

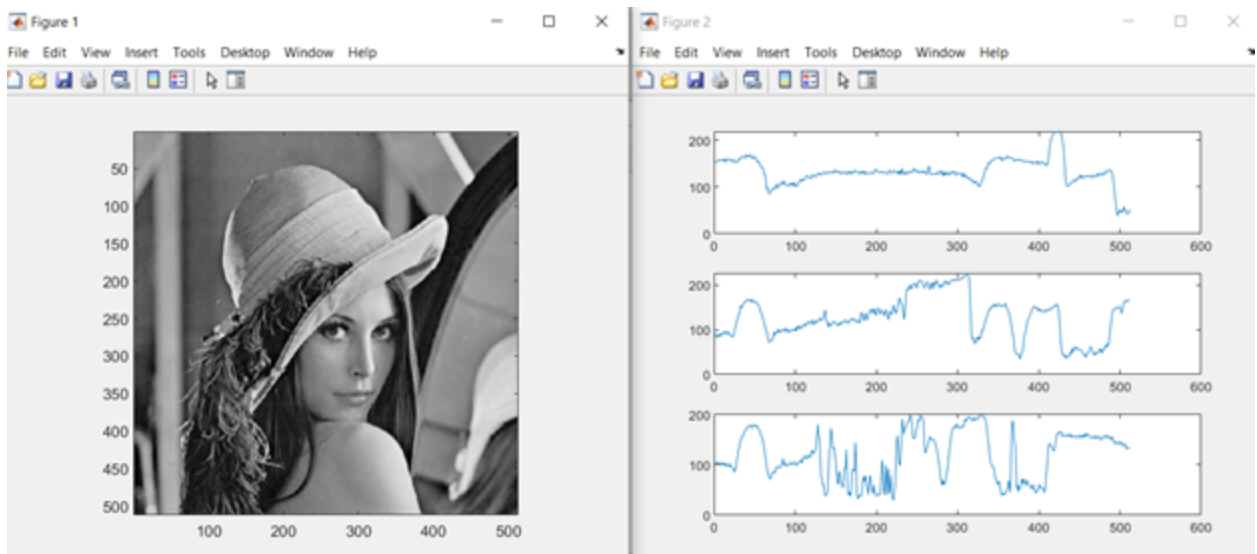
LAB 2

Filtering Image

1. Frequency content of an Image

Load and display "lena512.mat" and plot the value of the 20th row, the value of the 100th row, the value of the 250th row in the same graph (you can use subplot function). Compare the graph of these 3 rows and answer the following question:

Questions



- a. From the plot of signal from these 3 rows, which one has the highest frequency and which one has the lowest frequency?

ans **lena20 is the lowest and lena250 is the highest.**

- b. Compare the plots of these signals, and explain the characteristic of image having low and high frequency?

ans **แกน y คือระดับความเข้มของสี เมื่อสีมีความสว่างมากค่ายิ่งมาก บริเวณที่ภาพมีละลายเอียงเออะค่าสีมีการเปลี่ยนแปลงเยอะก็就会有ความถี่สูง บริเวณพื้นหลังมักจะมีความถี่ต่ำ**

- c. From a) and b), what is the meaning of frequency in image?

ans **ความถี่ในรูปนี้หมายถึงการเปลี่ยนแปลงของสีในแต่ละแถว หากในแถวนั้นมีการเปลี่ยนแปลงของสีในพิกเซลที่อยู่ติดกันมากก็就会有ความถี่มาก หากสีในบริเวณเดียวกันมีความใกล้เคียงกันไม่มีการเปลี่ยนแปลงเท่าไรก็就会有ความถี่ต่ำ**

2. Filtering Image (Blurring)

One dimension filtering

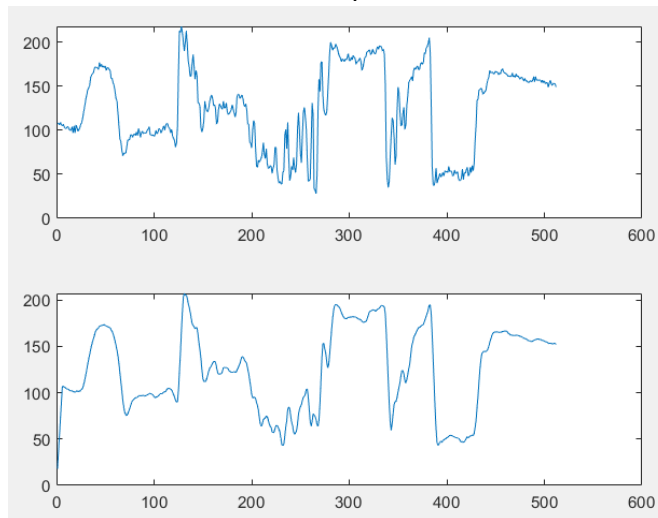
After we extract a sequence of value for one row from the entire image, we can filter this sequence with a “filter” function (same as lab2). The following are the MATLAB statement for filtering. The following statement is the example of filtering signal of row 200 for image lena512:

lab3_3.m

```
clear;
clc;
load('lena512.mat');
show_img(lena512,1,1);
lena20=lena512(200,:);
B=.....; % B parameter of filter function
A=.....; % A parameter of filter function
y_lena20=filter(B,A,lena20);
subplot(2,1,1)
plot(lena20);
subplot(2,1,2)
plot(y_lena20);
```

Questions

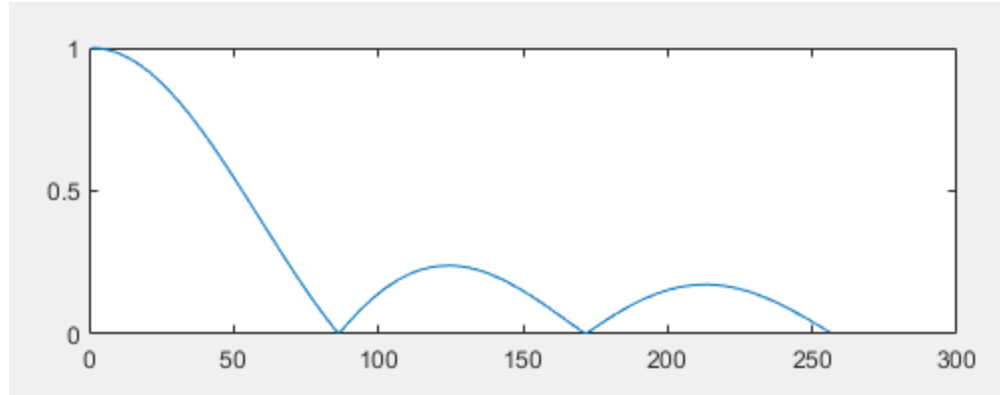
- Given the different equation: $y[n]=1/6(x[n]+x[n-1]+x[n-2]+x[n-3]+x[n-4]+x[n-5])$. What is the impulse response of this system?
ans $h[n]=\frac{1}{6}(d[n]+d[n-1]+d[n-2]+d[n-3]+d[n-4]+d[n-5])$
- Compare the plots of signal lena20 and y_lena20. Observe whether or not the filtered waveform is smoother than the input?



ans y_lena20 smooth ขึ้นจริง เนื่องจากการเฉลี่ยความเข้มสีของพิกเซลที่อยู่ใกล้เคียงกันจึงทำให้ความถี่ลดลง

62070501005, 62070501064, 62070501065, 62070501074

- c. Use “freqz” function to display the magnitude of frequency response of the above system. What kind of filter (lowpass, highpass, bandpass) for this system?
- ans lowpass filter คือการกรองให้คลื่นความถี่ต่ำผ่านไปได้เท่านั้น
highpass filter จะกรองให้คลื่นที่มีความถี่สูงสามารถผ่านไปได้
bandpass filter จะกรองคลื่นความถี่ในช่วงที่กำหนด



ซึ่งจากรูปจะเห็นได้ว่า มีเพียงค่าความถี่ต่ำเท่านั้นที่ผ่านไปได้ จึงสรุปได้ว่าระบบนี้เป็น lowpass filter

Two dimension filtering

In order to filter the entire image, you can use for loop to filter all the rows. This would create a new image made up of the filtered rows:

$$y[m,n]=1/6(x[m,n]+x[m,n-1]+x[m,n-2]+x[m,n-3]+x[m,n-4]+x[m,n-5])$$

However, this image $y[m,n]$ would only be filtered in the horizontal direction. Filtering the columns would require another for loop, and finally you would have the completely filtered image:

$$yy[m,n]=1/6(y[m,n]+y[m-1,n]+y[m-2,n]+y[m-3,n]+y[m-4,n]+y[m-5,n])$$

In this case, the image $yy[m,n]$ has been filtered in both directions by a 6point average.

Fortunately, Matlab has a built-in function `conv2` that will do this with a single call. It performs a more general filtering operation than rowcolumn filtering and it will be very helpful in this lab.

For example: To filter the image in the horizontal direction using a 6 point average, we set up a row vector of filter coefficients as:

lab3_4.m

```
clear;
clc;
load('lena512.mat'); % you can use whos to view the loaded variable
B=[1/6 1/6 1/6 1/6 1/6 1/6];
y_lena=conv2(lena512,B);
show_img(y_lena,1,1);
```

To filter the image in the vertical direction using a 6 point average, we set up a column vector of filter coefficients as:

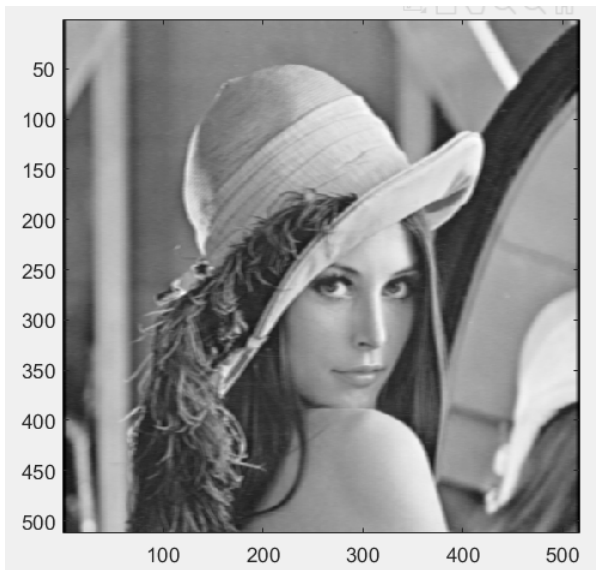
lab3_5.m

```
clear;
clc;
load('lena512.mat'); % you can use whos to view the loaded variable
B=[1/6 1/6 1/6 1/6 1/6 1/6]'; transpose to make B be a column vector
y_lena=conv2(lena512,B);
show_img(y_lena,1,1);
```

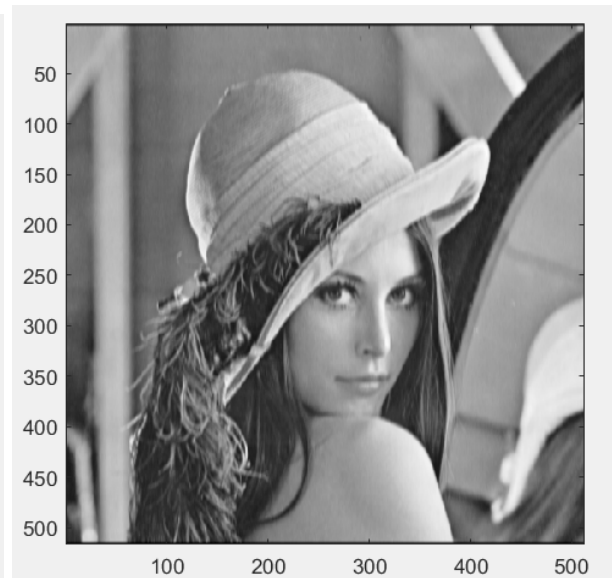
62070501005, 62070501064, 62070501065, 62070501074

Questions

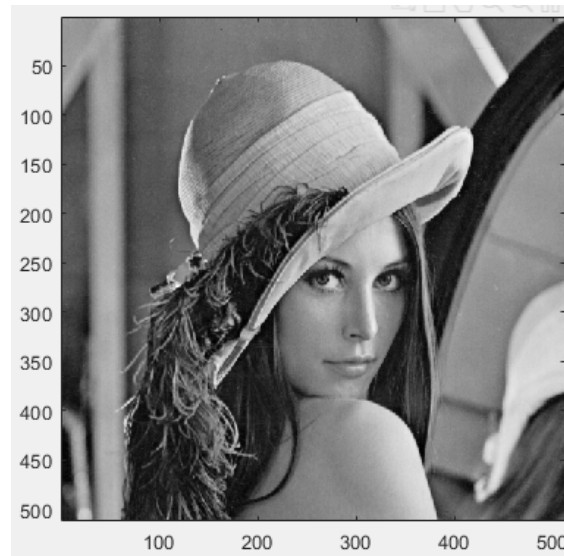
- a. Compare the output image from lab3_4.m, lab3_5.m, and the original image, what are differences?



lab3_4



lab3_5

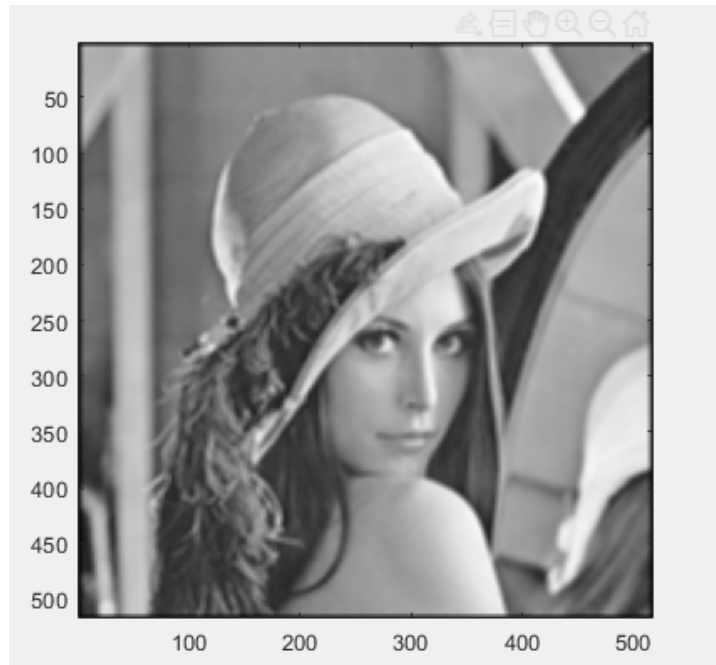


original

ans หากสังเกตจากขอบด้านข้างของภาพ lab3_4 และด้านบนของภาพ lab3_5 จะเห็นได้ว่ามีขนาดที่แตกต่างกันเมื่อเทียบกับภาพ original เนื่องจากภาพ lab3_4 ถูกเฉลี่ยค่าสีในแนวนอน และภาพ lab3_5 ถูกเฉลี่ยค่าสีในแนวตั้ง

62070501005, 62070501064, 62070501065, 62070501074

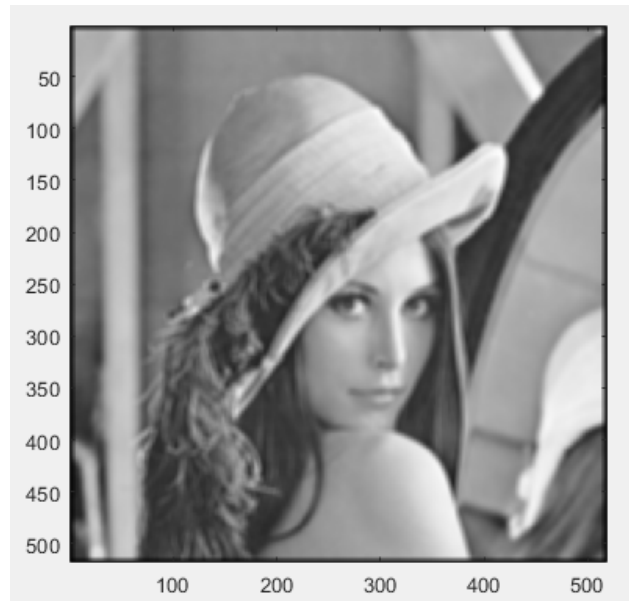
- b. Read help of “conv2” function, and implement the code that performs both horizontal and vertical filtering using the 6 point average method.



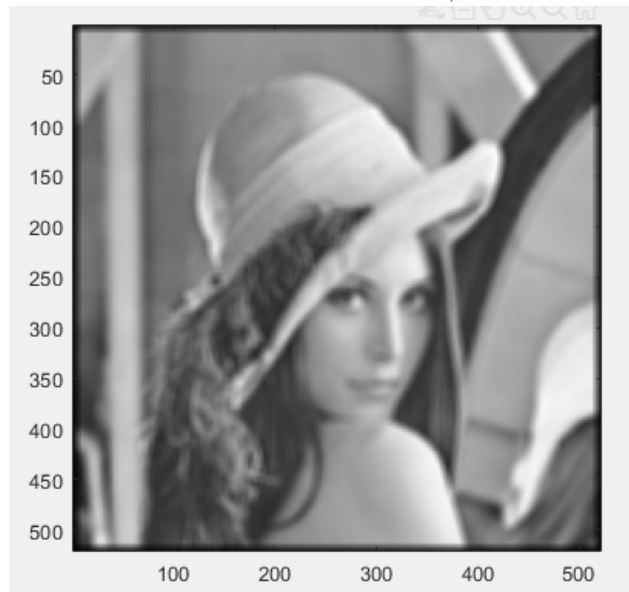
ภาพที่ได้เกิดจากการนำเมทริกซ์ % มา transpose แล้วนำมา convolute กับเมทริกซ์เดิม เพื่อให้เป็นเมทริกซ์ 2 มิติ จากนั้นนำมา filter กับรูปเดิม

```
x=[1/6 1/6 1/6 1/6 1/6 1/6];  
y=x';  
B=conv2(x,y);  
y_lena=conv2(lena512,B);  
show_img(y_lena,1,1);
```

- c. What happen to the output image if we increase the number of point average of the impulse response?



ภาพจากการเฉลี่ย 7 จุด



ภาพจากการเฉลี่ย 10 จุด

หลังจากเพิ่มจำนวนจุดที่นำมาเฉลี่ย พบว่ารูปภาพมีความเบลอมากขึ้น

- d. Why you can blur the image by using a point average filtering?
เนื่องจากเมื่อค่าเฉลี่ยของแต่ละจุดที่ใกล้เคียงกัน มีค่าที่ใกล้เคียงกันมากขึ้น ความแตกต่างก็น้อยลง แล้วรูปที่ได้จะมีค่าสีใกล้เคียงกัน ทำให้ภาพดูเบลอขึ้น

3. Edge of image

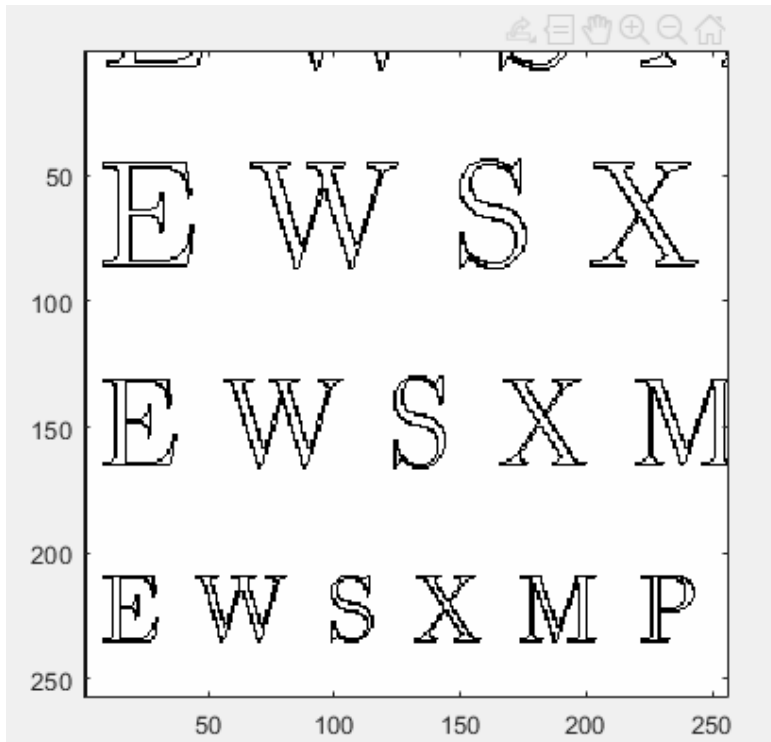
The concept of edge detection is the extraction of the difference between two concatenated pixels. One of the methods is evaluating the gradients generated along two orthogonal directions. An edge is judged present if the gradient of the image exceeds our defined threshold value, $t = T$. The gradient can be computed as the derivatives along both orthogonal axes.

Questions

- a) From the study of one-dimension image filtering and blurring image, how to extract the edge of the image?

ans นำพิกเซลที่อยู่ติดกันมาหาผลต่างทั้งแนวนอนและแนวตั้ง จากนั้นนำผลลัพธ์จากแนวนอนและแนวตั้งมารวมกันโดยใช้ฟังก์ชัน min เนื่องจากจุดที่เป็นสีดำคือ 0

- b) Use the image "echart.mat", to show the result of edge extraction from the method that you proposed in a)



```
clear;
clc;
load('echart.mat');

y=filter([1,-1],1,echart);
for j=1:256
    for i=1:257
        y_new(i,j)=255-abs(y(i,j));
    end
end

x=filter([1,-1],1,echart');
for j=1:257
    for i=1:256
        x_new(i,j)=255-abs(x(i,j));
    end
end
x_new = x_new';

echart_new = min(x_new,y_new);
show_img(echart_new,1,1);
```

นำพิกเซลที่อยู่ติดกันมาหาผลต่างทั้งแนวนอนและแนวตั้ง จากนั้นนำผลลัพธ์จากแนวนอนและแนวตั้งมารวมกันโดยใช้ฟังก์ชัน min เนื่องจากจุดที่เป็นสีดำคือ 0