MÜHENDİSLİK FAKÜLTESİ
FACULTY OF ENGINEERING
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ
DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING

ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

DUMLUPINAR BULVARI 06800
ÇANKAYA ANKARA/TURKEY
T: +90 312 210 23 02
F: +90 312 210 23 04
ee@metu.edu.tr
www.eee.metu.edu.tr

EXPERIMENT 7. IMAGE PROCESSING, 2D FFT, FILTERING, EDGE DETECTION

PART 1

LABORATORY REPORT

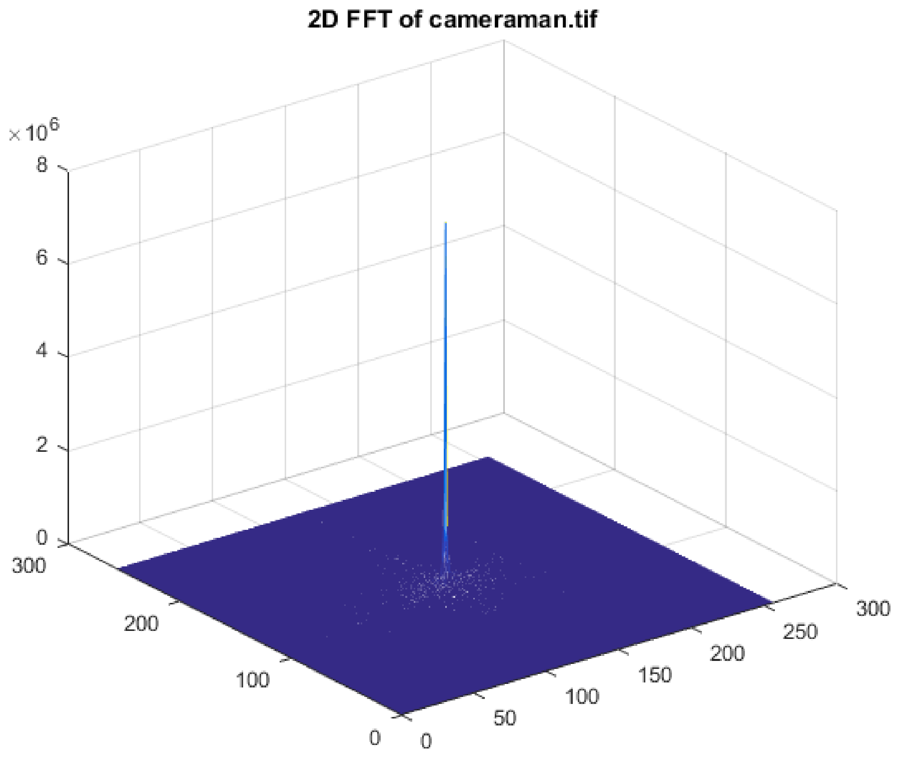
Student 1: Süleyman Emre CAN-2093524
Student 2: Yekta Demirci - 2093607

MATLAB Programming Tasks

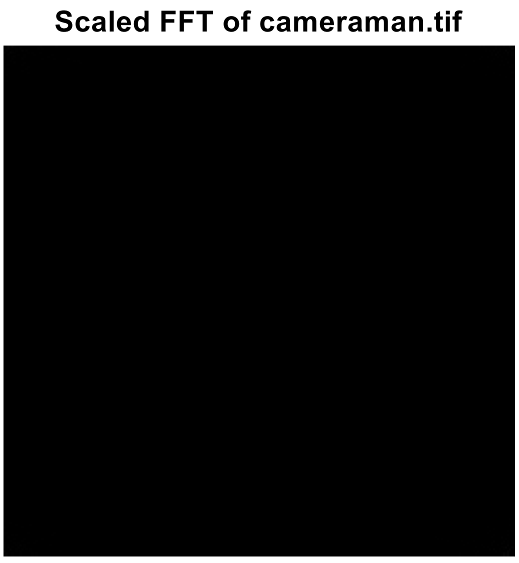
a) Write a ***"expriment7.m"*** file to load two images, ***'cameraman.tif'*** and ***'lena.png'***. Use ***'imshow'*** to display the images.

b) Take the 2D FFT of ***cameraman*** in order to visualize the low and high frequency components. Use ***'fftshift'*** to center the DC component.

- First, use **"mesh"** command to observe 2D FFT magnitude transform. Note that DC component is extremely high. In order to plot magnitude transform using **"imshow"** as a grayscale image, you need to scale the magnitude transform.

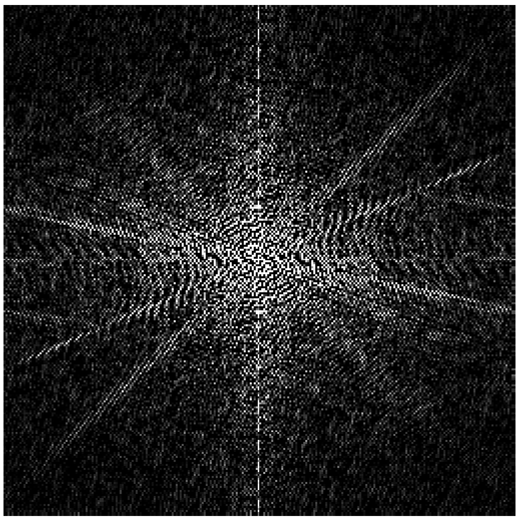


- Divide all the components of 2D FFT magnitude transform by the maximum element to normalize the magnitude transform. Then, plot the magnitude transform using **“imshow”. Attach the magnitude of 2D transform to your report.**



- Observe that scaling by the maximum element is not an effective approach in order to observe frequency distribution of the cameraman image. Now, scale the 2D FFT magnitude transform by 1/10000 and plot the magnitude transform using **"imshow". Attach the magnitude of 2D transform to your report.**

1/10000 Scaled Magnitude Plot of cameraman.tif



c) Show that 2D fft, ***'fft2'***, is equivalent to applying ***'fft'*** to rows and columns respectively.

- For this purpose, consider the following matrix,

```
X= [  1    2    3    4;
      5    6    7    8;
      9   10   11   12];
```

- Take 2D FFT of X and denote it by Y. **Write the elements of Y in matrix form.**

78	-6,00000000000000 + 6,00000000000000i	-6	-6
-24,0000000000000 + 13,8564064605510i	0	0	0
-24,0000000000000 + 13,8564064605510i	0	0	0

- Now, apply fft to the rows of X and denote the result by Z. **Write the elements of Z in matrix form.**

10	-2.0000 + 2.0000i	-2.0000 + 0.0000i	-2.0000 - 2.0000i

26	-2.0000 + 2.0000i	-2.0000 + 0.0000i	-2.0000 - 2.0000i
42	-2.0000 + 2.0000i	-2.0000 + 0.0000i	-2.0000 - 2.0000i

- Now, apply fft to the columns of Z and denote the result by T. **Write the elements of T in matrix form.**

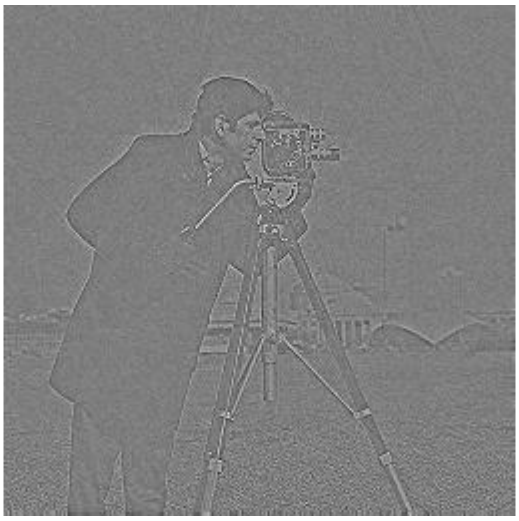
78	-6,00000000000000 + 6,00000000000000i	-6	-6
-24,0000000000000 + 13,8564064605510i	0	0	0
-24,0000000000000 + 13,8564064605510i	0	0	0

As it can be seen, matrix Y& T are equal.

d) Obtain the phase components of 2D fft of ***cameraman***. Assume that the magnitude components are **one** and use ***ifft2*** to reconstruct the image by using only the phase information as shown in Fig. 5. Selecting magnitudes **randomly** may result **complex image after ifft2**. In this case, show the **real part of the image** in order to preserve the phase components of the original cameraman image. **Attach the reconstructed image**. ***Comment on*** your findings.

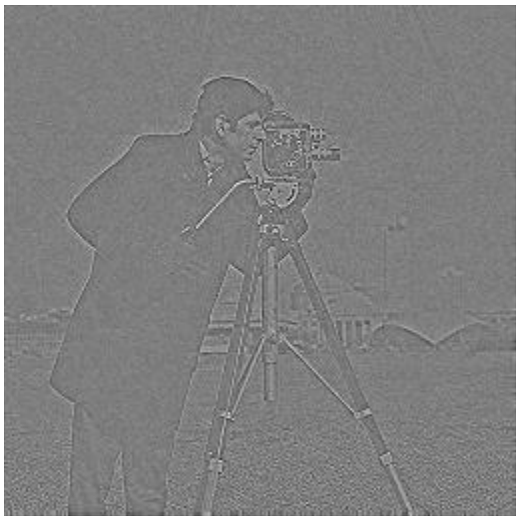
- In order to bring the resultant image into the range 0-1, you can use ***"mat2gray"*** command before **"imshow",** i.e., **imshow(mat2gray()).**

Reconstruction of the Image w/ Magnitude Components=1

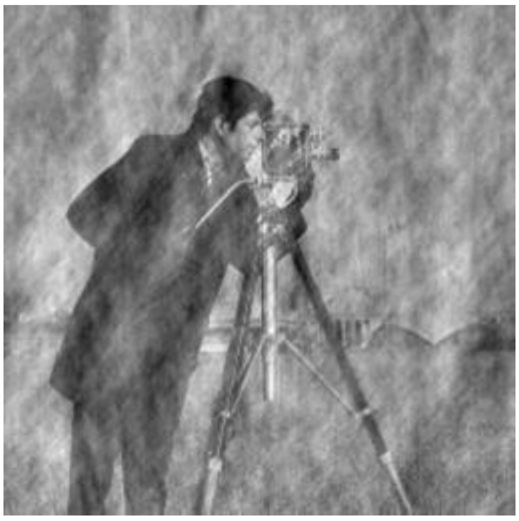


- Try alternatives for the magnitude terms during the reconstruction. One alternative is to use random numbers, i.e. ***rand()***. Another alternative is to use the 2D FFT magnitude of ***lena*** image. **Attach the reconstructed image for each case**. Which one gives ***better reconstruction***? **Comment on** the results.

Reconstruction of the Image w/ Mag Components are rand()



Reconstruction of cameraman.tif w/Mag Components of lena.png



Using magnitude components are as 1 or random didn't have much effect; however, the magnitude component of the lena picture significantly improved the reconstruction of image. This result is obtained due to the general behavior of the images. Since images are generally contain low frequency components, magnitude part of an different image helped more than random or constant number.

e) Consider the following ***9x9 nonlinear phase lowpass kernel***.

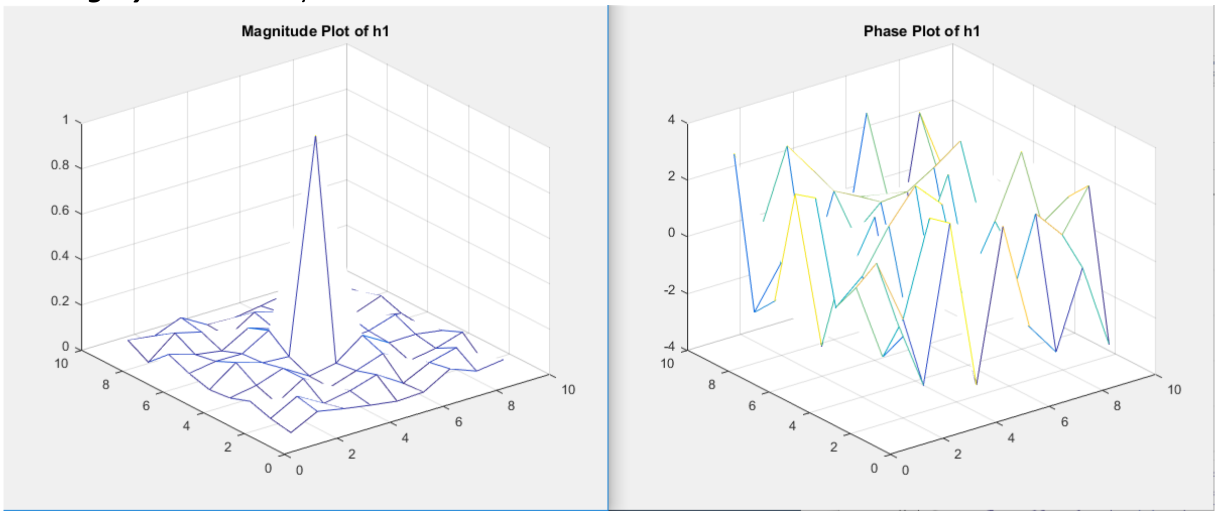
```
h1=1/4248*[92    35    75    96    22    39    70    30    81;
           82    40    38    34   100    49    36     3     3;
           16    66    60    71    57    88    80     8    21;
           82    37    15    93    81    90    98    17    91;
```

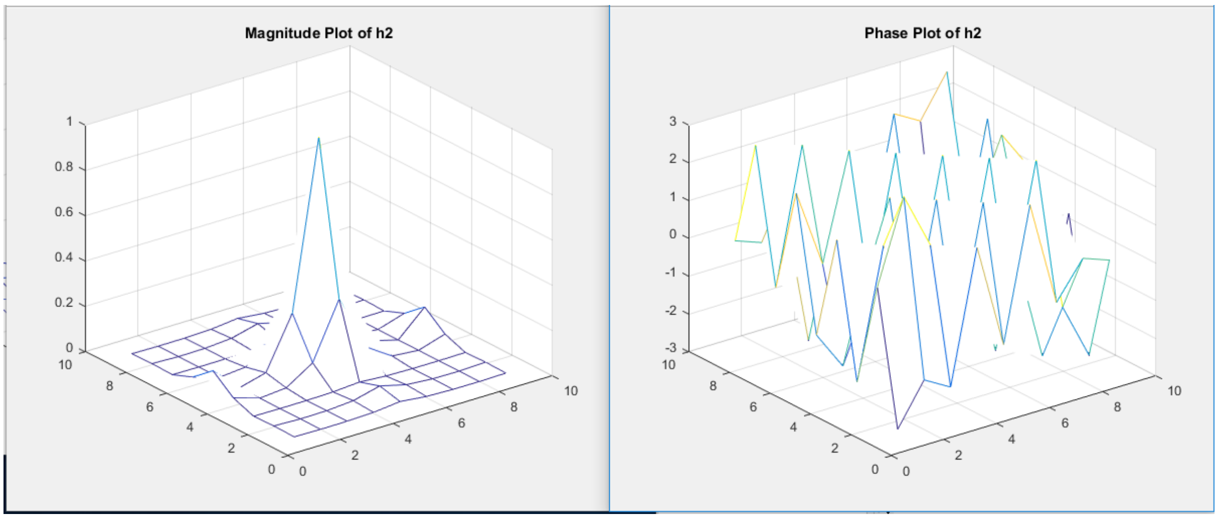
```
        82        9        4       41       52       92       68       34       82;
        49       37       56       54       84        6       62       98       50;
        82        9       28       88        9       36       73       83       40;
         2       58        2       40       48       38       54       79       19;
        80       77        6       54       58       50       35       60       84;];
```

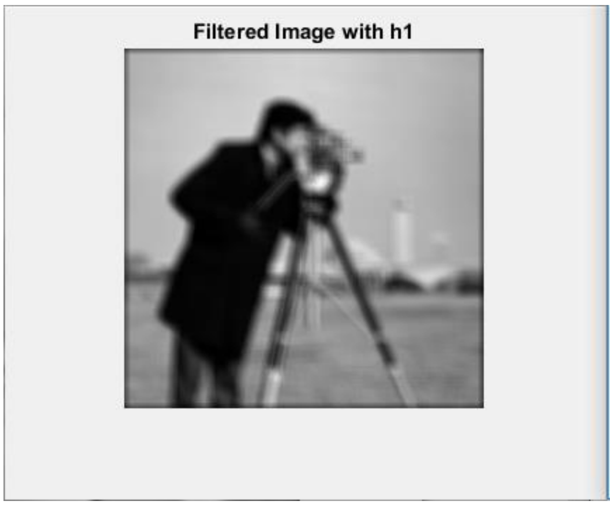
Design a ***9x9 separable lowpass linear phase filter***, h2=(ha'*hb)/sum(sum(ha'*hb)). Let

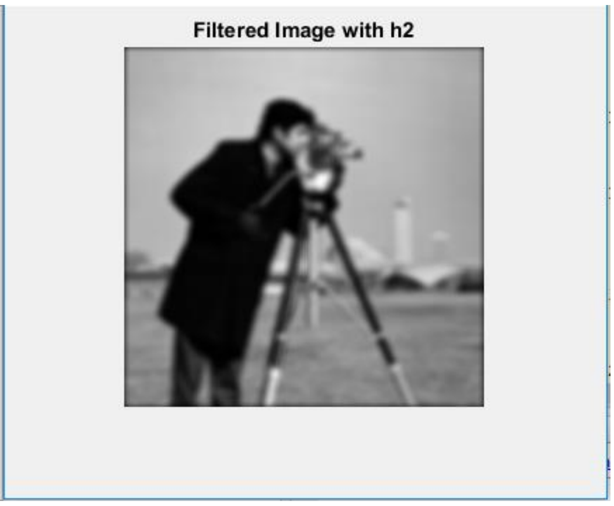
```
ha=[1 2 3 4 5 4 3 2 1];
hb=[1 1 2 2 3 2 2 1 1];
```

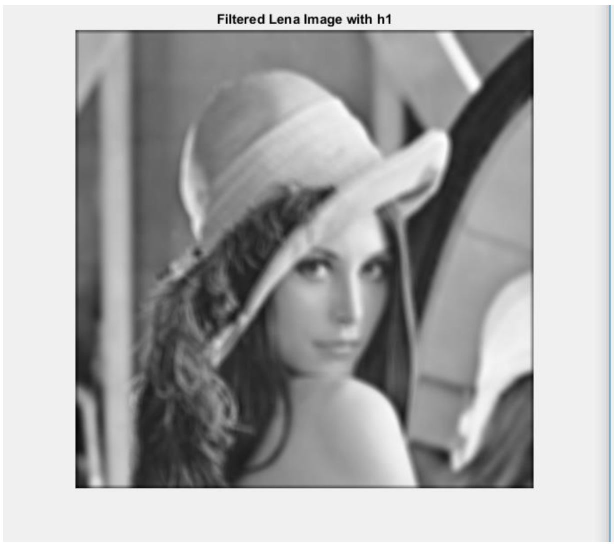
Plot the ***magnitude*** and ***phase*** characteristics of these filters as shown in Fig. 6 using ***"mesh"*** function. **Attach the plots to your report**. Filter the ***cameraman*** and ***lena*** images by these filters and ***comment on*** the phase distortions introduced by the nonlinear phase filter. Use **"double"** command before **filter2** command in order to convert uint8 data type of the loaded images to double.
Note that you may need appropriate scaling for better visualization (You can use ***"mat2gray"*** command).

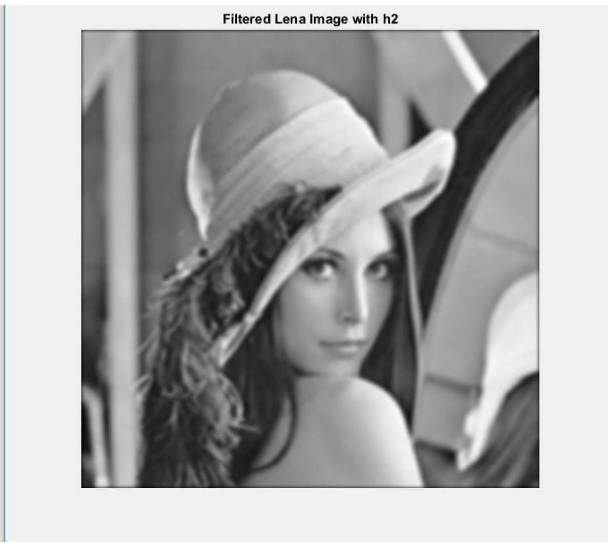












f) Consider the filter, ***h2***, in ***e)***. Filter the ***lena*** image with this filter and obtain the output ***y(n₁,n₂)*** as shown in Fig. 7 by adding Gaussian random noise with a known variance. Take the **standard deviation as 1**. Display the resulting image with ***imshow*** command. Design the inverse filter as it is given in equation (7). You can take the **1/SNR as constant at 1**. You should use an appropriate FFT2 size to obtain linear convolution. Filter the output, $y(n_1,n_2)$ with the inverse filter. Display the result of the inverse filtering and ***comment on*** the deconvolution performance. **Attach both noise added and reconstructed images.** ***Determine and write*** the ***least squares error*** in reconstruction. Use **mat2gray** while calculating LSE.

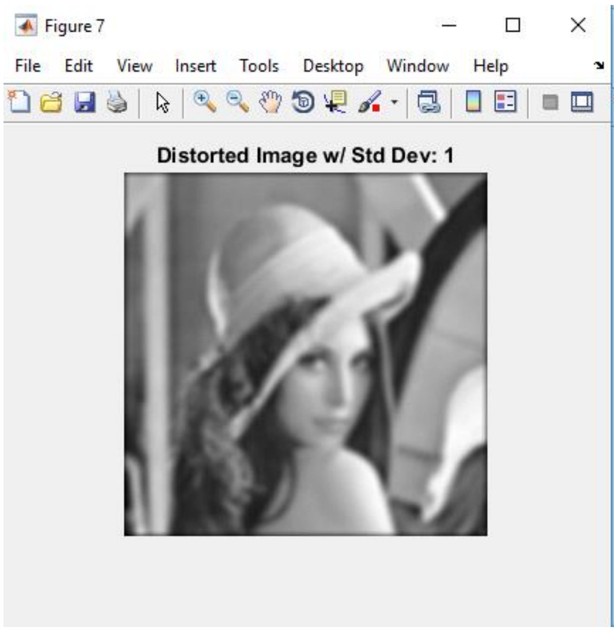


Figure 7
File Edit View Insert Tools Desktop Window Help
Distorted Image w/ Std Dev: 1

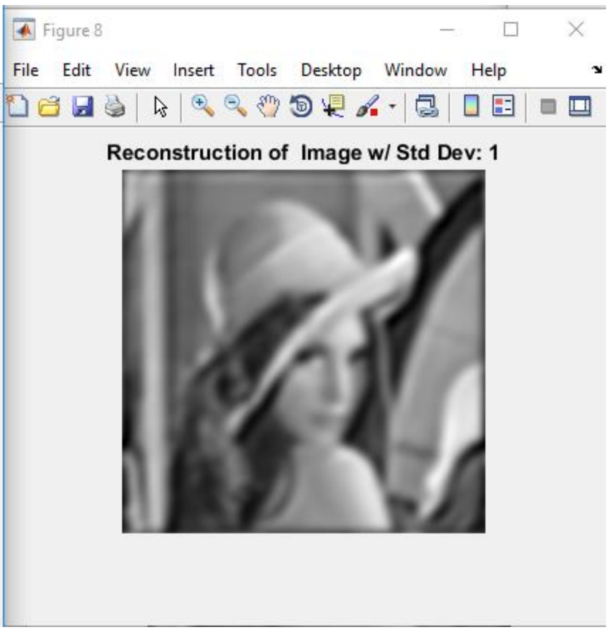
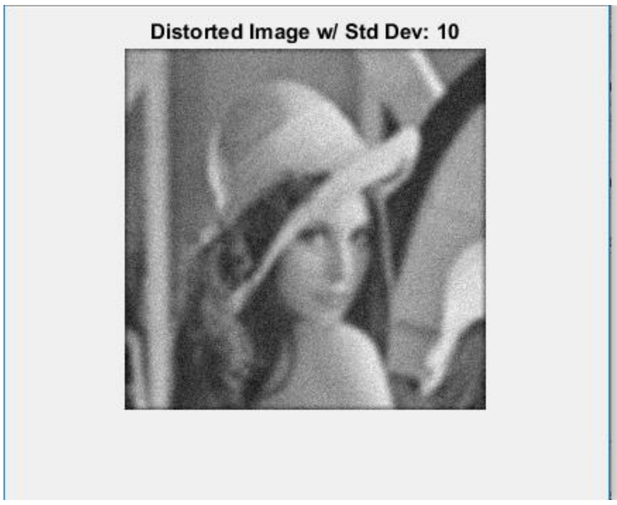


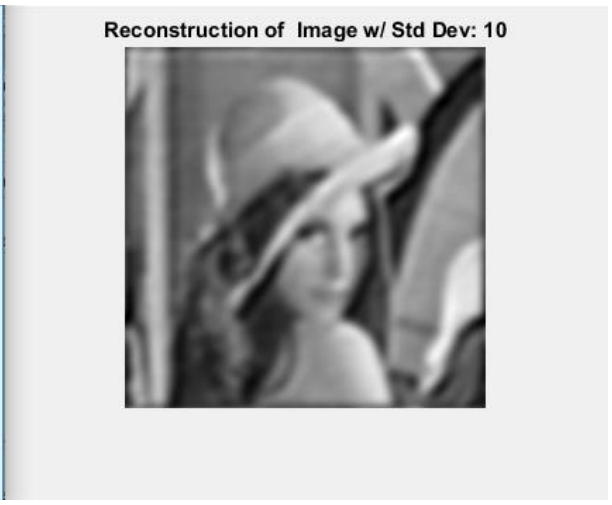
Figure 8
File Edit View Insert Tools Desktop Window Help
Reconstruction of Image w/ Std Dev: 1

LSE =

0.0095

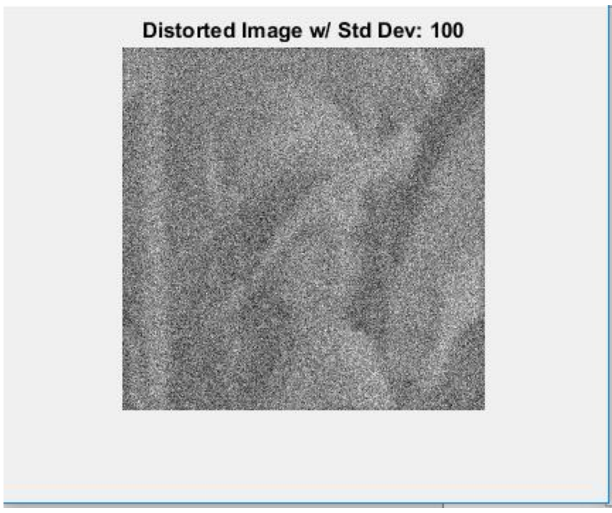
- Try different noise levels (take **standard deviation as 10 and 100**) and **comment** on the reconstruction performance. You can change 1/SNR constant for better reconstruction for different noise levels. ***Determine and write*** the ***least squares error*** in reconstruction for each noise level.

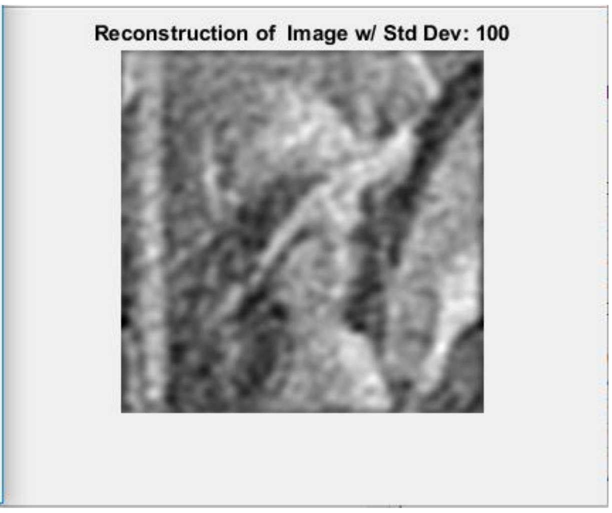




LSE =

0.0095 (Expecting slightly higher than previous value)





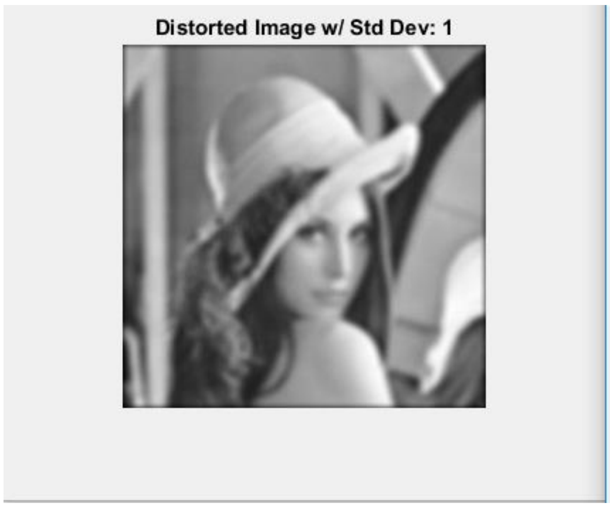
LSE =

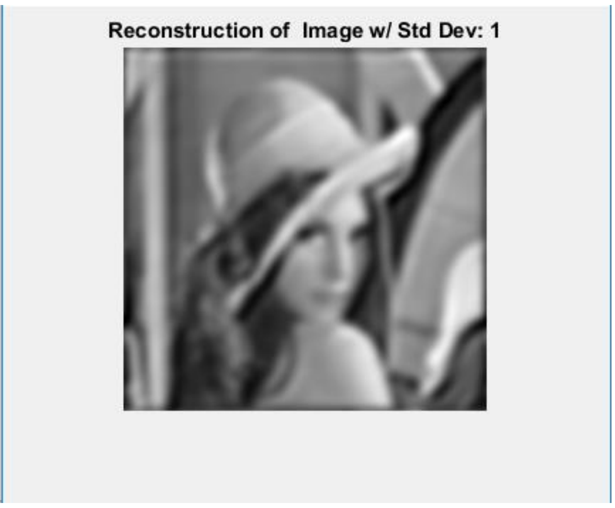
    0.0141

- **Change the h2 filter (by changing ha and hb)** as follows,

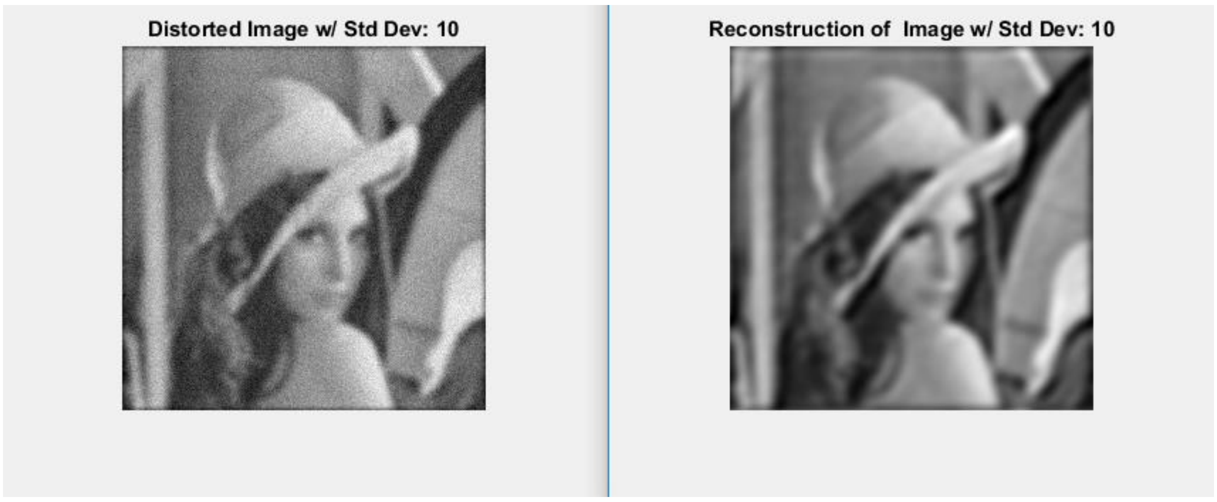
```
ha_new=[1.0000   7.9139   27.5038   54.7921   68.4044   54.7921   27.5038
7.9139   1.0000];

hb_new=ha_new;
h2_new=(ha_new'*hb_new)/sum(sum(ha_new'*hb_new)).
```

- **Repeat the previous parts** (take noise standard deviation as 1, 10 and 100) and ***comment on*** the inverse filtering quality. Don't **forget to attach the images and LSE values for each noise level.**

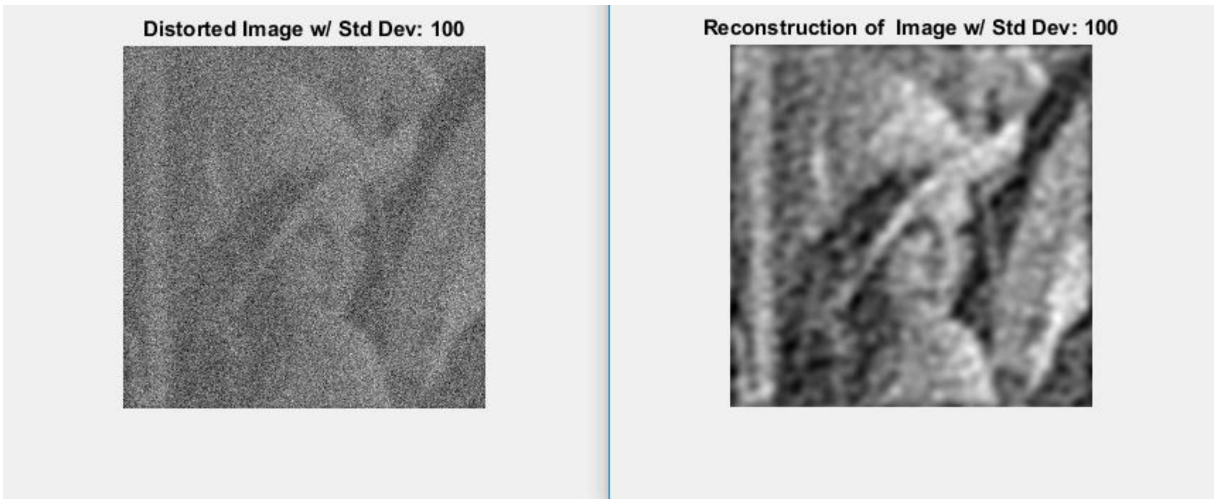




LSE = 0.0095


Distorted Image w/ Std Dev: 10
Reconstruction of Image w/ Std Dev: 10

```
LSE =

   0.0097
```


Distorted Image w/ Std Dev: 100
Reconstruction of Image w/ Std Dev: 100

```
LSE =

   0.0146
```

- Find the zeros of ha and hb using **“roots”** command. Find also zeros of ha_new. **What is the effect of zeros on the reconstruction quality based on your observations? Comment on** the results.

Although the LSE values are close to each other, zeros of the h_new_a are outside the unit circle, which yields unstable poles on the inverse filter. As a result, the obtained LSE values are higher due to unstability of inverse filter.

```
Zeros of h_a:
   0.3090 + 0.9511i
   0.3090 - 0.9511i
   0.3090 + 0.9511i
   0.3090 - 0.9511i
  -0.8090 + 0.5878i
  -0.8090 - 0.5878i
  -0.8090 + 0.5878i
  -0.8090 - 0.5878i

Zeros of h_b:
  -0.8090 + 0.5878i
  -0.8090 - 0.5878i
  -0.5000 + 0.8660i
  -0.5000 - 0.8660i
   0.5000 + 0.8660i
   0.5000 - 0.8660i
   0.3090 + 0.9511i
   0.3090 - 0.9511i

Zeros of h_a_new:

  -1.1747 + 0.4333i
  -1.1747 - 0.4333i
  -1.2915 + 0.0000i
  -0.7493 + 0.2764i
  -0.7493 - 0.2764i
  -1.0000 + 0.0000i
  -1.0000 - 0.0000i
  -0.7743 + 0.0000i
```