
Image Processing with MATLAB

Lecture 5: Filtering Images

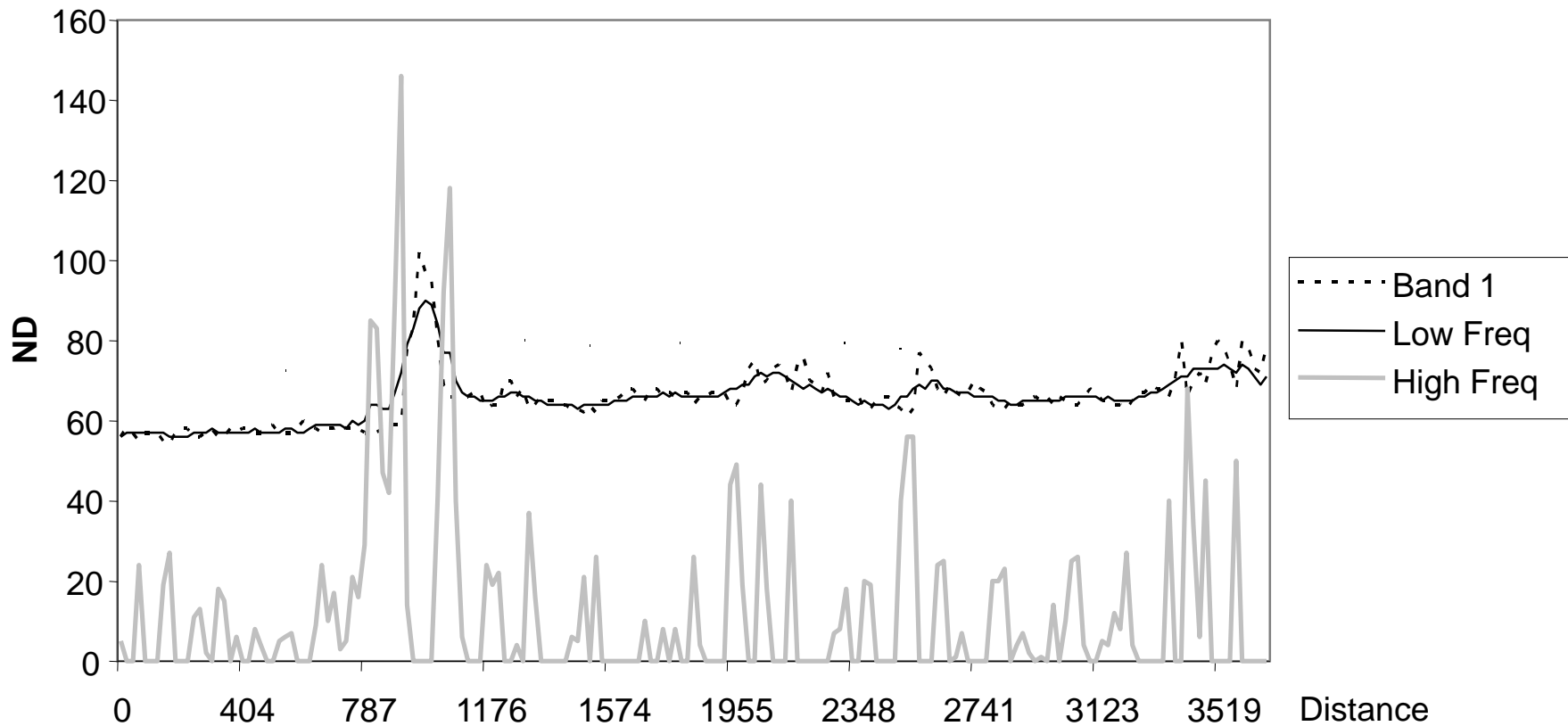
Dr.Eng. Hassan Mohamed

Hassan.husseain@feng.bu.edu.eg

Lecture Contents:

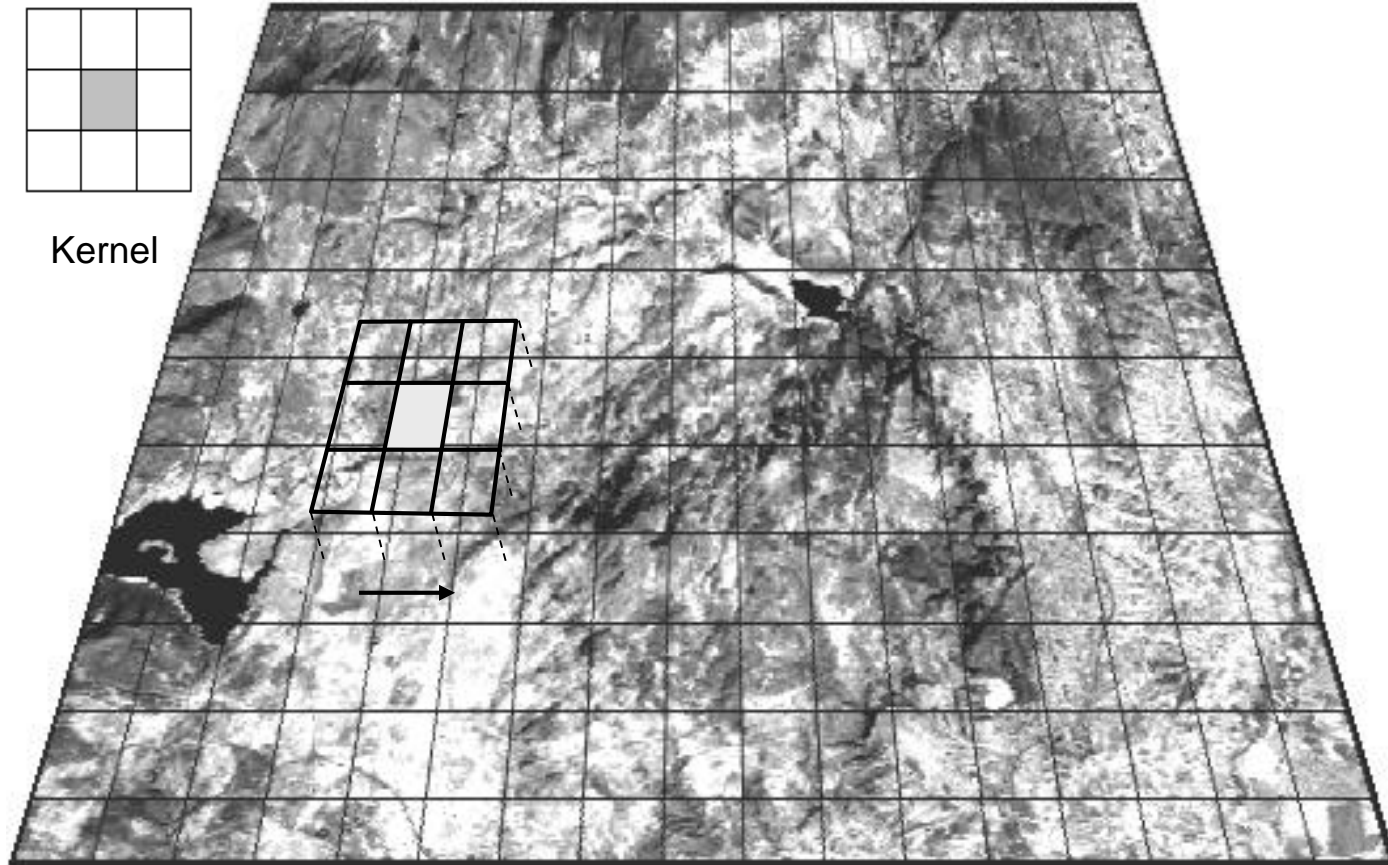
1. Filtering basics
 2. Spatial filters
 3. Filter kernels (Low Pass Smoothings)
 4. Filter kernels (High Pass Sharpening)
-

Filtering basics



A radiometric profile is composed of low and high frequencies of spatial variation

Spatial filters



Filtering through the use of spatial moving window (kernel) that is used to define the neighbor pixels.

Filter kernels (Low Pass Smoothings)

- A low pass filter is the basis for most smoothing methods. An image is smoothed by decreasing the disparity between pixel values by averaging nearby pixels.
- This electronic filter, on the other hand, is the opposite of the high-pass filter which allows frequency that is below the cutoff frequency to pass through.
- The edge content is reduced. +

(1)

1.00	1.00	1.00
1.00	1.00	1.00
1.00	1.00	1.00

(2)

1.00	1.00	1.00
1.00	2.00	1.00
1.00	1.00	1.00

(3)

0.25	0.50	0.25
0.50	1.00	0.50
0.25	0.50	0.25

Filter kernels (High Pass Sharpening)

- A high pass filter is the basis for most sharpening methods. An image is sharpened when contrast is enhanced between adjoining areas with little variation in brightness or darkness.
- The High-Pass filter is the type of electronic filter that allows high frequency waves that are above a certain boundary or a cut off frequency to pass through.
- The edge content is increased. + -

(4)

-1	-1	-1
-1	9	-1
-1	-1	-1

(5)

0	-1	0
-1	5	-1
0	-1	0

Example of digital filtering

12	14	17	24	32	34
10	18	21	35	38	40
25	15	17	27	40	43
18	16	18	24	29	39
14	16	20	20	27	36

1	1	1
1	2	1
1	1	1

0	0	0	0	0	0
0	17	21	29	35	0
0	17	21	28	35	0
0	18	19	25	31	0
0	0	0	0	0	0

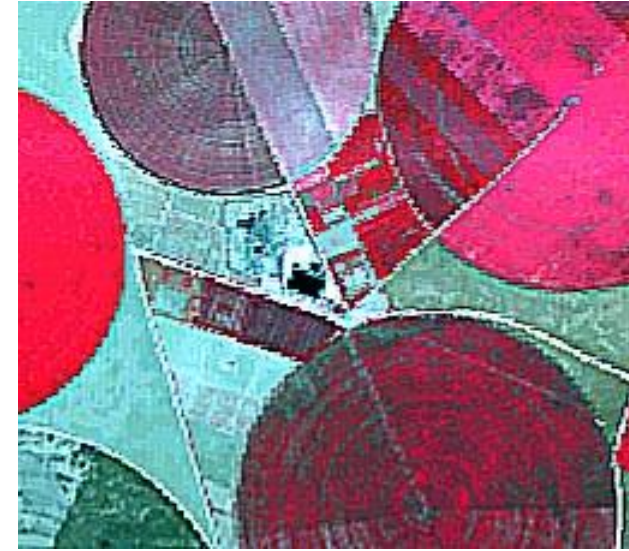
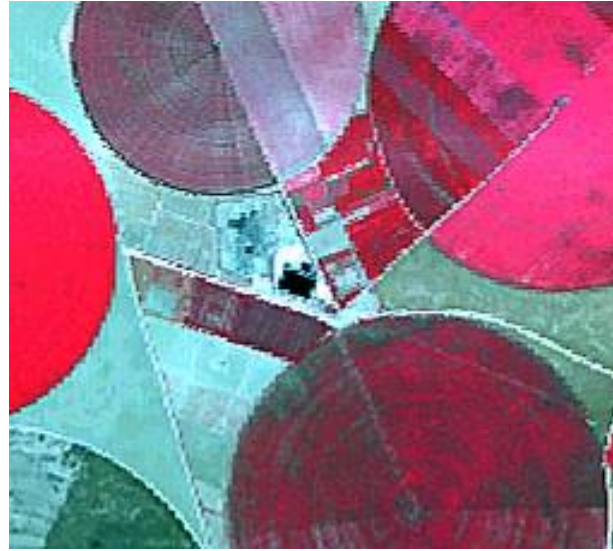
Original Image

Filtered Image

<https://www.youtube.com/watch?v=pFWmqGbkoWg>

Effects of different filters

High pass

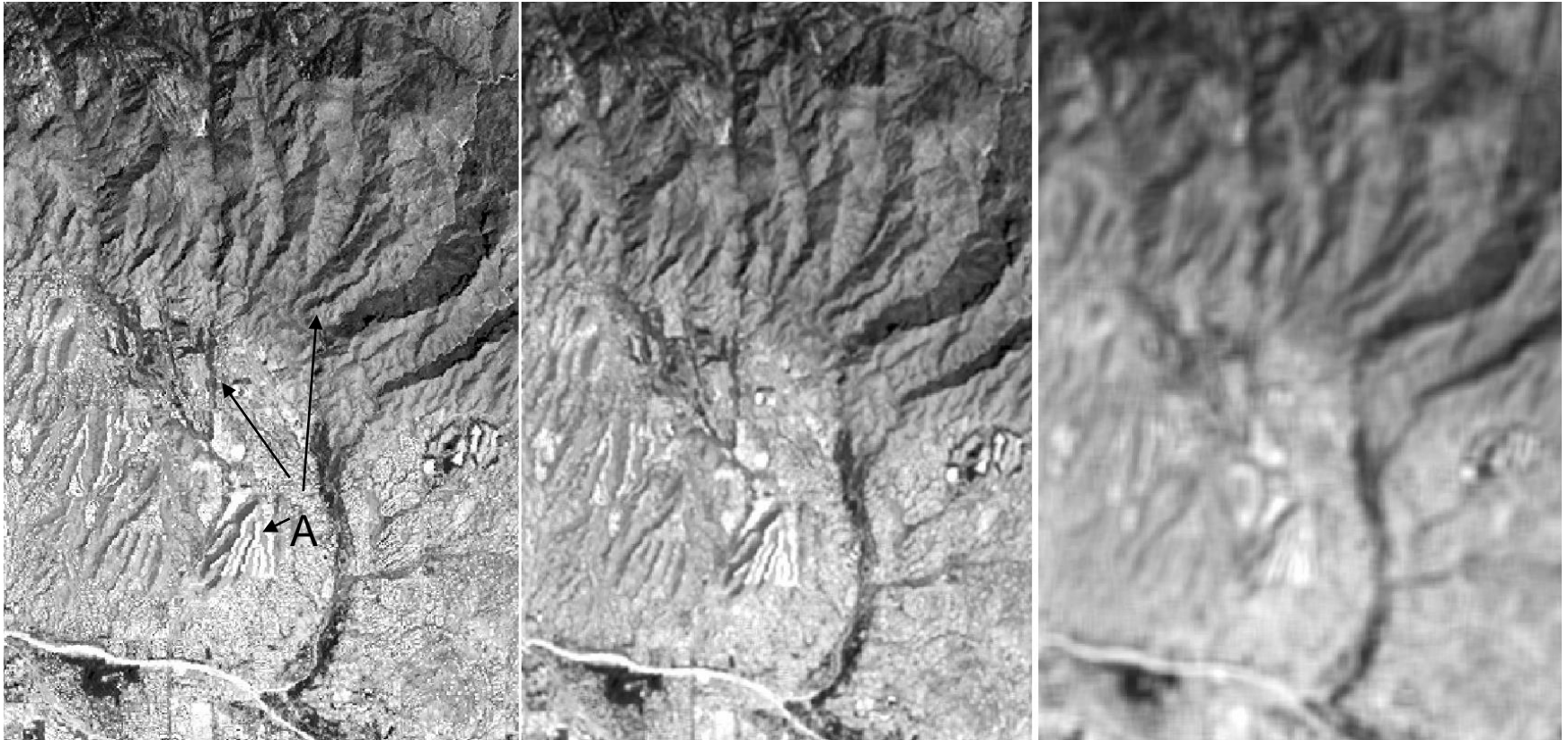


Original



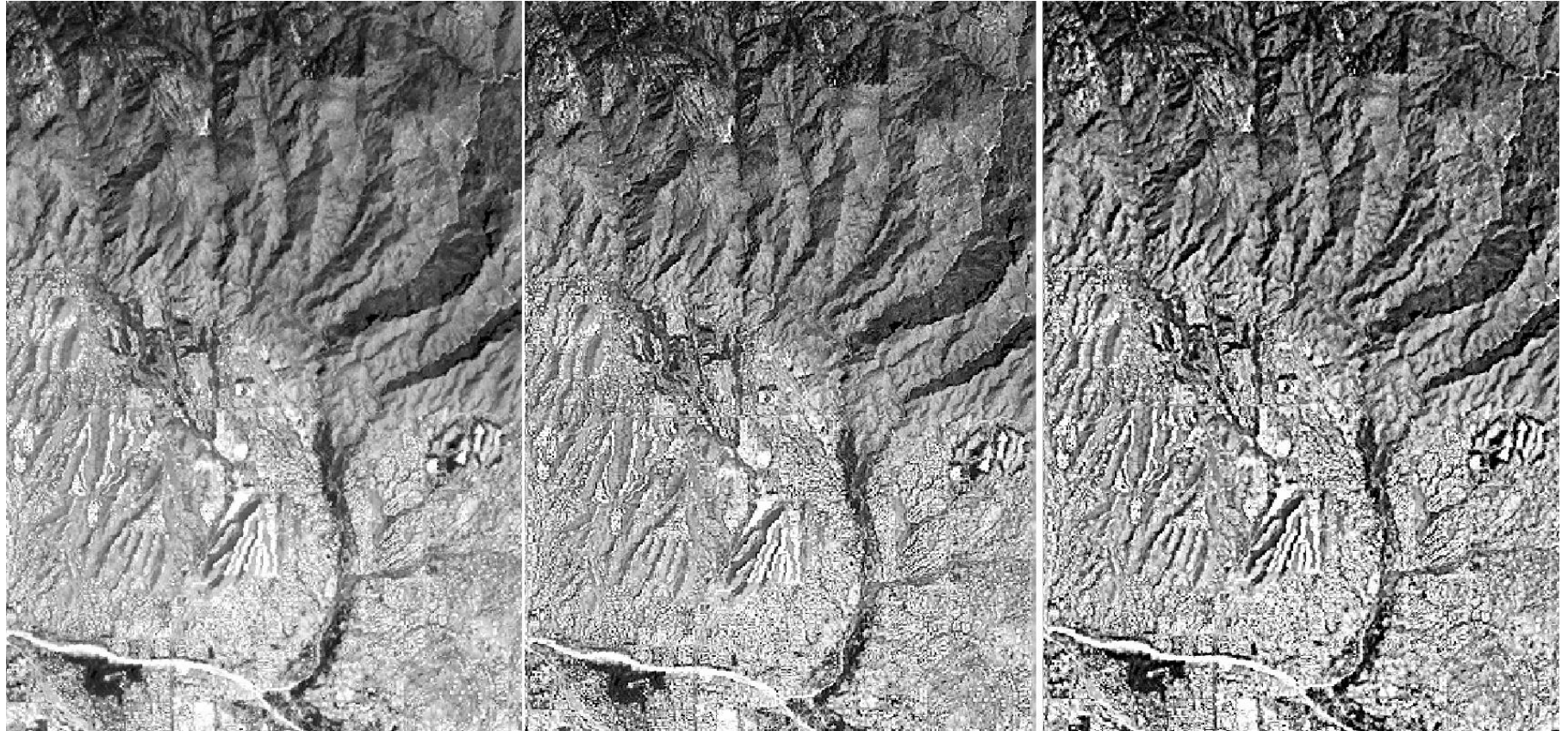
Low pass

Effects of low-pass filters



Low pass filters on a band 3 window of the Tucson ETM+ image. From left to right: original image, 3x3 filtered, and 9x9 filtered images

Effects of high-pass filters



High pass filters on a band 3 window of the Tucson ETM+ image. From left to right: original image, 3x3 filtered, and 9x9 filtered images

Mean Filter

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

```
img = imread('hawk.png');  
mf = ones(3,3)/9;
```

```
>> mf = ones(3,3)/9
```

```
mf =
```

```
0.1111    0.1111    0.1111  
0.1111    0.1111    0.1111  
0.1111    0.1111    0.1111
```

```
img = imread('cameraman.tif');  
imgd = im2double(img);    % imgd in [0,1]  
f = ones(3,3)/9;  
img1 = filter2(f, imgd);  
subplot(121);imshow(img);  
subplot(122);imshow(img1);
```



Imagery Denoising

```
img = imread('cameraman.tif');  
imgd = im2double(img); % imgd in [0,1]  
imgd = imnoise(imgd,'salt & pepper',0.02);  
f = ones(3,3)/9;  
img1 = filter2(f, imgd);  
subplot(121);imshow(imgd);  
subplot(122);imshow(img1);
```



Median filter - medfilt2()

```
I = imread('cameraman.tif');  
J = imnoise(I, 'salt & pepper', 0.02);  
K = medfilt2(J);  
subplot(121); imshow(J);  
subplot(122); imshow(K);
```



Removing noise in RGB image

```
I = imread('hawk.png');  
J = imnoise(I, 'salt & pepper', 0.2);  
  
% filter each channel separately  
r = medfilt2(J(:, :, 1), [3 3]);  
g = medfilt2(J(:, :, 2), [3 3]);  
b = medfilt2(J(:, :, 3), [3 3]);  
  
% reconstruct the image from r,g,b channels  
K = cat(3, r, g, b);
```

```
figure  
subplot(121); imshow(J);  
subplot(122); imshow(K);
```



fspecial()

```
h = fspecial(type)
```

```
h = fspecial(type, parameters)
```

Value	Description
average	Averaging filter
disk	Circular averaging filter (pillbox)
gaussian	Gaussian lowpass filter
laplacian	Laplacian of Gaussian filter
motion	Approximates the linear motion of a camera
prewitt	Prewitt horizontal edge-emphasizing filter
sobel	Sobel horizontal edge-emphasizing filter

Gaussian High pass Filter

```
function GaussianHighpass
a=imread('cameraman.tif');
figure(1)
imshow(a)
[m n]=size(a);
f_transform=fft2(a);
f_shift=fftshift(f_transform);
p=m/2;
q=n/2;
d0=70;
for i=1:m
for j=1:n
distance=sqrt((i-p)^2+(j-q)^2);
low_filter(i,j)=1-exp(-(distance)^2/(2*(d0^2)));
end
end
filter_apply=f_shift.*low_filter;
image_original=ifftshift(filter_apply);
image_filter_apply=abs(ifft2(image_original));
figure(2)
imshow(image_filter_apply,[])
```



https://www.youtube.com/watch?v=z_dkB7vtDL0

Supplementary files:

- MATLAB Tutorial:

http://www.mathworks.com/products/matlab/matlab_tutorial.html

- MATLAB documentation:

<http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml>

**Please don't use this presentation without getting a
permeation from its original owner**

Dr.Eng. Hassan Mohamed
