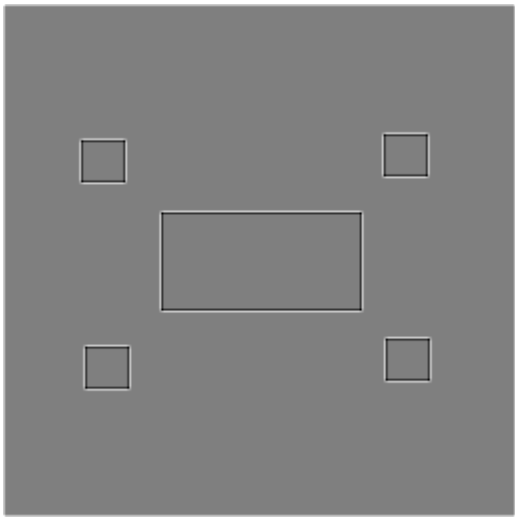


Observation : The Kernel tent.filt blurs the input image.On applying the same filter repeatedly the blurness increases.

3) High Pass Filter

Input Image: Squares.png

Convoluted Image with Kernel hp.filt:

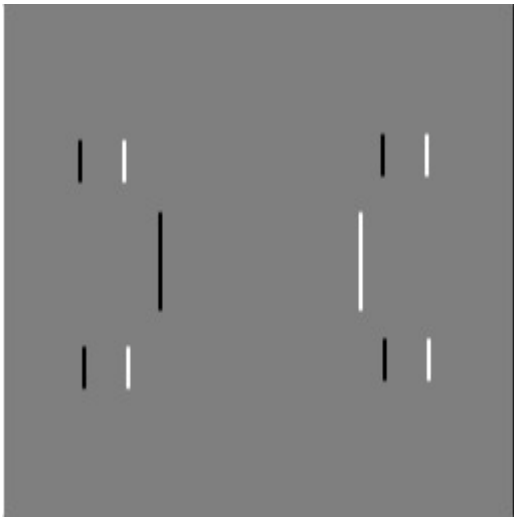


Observation : The Kernel hp.filt detects the edges of the input image.A high pass filter tries to retain the high frequency information within an image while reducing the low frequency information.It can be used for image sharpening and egde detection since it increases the contrast when there is variation in the pixel intensity.

4) Horizontal Sobel Filter

Input Image: Squares.png

Convoluted Image with Kernel sobel-horiz.filt:

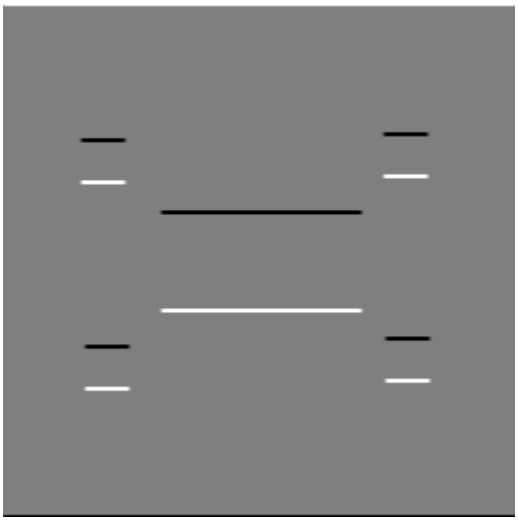


Observation : The Kernel sobel-horizon.filt is used to detect edges of the input image.
It will enhance the horizontal edges in an image.

4) Verticle Sobel Filter

Input Image: Squares.png

Convoluted Image with Kernel sobel-vert.filt:



Observation : The Kernel sobel-vert.filt is used to detect edges of the input image.
It will enhance the verticle edges in an image.

4) MyOwn1 Filter (Idea to Darken image)

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

The kernel utilizes the intensity of the surrounding pixels to decrease the intensity of the current pixel.

Input Image: waves.png

Convoluted Image with Kernel myown1.filt:

Oberservation: On repeated use of this kernel the color alternates between light and dark intensity and finally reaching greyscale with image distortion.

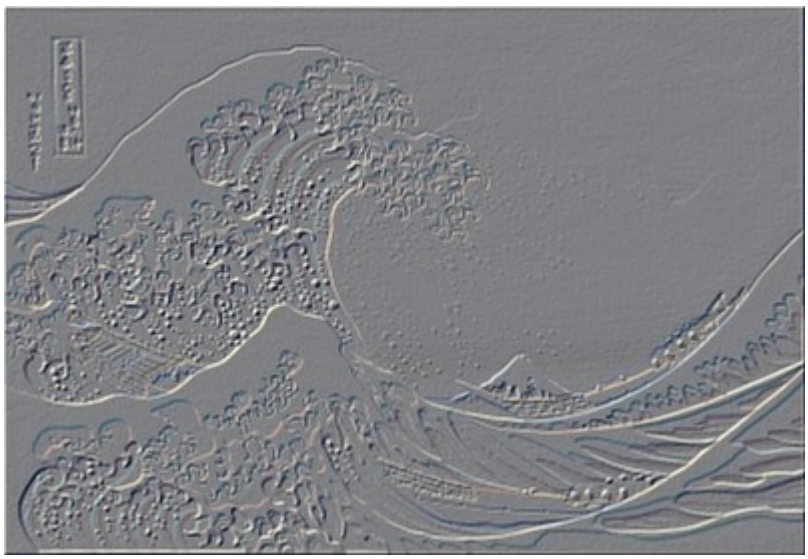
5) MyOwn2 Filter (Idea to convert to Greyscale)

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

The kernel utilizes the intensity of the surrounding pixels to convert the intensity of the current pixel to greyscale.

Input Image: waves.png

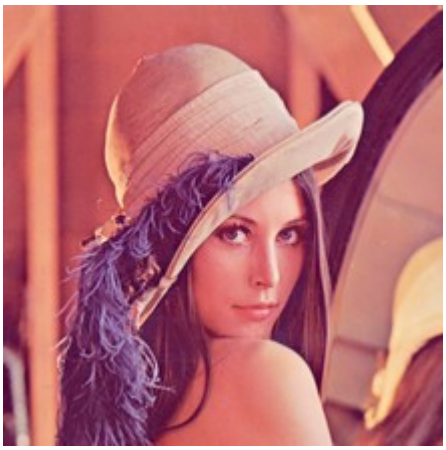
Convoluted Image with Kernel myown2.filt:



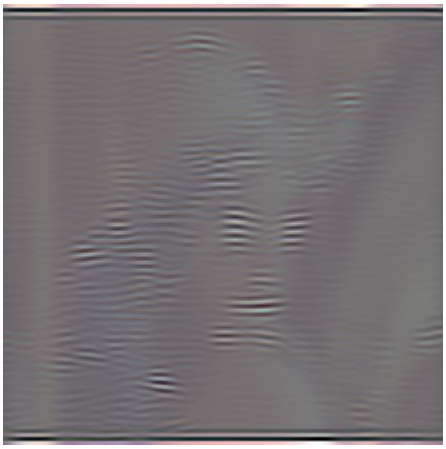
Oberservation: On repeated use of this kernel the color maintians to be greyscale but finally results in image distortion.

Results of Extension Requirements:

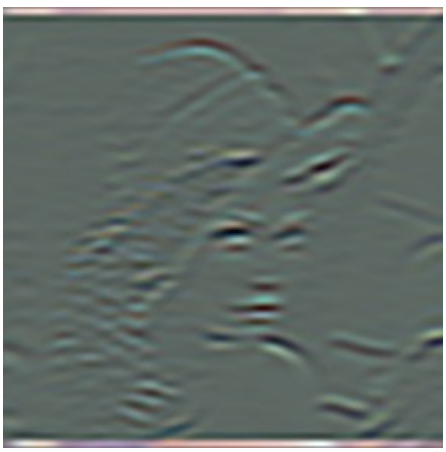
a)Input Image: Lenna.png



1) Gabor filter with theta = 0,sigma = 4 and period = 4

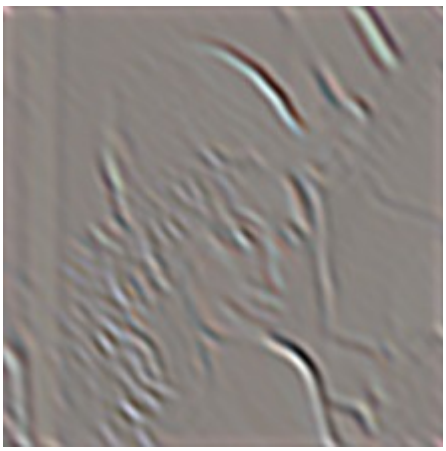


2) Gabor filter with theta = 0,sigma = 4 and period = 8

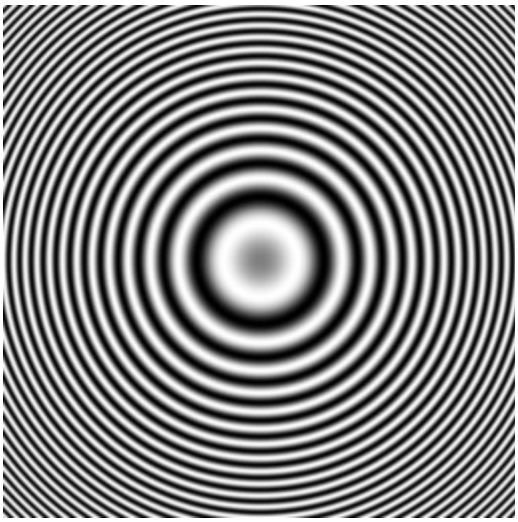


3) Gabor filter with theta = 45,sigma = 4 and period = 4

4) Gabor filter with theta = 45,sigma = 4 and period = 8

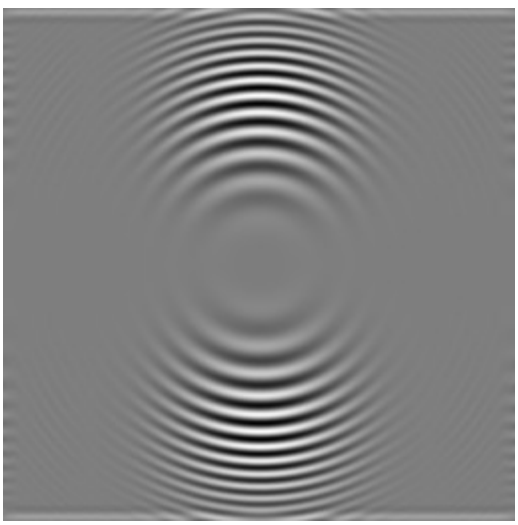


b)Input Image: sines.png



1) Gabor filter with theta = 0,sigma = 4 and period = 4

2) Gabor filter with theta = 0,sigma = 4 and period = 8



3) Gabor filter with theta = 45,sigma = 4 and period = 4

4) Gabor filter with theta = 45,sigma = 4 and period = 8