6th tutorial in IVP

4th *April* 2024

Written by - Anurag Paul, 20EC01045.

Input image.

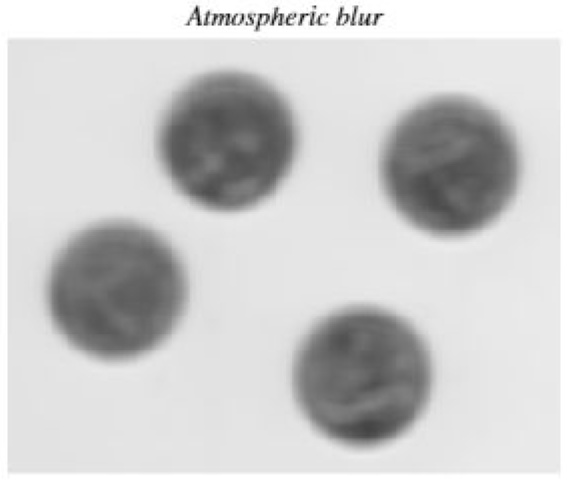
```
i = imread('eight.tif'); imshow(i);
[h1, h2] = size(i);
title(['Original Image, Size: ' num2str(h1)...
    ' $\times$ ' num2str(h2)], 'interpreter', ...
    'latex');
```

```
% xlabel('\it U', 'interpreter', 'latex')
```

To grayscale the image.

```
% g = rgb2gray(i); imshow(g);
% title('\it{Greyed Image}','interpreter',
% 'latex');
% max(g, [], 'all')
```

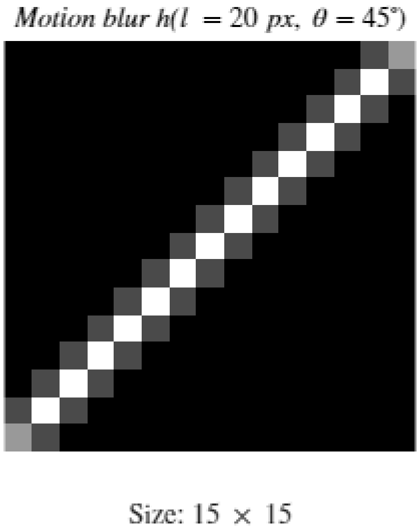
```
H = fspecial("gaussian",11,5);
title('\it Atmospheric blur', 'interpreter', ...
    'latex');
```



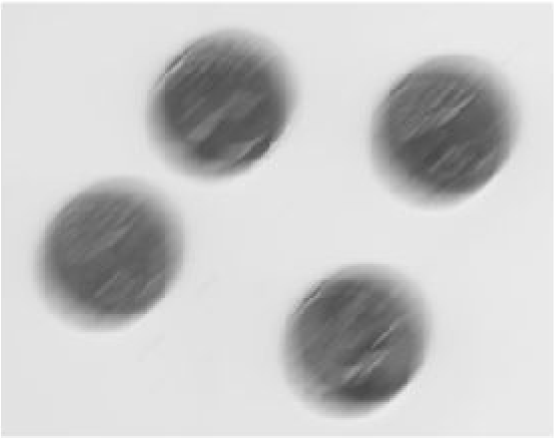
```
Blur = imfilter(i,H,'replicate');
% imshow(Blur);
```

20 *px*

```
H = fspecial('motion',20,45); imshow(H,[])
% title('\it Motion blur H', 'interpreter',
% 'latex');
title(['\it Motion blur h(l $= 20$ $px,$ $' ...
    '\theta = 45^\circ$)'], 'interpreter','latex');
[h1, h2] = size(H);
xlabel(['Size: ' num2str(h1) ' $\times$ '...
    num2str(h2)], 'interpreter', 'latex')
```

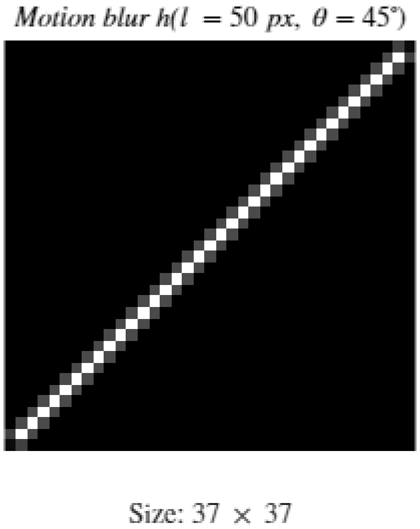


```
MotionBlur = imfilter(i,H,'replicate');
imshow(MotionBlur);
```

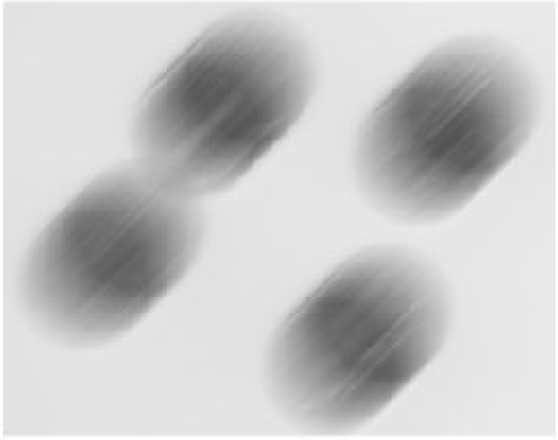


50 *px*

```
H = fspecial('motion',50,45); imshow(H,[])
title(['\it Motion blur h(l $= 50$ $px,$ $' ...
    '\theta' ...
    ' = 45^\circ$)'], 'interpreter','latex');
MotionBlur = imfilter(i,H,'replicate');
% title('\it Motion blur H', 'interpreter',
% 'latex');
[h1, h2] = size(H);
xlabel(['Size: ' num2str(h1) ' $\times$ '...
    num2str(h2)], 'interpreter', 'latex')
```



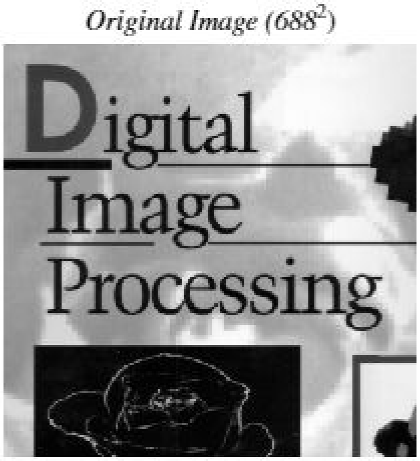
```
imshow(MotionBlur);
```



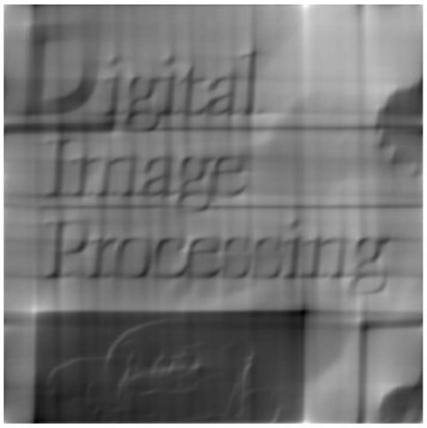
Out of focus

```
H = fspecial('disk',10);
% blurred =
imfilter(i,H,'replicate');
title('\it Out of focus', 'interpreter','latex');
% imshow(blurred);
```

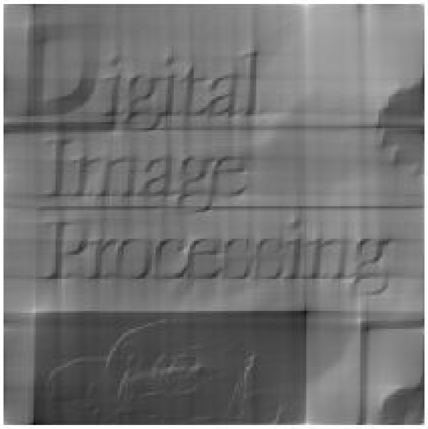
```
im = imread(['http://users.rowan.edu/' ...
    '~shreek/' ...
    'fall09/dip/lab3/GW_Fig5_26.jpg']);
imshow(im);
[m, n] = size(im); [u, v] = meshgrid(1:m, 1:n);
title(['\it Original Image (' num2str(m) ...
    '$^2$)'], 'interpreter','latex');
```

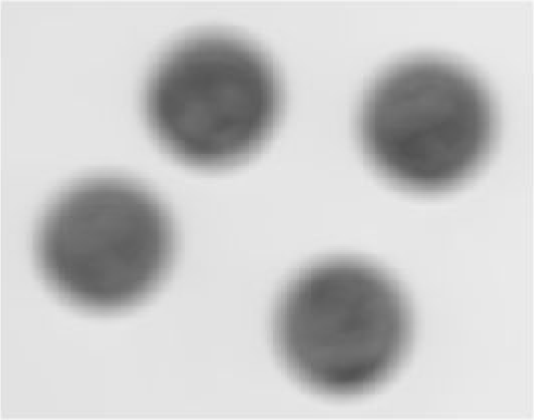


```
k = .001;

% x = pi*(u*a+v*b);
% x = pi*((u-m/2)*a+(v-n/2)*b);
Huv=exp(-k*(u.^2 + v.^2).^(5/6));
% Hmb = (T./x).*(sin(x)).*(exp(-1j*x));
MB = ifft2(fft2(im).*Huv);
% MB = ifft2(fft2(im).*Hmb);
% MB = ifft2(fft2(double(im).*(-1).^(u+v)).*
% Hmb.*(-1).^(u+v));
% imshow(log(abs(MB)+1), []);
imshow(real(MB), []);
```



```
k = 1e-4;
% x = pi*(u*a+v*b);
% x = pi*((u-m/2)*a+(v-n/2)*b);
Huv=exp(-k*(u.^2 + v.^2).^(5/6));
% Hmb = (T./x).*(sin(x)).*(exp(-1j*x));
MB = ifft2(fft2(im).*Huv);
imshow(real(MB), []);
```





G

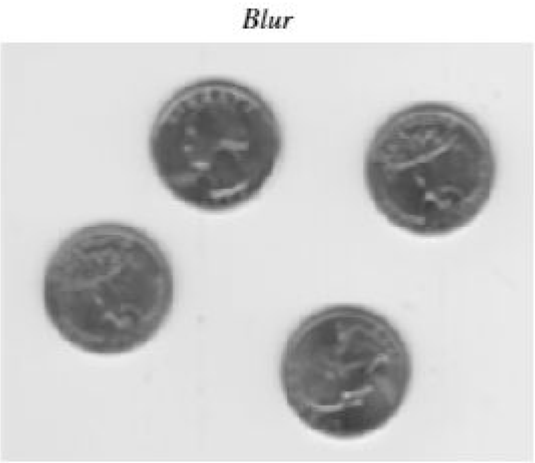
```
L = 5; % kernel width
sx=3;
sy=10;
theta=0;

I = imread('cameraman.tif');
x = -L:1.0:L;

[X,Y] = meshgrid(x,x);
rX = X.*cos(theta)-Y.*sin(theta);
rY = X.*sin(theta)+Y.*cos(theta);
H1 = exp(-((rX./sx).^2)-((rY./sy).^2));
Hflag = double((0.*rX+rY)>0);
H1 = H1.*Hflag;
comet_kernel = H1/sum((H1(:)));

smearedImage = conv2(double(I),comet_kernel, ...
    'same');

imshow(smearedImage,[]);
```

```
MotionBlur = imfilter(double(I),comet_kernel, ...
    'replicate');
imshow(MotionBlur);
```

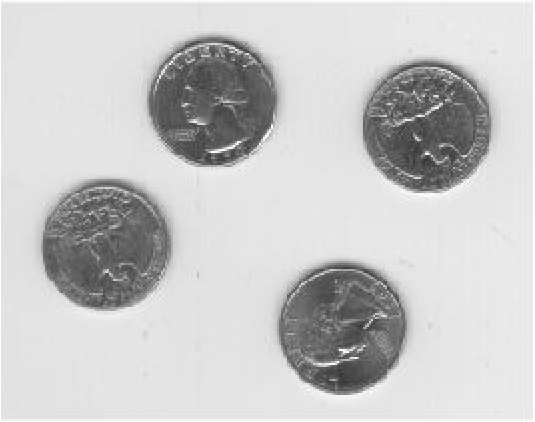
```
addpath('Subs/IVP');
```

```
n=.001; f=i;
% freadbin('lenna.256',N,N);
% imagesc(f)
```

```
b=ones(4,4)/4^2; F=fft2(f);
[h1, h2] = size(i);
B=fft2(b,h1,h2);
G=F.*B; g=ifft2(G);
% +10*randn(h1,h2);
G=fft2(g); bl = abs(ifft2(G));
imshow(uint8(bl))
title('\it Blur', 'interpreter','latex');
```



```
BF=(abs(B)<n);
%B(BF)=max(max(B))/1.5;
B(BF)=n; % V /0
H=1./B; I=G.*H;
im=uint8(abs(ifft2(I)));
imshow(im)
```



S_2

```
% estimated_nsr = noise_var / var(I(:));
wnr3 = deconvwnr(MotionBlur, H);
figure, imshow(wnr3)
title(['Restoration of Blurred, Noisy Image' ...
    ' Using Estimated NSR']);
```

S_8

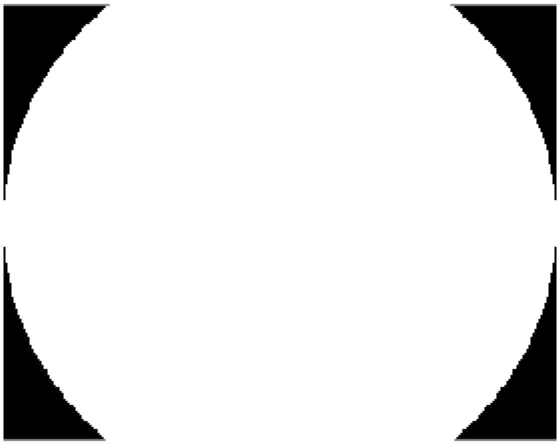
```
H = fspecial('motion',20,45);
%  = imfilter(i,H,'replicate');
MotionBlur = imfilter(i,H,'conv','circular');
imshow(MotionBlur);
title(['\it Motion blur, $l = 20$ $px,$ $' ...
    '\theta' ...
    ' = 45^\circ$'], 'interpreter','latex');
```

```
Ioriginal = imread('cameraman.tif');
imshow(Ioriginal)
title('Original Image')
```

```
PSF = fspecial('motion',21,41);
Idouble = im2double(Ioriginal);
blurred = imfilter(Idouble,PSF,'conv','circular');
imshow(blurred)
title('Blurred Image')
```

```
% 65
```

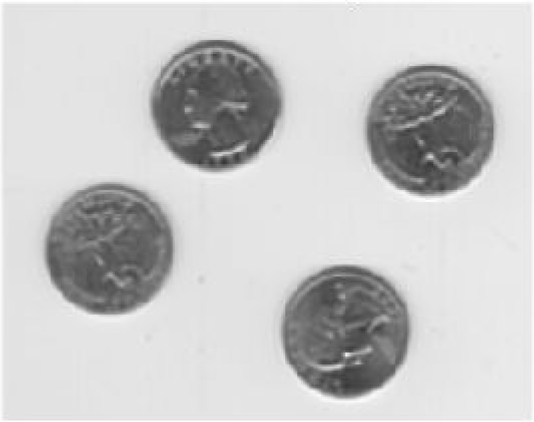
```
[r, c] = size(i);
u = -r/2:r/2-1; v = -(c-1)/2:(c-1)/2;
[uu, vv] = meshgrid(u, v);
% Cut-off Frequency
H = double(sqrt(uu.^2+vv.^2) < max(r,c)/2); % 150
% H = double(sqrt(uu.^2+vv.^2) < min(r,c)/2); % 150
figure; imshow(H')
title('\it Ideal LPF' , ...
    'interpreter', 'latex');
```



Ideal LPF

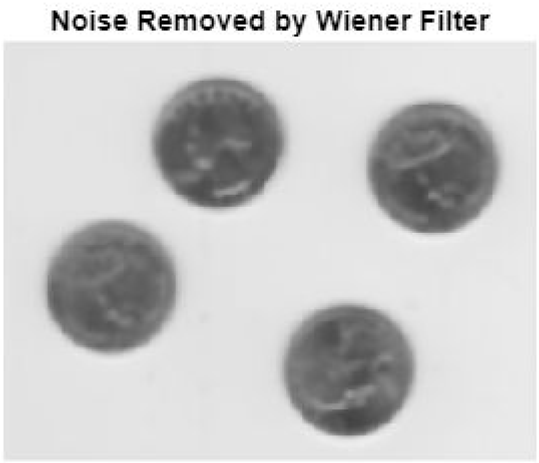
```
wnr1 = uint8(deconvwnr(bl,b));
imshow(wnr1)
title('Restored Blurred Image')
```

```
luc1 = uint8(deconvlucy(bl,b,5));
imshow(luc1)
title('Restored Image')
```



Restored Image

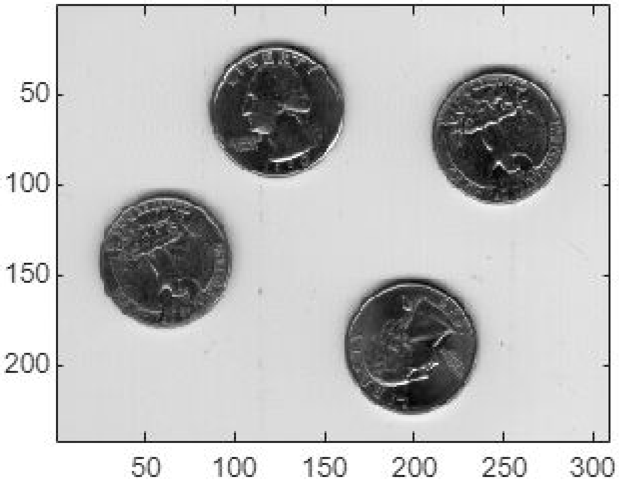
```
K = uint8(wiener2(bl,[5 5]));
imshow(K);
title('Noise Removed by Wiener Filter');
```



```
% BF=(abs(B)<n);
```

```
BF =

  0x1 empty double column vector
```

```
%B(BF)=max(max(B))/1.5;
B(BF)=n;
H=1./(H.*B + (2-H)*n);
I=G.*H;
im=abs(ifft2(I));
imagesc(im)
```



```
% Get the noise-only image
noiseOnlyImage = abs(bl - double(i));

% Calculate the mean of each
signalMean = mean(i(:));
signalMean = 198.2587
```

```
% noiseOriginal = mean(noisyImage(:))
noiseOnlyMean = mean(noiseOnlyImage(:))
```

```
noiseOnlyMean = 5.6073
```

```
% Get ratio of the means
snr1 = signalMean/noiseOnlyMean
```

```
snr1 = 35.3571
```

```
MSE = sum(noiseOnlyImage.^2, 'all')/h1/h2
```

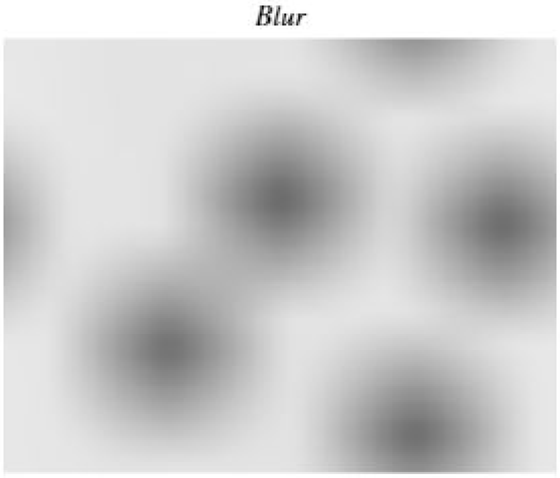
```
MSE = 190.4587
```

```
PSNR = 10*log10(255^2/MSE)
```

```
ans = 25.3328
```

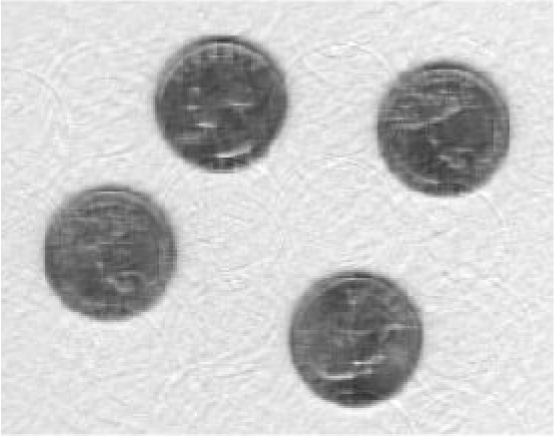
64

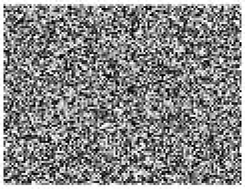
```
N=64; n=.001;
b=ones(N,N)/N^2; F=fft2(f);
[h1, h2] = size(i);
B=fft2(b,h1,h2);
G=F.*B; g=ifft2(G);
% +10*randn(h1,h2);
G=fft2(g); figure;
imshow(uint8(abs(ifft2(G))))
title('\it Blur', 'interpreter','latex');
```



```


BF=(abs(B)<n);
%B(BF)=max(max(B))/1.5;
```

```
B(BF)=n; % V /0
H=1./B; I=G.*H;
im=uint8(abs(ifft2(I))); figure;
imshow(im)
```





```
signalMean = 100
noiseOriginal = 99.9879
noiseOnlyMean = 5.0069
snr1 = 19.9724
snr2 = 166.1157
```