# Lecture 3

## Exercise 1

close all, clear all

I = imread('Static/test.jpg');

R = I(:,:,1); %Red

G = I(:,:,2); %Green

B = I(:,:,3); %Blue

G = sqrt(double(G));

G = uint8(G);

B = double(B);

B = B.^2;

B = uint8(B);

newI = cat(3, R, G, B);

%newI = I;

%newI(:,:,1) = R;

%newI(:,:,2) = uint8(G);

%newI(:,:,3) = uint8(B);

figure(1)

subplot(1,2,1)

imshow(I);

subplot(1,2,2)

imshow(newI)

figure(2)

subplot(3,1,1)

imhist(R), title('Histography, red channel')

subplot(3,1,2)

imhist(G), title('Histography, green channel')

subplot(3,1,3)

imhist(B), title('Histography, blue channel')

## Exercise 2

The convolution of two matrices/vectors u and v represents the area of overlap under the points as v slides across u.

Cross correlation of two matrices/vector u and v checks for any correlation between u and v. That is, can u be expressed by v?

Note when the vectors u and v are symmetric, the convolution and cross correlation of these vectors will be the same.

S2 is a periodic sample which repeats its pattern after 100 values, while S1 a logarithmic sample, increasing for every sample.

As seen in the convolution plot, the convolution climbs almost linearly, until it reaches every 100th point, where it takes a small dip (due to the periodic characteristics of S2). After the third period of S2, the convolution descends with a rather identical shape to the ascent, with small increases after every 100th point.

In the cross-correlation it seems that ascent and descent are equal in shape, only inverted, hence why the center of the cross correlation is not at x = 300.

## Exercise 3

Below is different filter options tried on a .jpg image with convoluting the original image and a

matrix. Essentially were three kinds of filters added: A low pass filter (h\_lp), a high pass (h\_hori, h\_veri, h\_both) and Gaussian filter (h\_gauss).

%Exercise 2

close all, clear all

image = imread('Static/test.jpg');

h\_lp = [1/9,1/9,1/9;

        1/9,1/9,1/9;

        1/9,1/9,1/9];

h\_hori = [-1 2 1;

          -1 2 1;

          -1 2 1];

h\_vert = [-1 -1 -1;

           2 2 2;

           -1 -1 -1];

h\_both = [-1 -1 -1;

          -1 8 -1;

          -1 -1 -1];

h\_gauss = [ 0.0113    0.0838    0.0113;

            0.0838    0.6193    0.0838;

            0.0113    0.0838    0.0113];

%imageFilter = conv2(double(image), h\_lp, 'same');

imageHlp= imfilter(image,h\_lp, 'conv');

imageHhori = imfilter(image,h\_hori,'conv');

imageHvert = imfilter(image,h\_vert,'conv');

imageHboth = imfilter(image,h\_both, 'conv');

imageHgauss = imfilter(image,h\_gauss,'conv');

figure('Name','Different filters applied to lenna.jpg')

subplot(3,2,1)

imshow(image), title('Original')

subplot(3,2,2)

imshow(imageHlp), title('Hlp filter')

subplot(3,2,3)

imshow(imageHhori), title('Horisontal filter')

subplot(3,2,4)

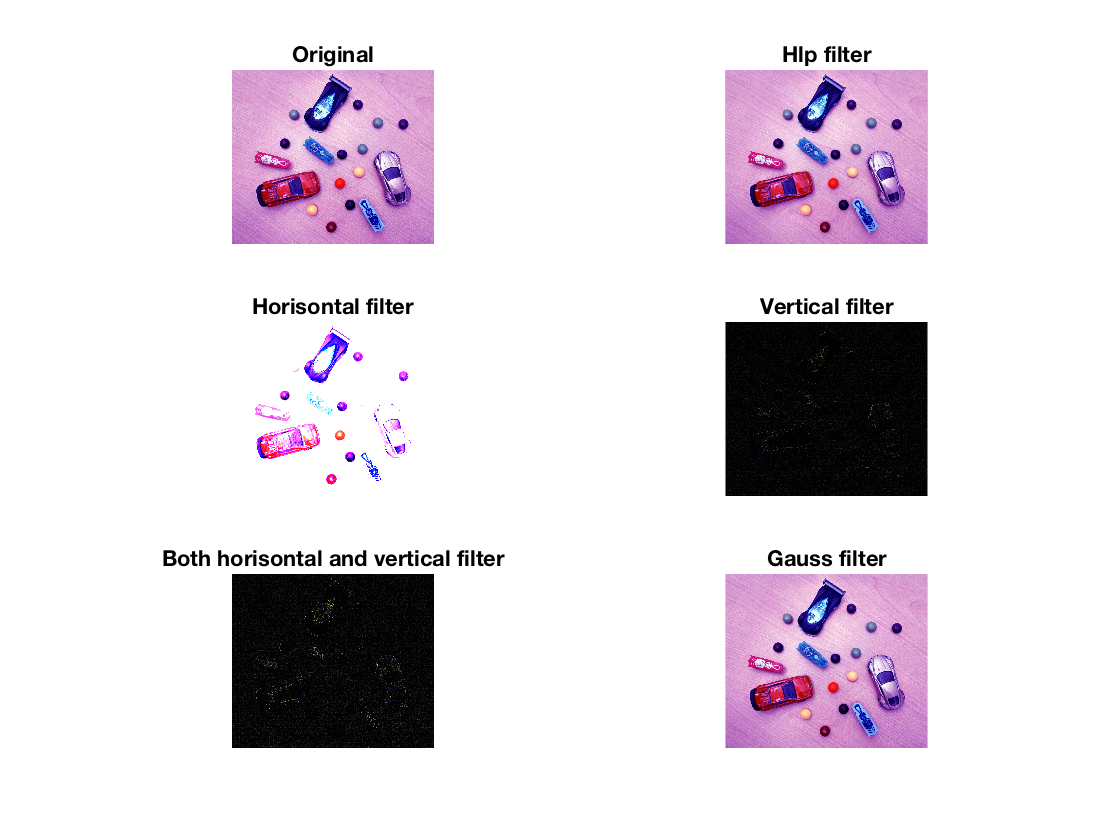
imshow(imageHvert), title('Vertical filter')

subplot(3,2,5)

imshow(imageHboth), title('Both horisontal and vertical filter')

subplot(3,2,6)

imshow(imageHgauss), title('Gauss filter')



## Exercise 4

Below are three different filter options attempted to filter out the noise gained from the ‘salt & pepper’ effect of imnoise(), with a noise density of 0.5. With the averaging filter, I tried with two different matrix sizes, 3x3 and 5x5. The second filter is the Gaussian with standard deviation of 2 (instead of the default at 0.5). The third filter option is median filtering of each of the RGB channels.

From observing, I noticed that the average filter with a 5x5 matrix filtered better than with the 3x3 matrix. Same pattern could be noticed with Gaussian by increasing the standard deviation.

The best result however was obtained with the median filtering.

clc, close all, clear all

I = imread('Static/lenna.png');

noisyI = imnoise(I,'salt & pepper', 0.05);

%Gaussian filtering

%hGaussian = fspecial('gaussian',3,0.5);

%FilterGaussian = imfilter(noisyI, hGaussian);

FilterGaussian = imgaussfilt(noisyI, 2);

%Average filtering

hAverage = fspecial('average',3);

hAverage5 = fspecial('average',5);

FilterAverage = imfilter(noisyI, hAverage);

FilterAverage5 = imfilter(noisyI, hAverage5);

%Median filter

FilterR = uint8(medfilt2(double(noisyI(:,:,1))));%Red channel: I(:,:,1)

FilterG = uint8(medfilt2(double(noisyI(:,:,2))));%Green channel: I(:,:,2)

FilterB = uint8(medfilt2(double(noisyI(:,:,3))));%Blue channel: I(:,:,3)

FilterMedian = cat(3,FilterR, FilterG, FilterB);

subplot(3,2,1)

imshow(I), title('Original')

subplot(3,2,2)

imshow(noisyI), title('Image with noise')

subplot(3,2,3)

imshow(FilterAverage), title('Image filtered with average 3x3')

subplot(3,2,4)

imshow(FilterAverage5), title('Image filtered with average 5x5')

subplot(3,2,5)

imshow(FilterGaussian), title('Image filtered with Gaussian \sigma = 2')

subplot(3,2,6)

imshow(FilterMedian), title('Image filtered with median')

