

# FFT

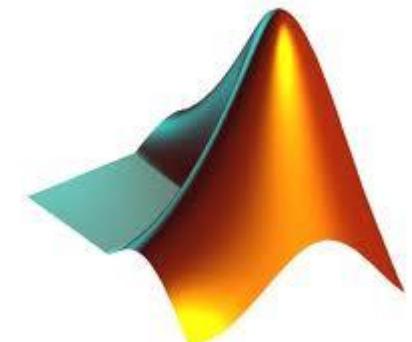
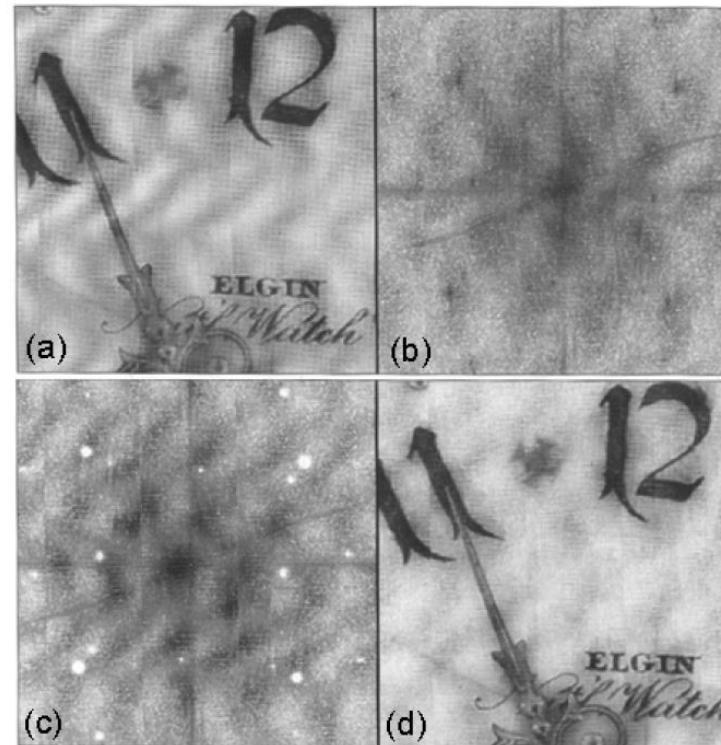
## Cvičenia z Počítačového Videnia

# FFT

Akákoľvek funkcia  $f(x)$  môže byť vyjadrená ako vážený súčet sínusov a kosínusov

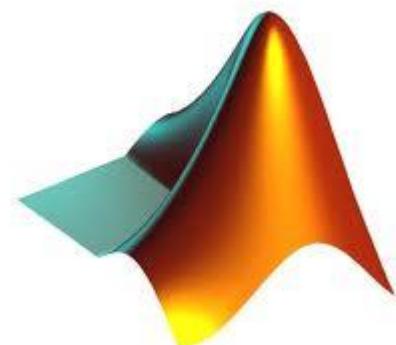
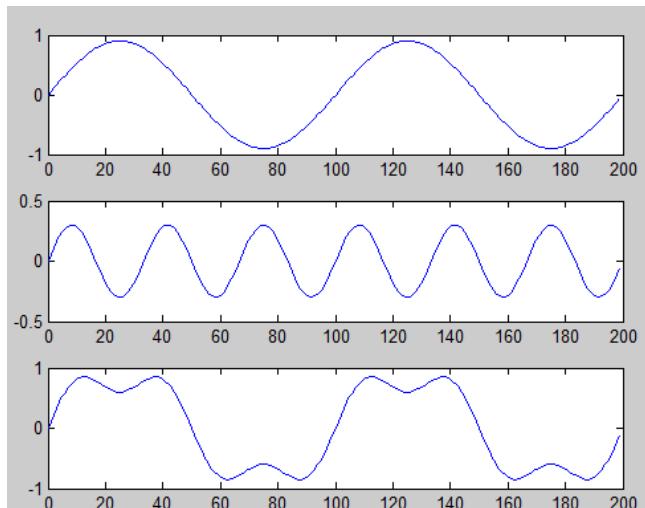
Využitie:

- lowpass filter
- highpass filter
- odstraňovanie artefaktov z obrazu



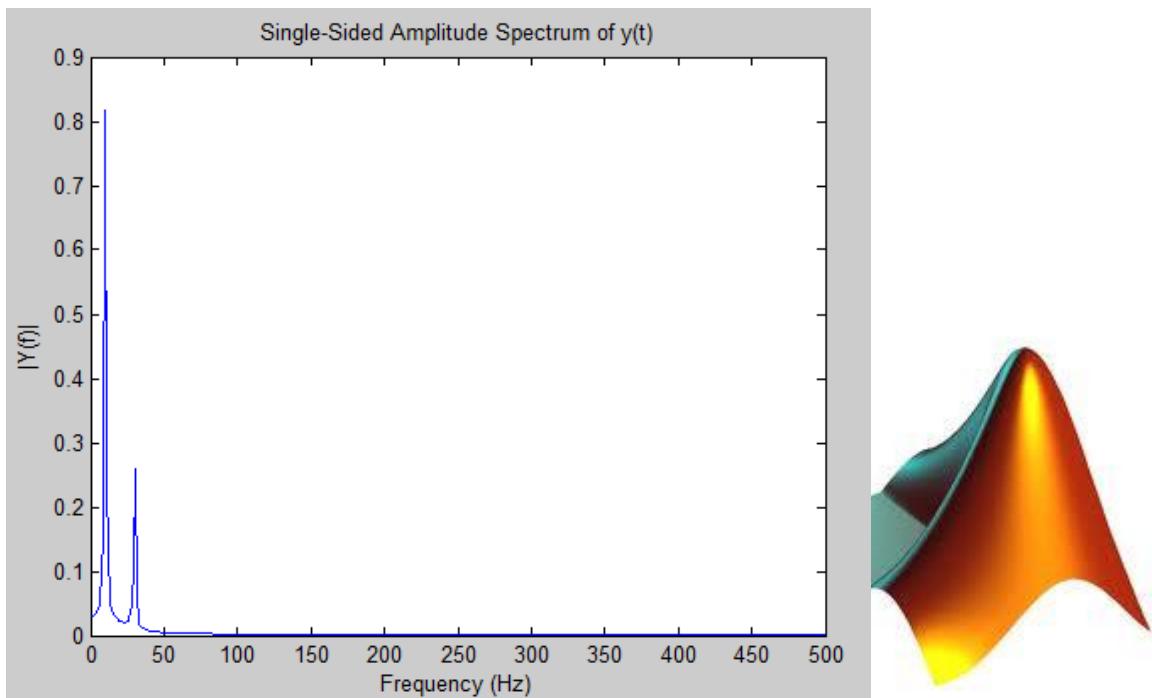
# FFT - Skladanie signálu MATLAB

```
Fs = 1000; % Sampling frequency  
T = 1/Fs; % Sample time  
L = 1000; % Length of signal  
t = (0:L-1)*T; % Time vector  
  
y1 = 0.9*sin(2*pi*10*t);  
y2 = 0.3*sin(2*pi*30*t);  
y3 = y1 +y2;  
  
figure, subplot(3,1,1),plot(Fs*t(1:200),y1(1:200));  
subplot(3,1,2),plot(Fs*t(1:200),y2(1:200));  
subplot(3,1,3),plot(Fs*t(1:200),y3(1:200));
```



# FFT - Skladanie signálu MATLAB

```
NFFT = 2^nextpow2(L); % Next power of 2 from length of y  
Y = fft(y3,NFFT)/L;  
f = Fs/2*linspace(0,1,NFFT/2);  
  
% Plot single-sided amplitude spectrum.  
figure  
plot(f,2*abs(Y(1:NFFT/2)))  
title('Single-Sided Amplitude Spectrum of y(t)')  
xlabel('Frequency (Hz)')  
ylabel('|Y(f)|')
```



# fftshift

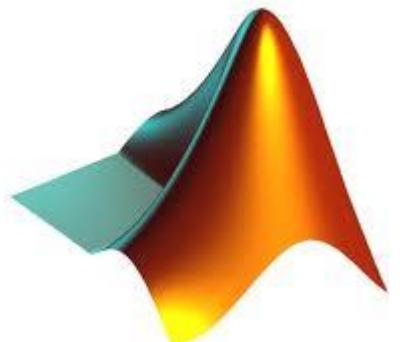
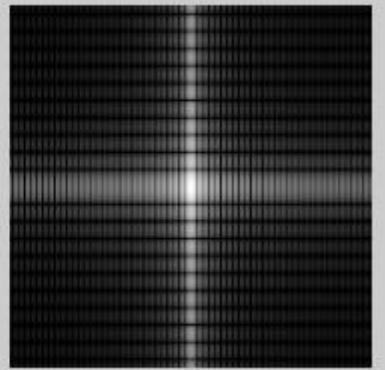
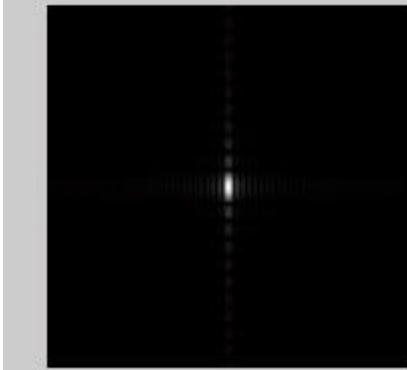
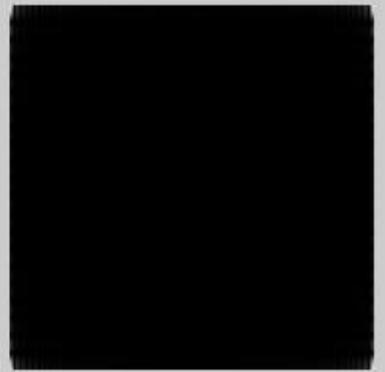
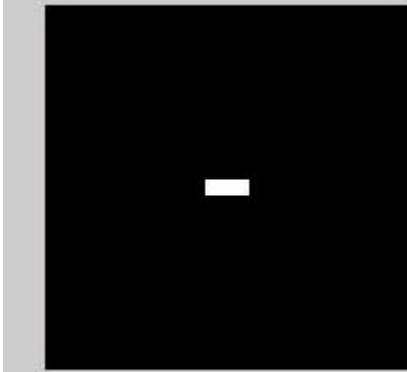
```
siz=512;  
siz2=floor(siz/2);
```

```
f1 = zeros(siz);  
f1(siz2-10:siz2+10,siz2-30:siz2+30) = 1;  
subplot(2,2,1);imshow(f1,[]);
```

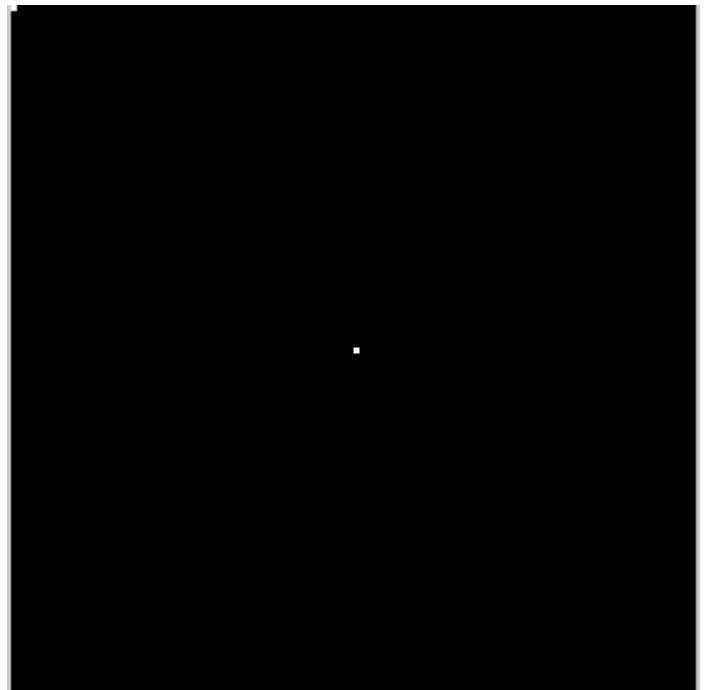
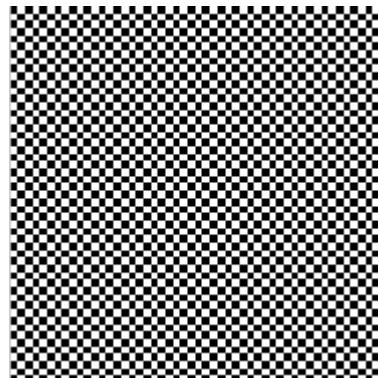
```
F = fft2(f1, siz,siz);  
S = abs(F);  
subplot(2,2,2); imshow(S,[]);
```

```
Fc = fftshift(F);  
S1 = abs(Fc);  
subplot(2,2,3); imshow(S1,[]);
```

```
S2 = log(1+S1);  
subplot(2,2,4);imshow(S2,[]);
```

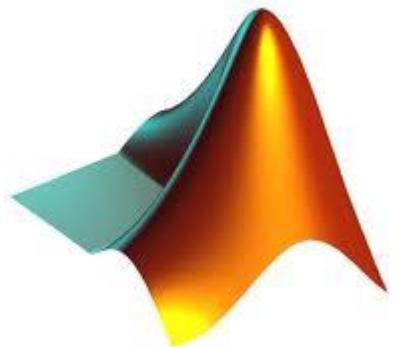


# FFT2

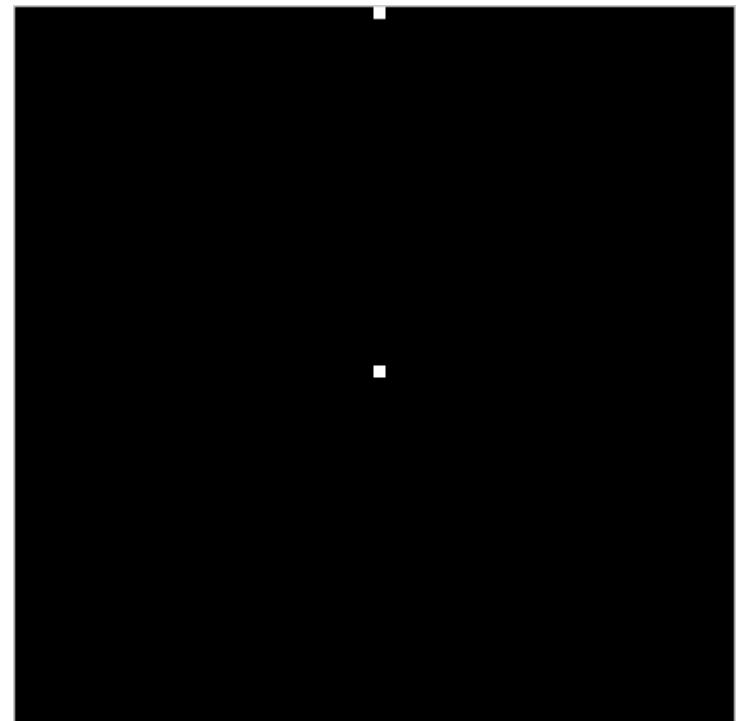
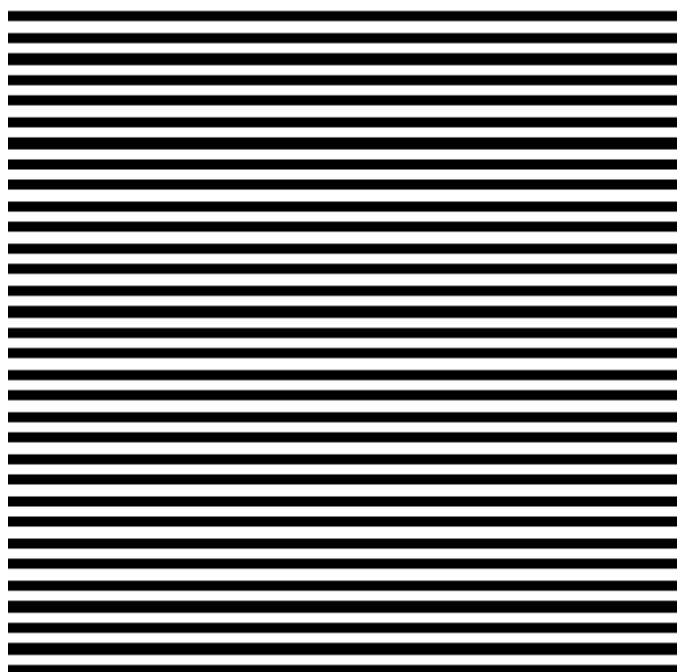


```
y=double(checkerboard(1,64,64)>0.5);  
figure, imshow(y)
```

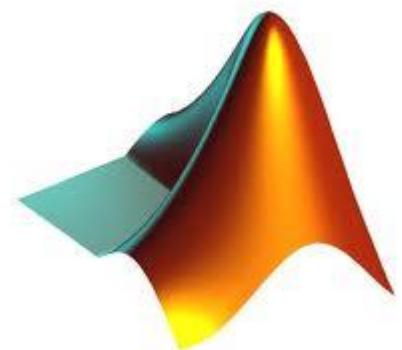
```
Y=fft2(y);  
figure  
imshow(abs(fftshift(Y)) ,[], 'InitialMagnification','fit')
```



# FFT



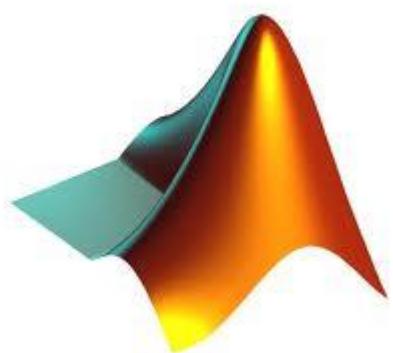
```
y=zeros(64);  
y(1:2:end,:)=1;
```



# FFT

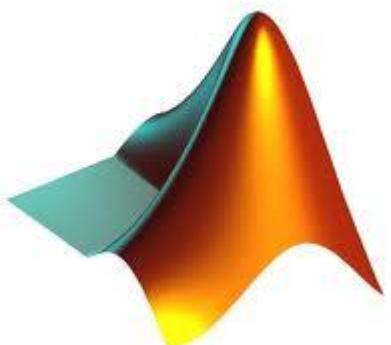
```
y=zeros(64);  
y(1:4:end,:)=1;  
y(2:4:end,:)=1;
```

```
y=zeros(64);  
y(1:4:end,:)=1;  
y(:,1:4:end)=1;
```



# FFT

```
Fs = 256; % Sampling frequency  
T = 1/Fs; % Sample time  
L = 512; % Length of signal  
t = (0:L-1)*T; % Time vector  
y = (0.5+0.5*cos(2*pi*t'))*ones(1,L);
```



# FFT

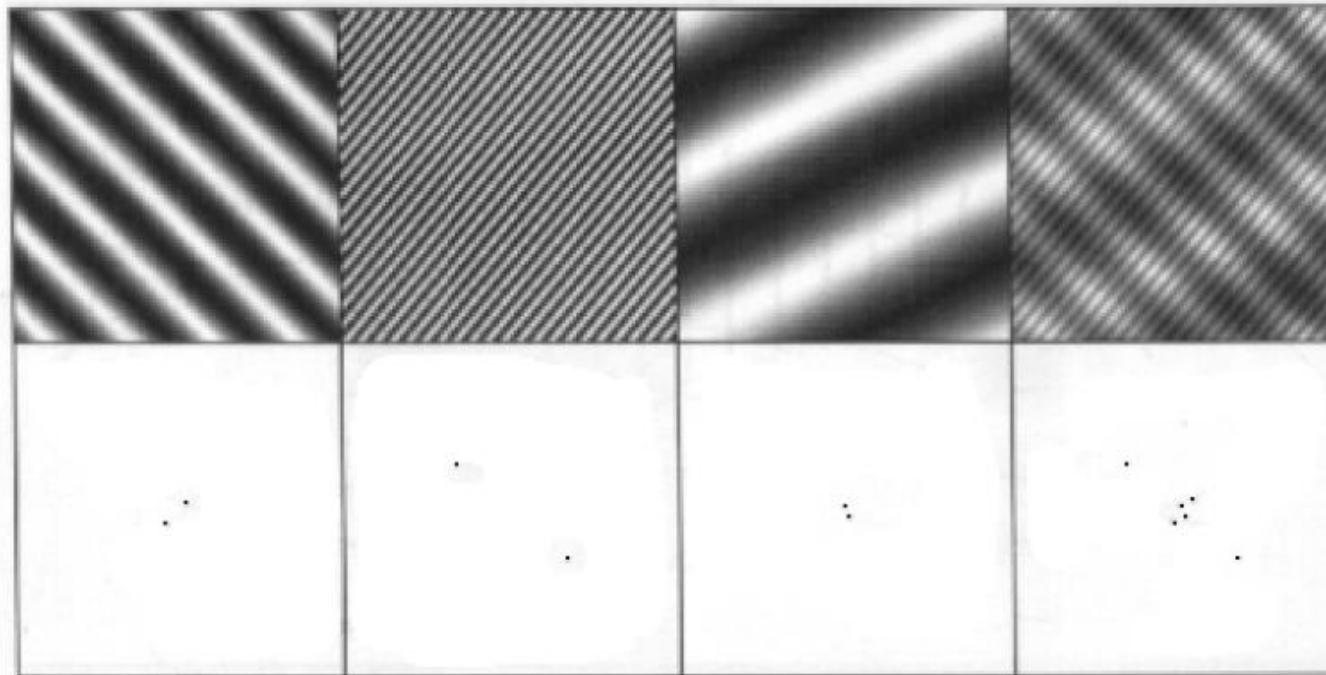
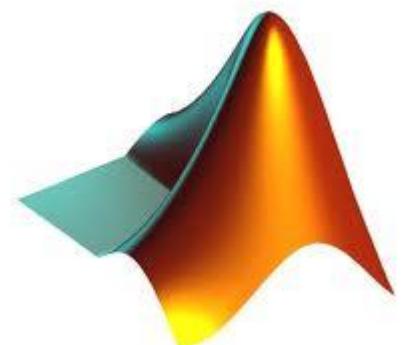


Figure 2: Images with perfectly sinusoidal variations in brightness: The first three images are represented by two dots. You can easily see that the position and orientation of those dots have something to do with what the original image looks like. The 4th image is the sum of the first three.

(Taken from p.177 of [1].)

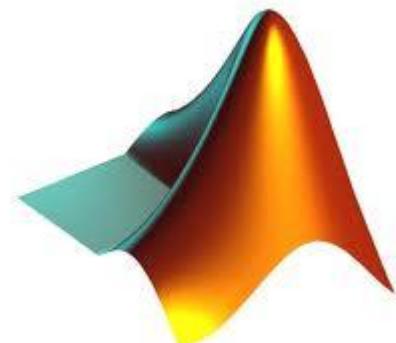


# FFT

```
imSize = 256;
X = 1:imSize; % X is a vector from 1 to imageSize
X0 = (X / imSize) - .5; % rescale X -> -.5 to .5
[Xm Ym] = meshgrid(X0, X0);
theta = 45; % grating orientation
lamda = 40; % wavelength (number of pixels per cycle)
phase = .25; % phase (0 -> 1)
freq = imSize/lamda; % compute frequency from wavelength
phaseRad = (phase * 2* pi); % convert to radians: 0 -> 2*pi
thetaRad = (theta / 360) * 2*pi; % convert theta (orientation) to radians
Xt = Xm * cos(thetaRad); % compute proportion of Xm for given orientation
Yt = Ym * sin(thetaRad); % compute proportion of Ym for given orientation
XYt = [ Xt + Yt ]; % sum X and Y components
XYf = XYt * freq * 2*pi; % convert to radians and scale by frequency
y = sin( XYf + phaseRad); % make 2D sinewave
```

```
figure, imshow(y,[], 'InitialMagnification','fit')
```

```
Y=fft2(y);
F2=abs(fftshift(Y));
figure, imshow(F2 ,[], 'InitialMagnification','fit')
```

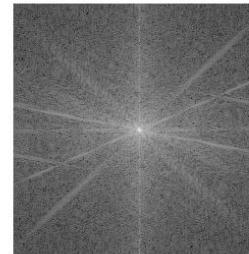


# FFT rekonštrukcia

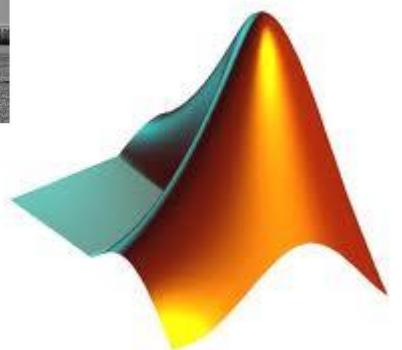
```
a = imread('cameraman.tif');  
figure; imshow(a,[]);
```



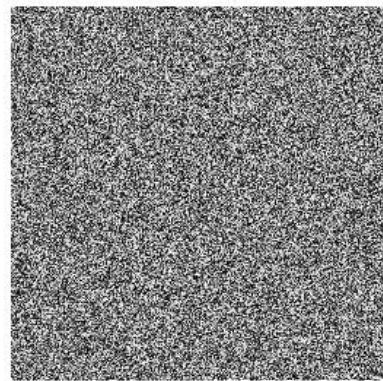
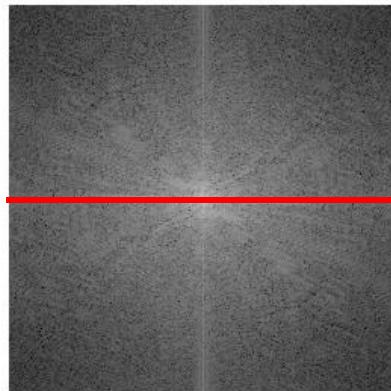
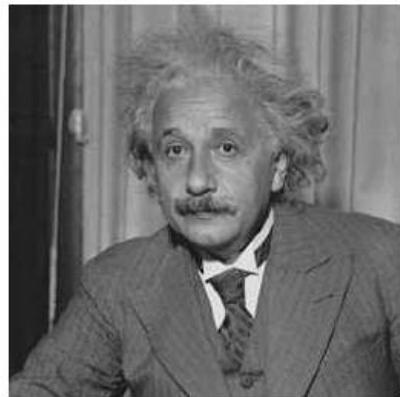
```
Da = fft2(a);  
figure; imshow(log(abs(fftshift(Da))),[]);
```



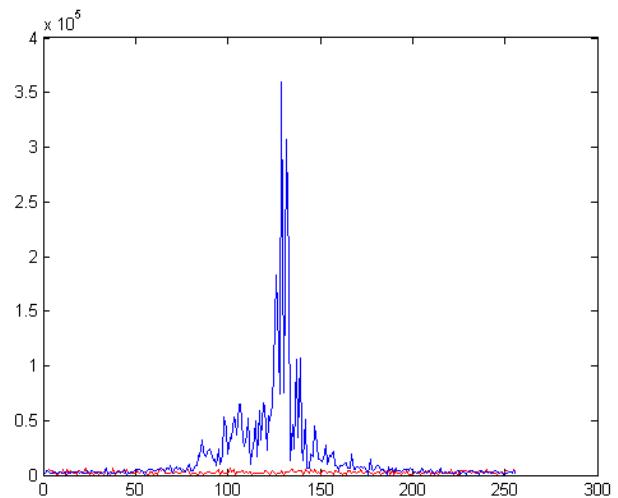
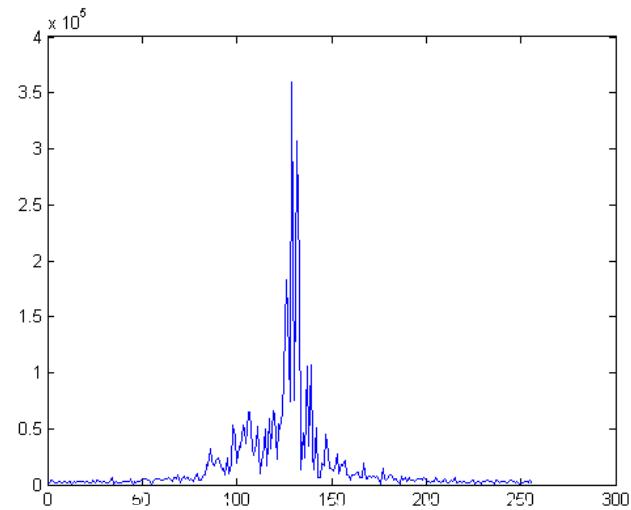
```
b = real(ifft2(Da));  
figure; imshow(b,[]);
```



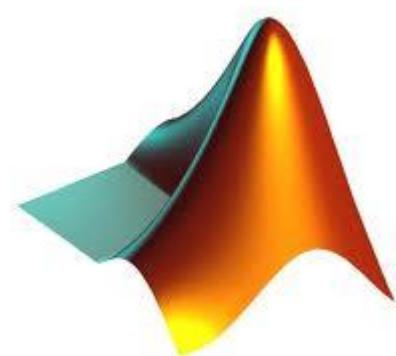
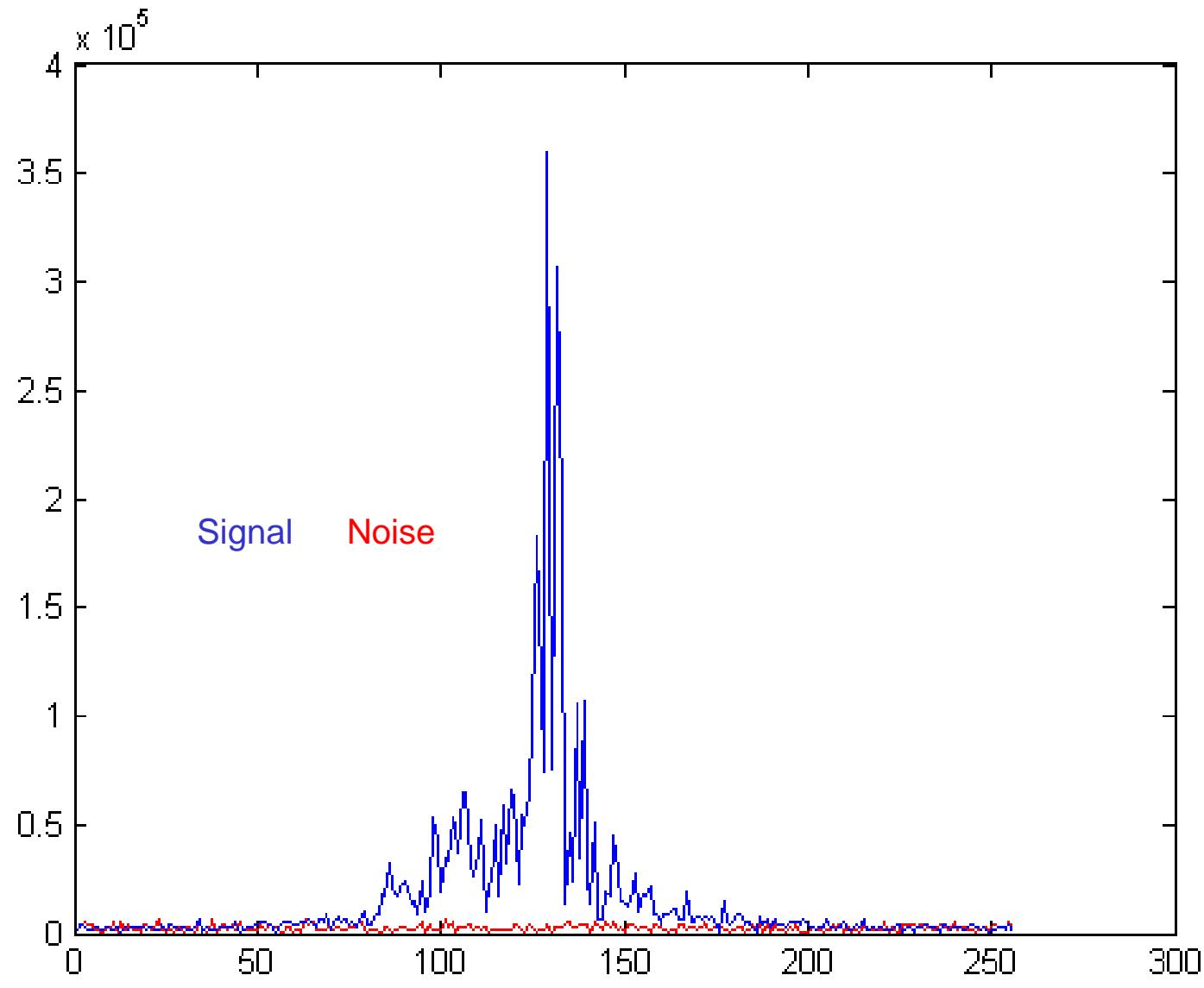
# FFT odstránenie šumu



Signal      Noise

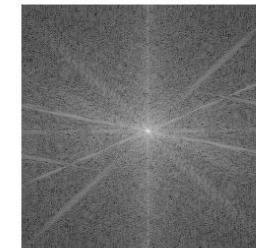


# FFT

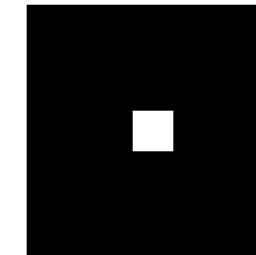


# FFT dolnopriepustný filter

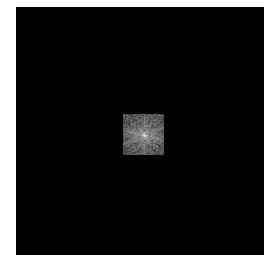
```
a = imread('cameraman.tif');  
Da = fft2(a);  
figure; imshow(log(abs(fftshift(Da))),[]);
```



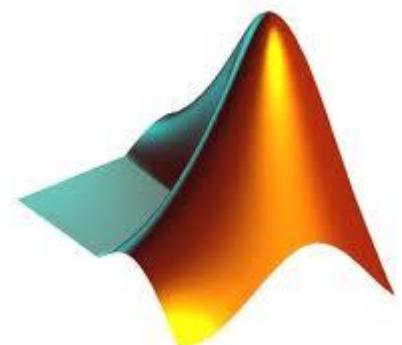
```
[m n]=size(Da);  
H = zeros(m,n);  
H(floor(m/2)-20:floor(m/2)+20,floor(n/2)-20:floor(n/2)+20) = 1;  
figure; imshow(H,[]);
```



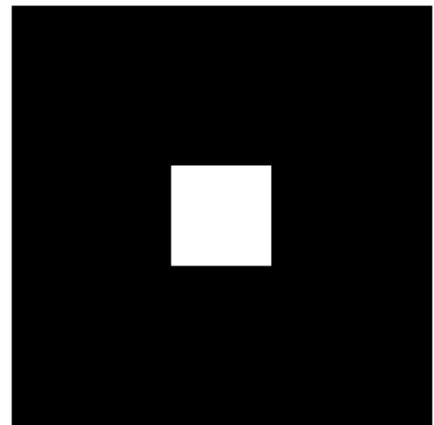
```
Db = fftshift(fftshift(Da).*H);
```



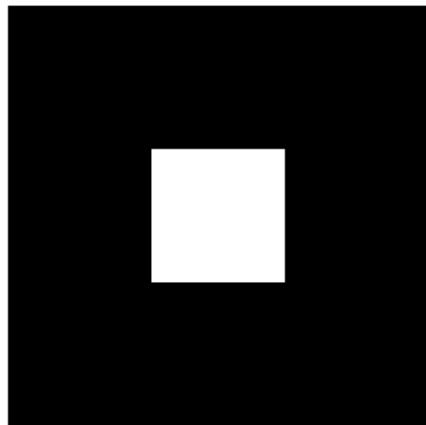
```
b = real(ifft2(Db));  
figure; imshow(b,[]);
```



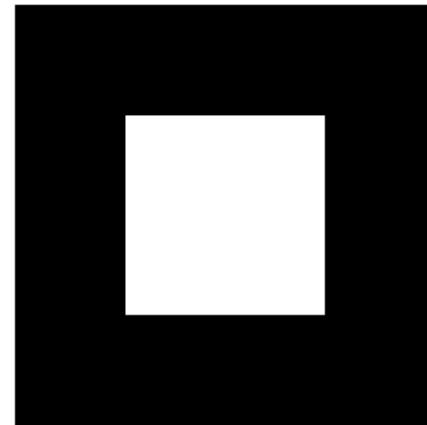
# FFT rekonštrukcia



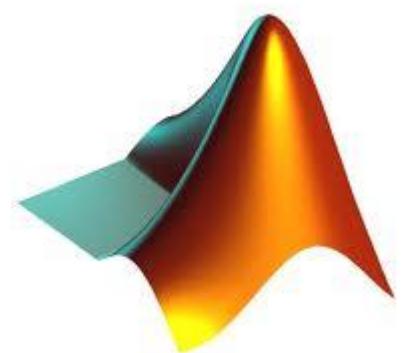
61x61



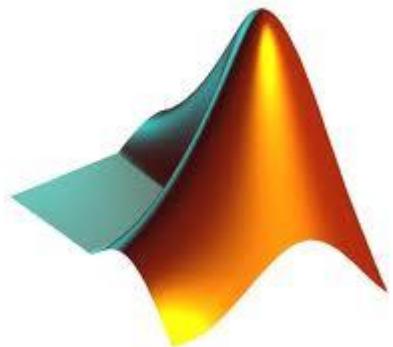
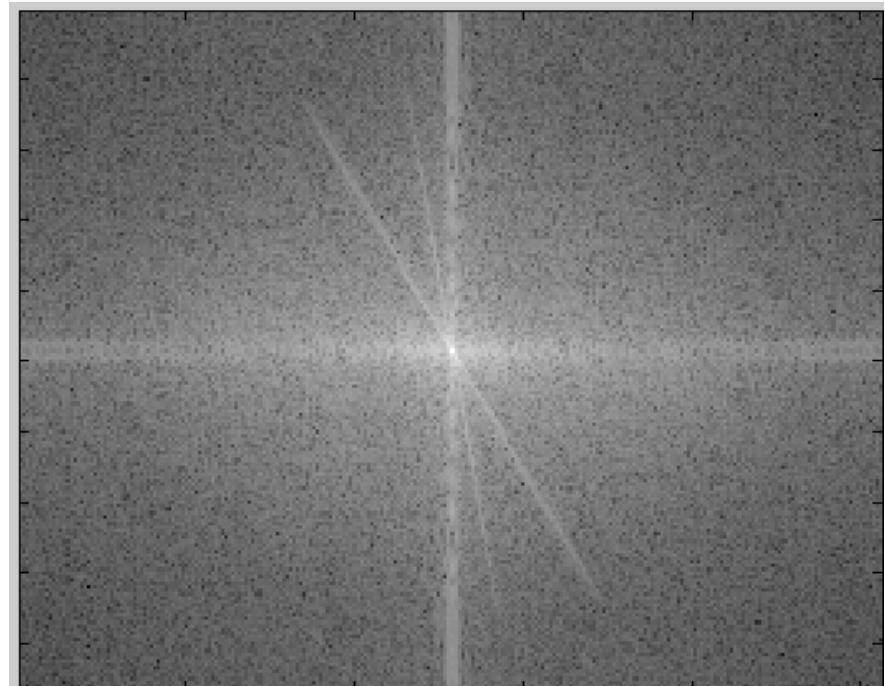
81x81



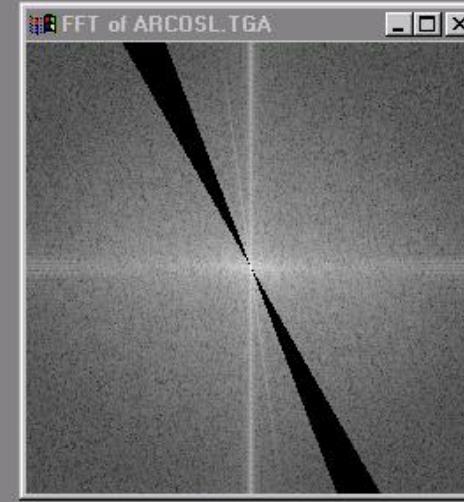
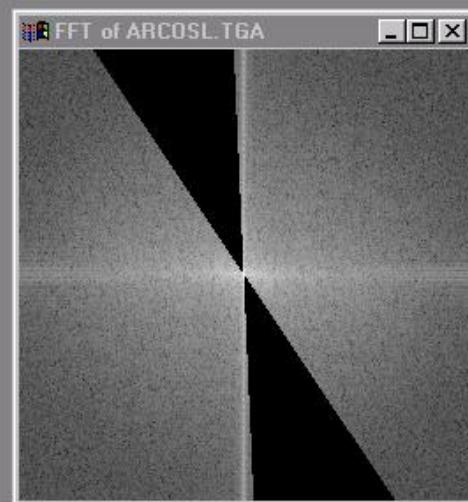
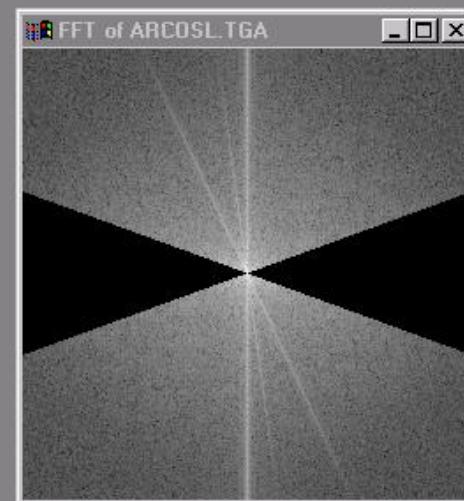
121x121



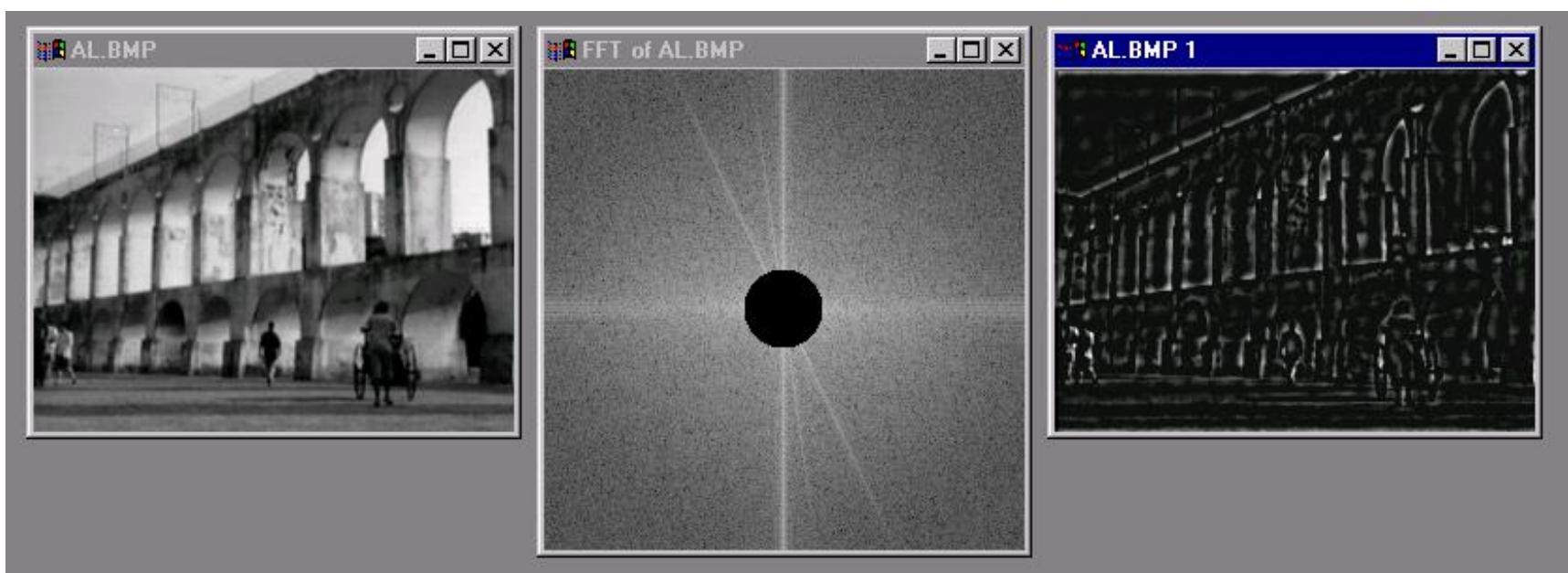
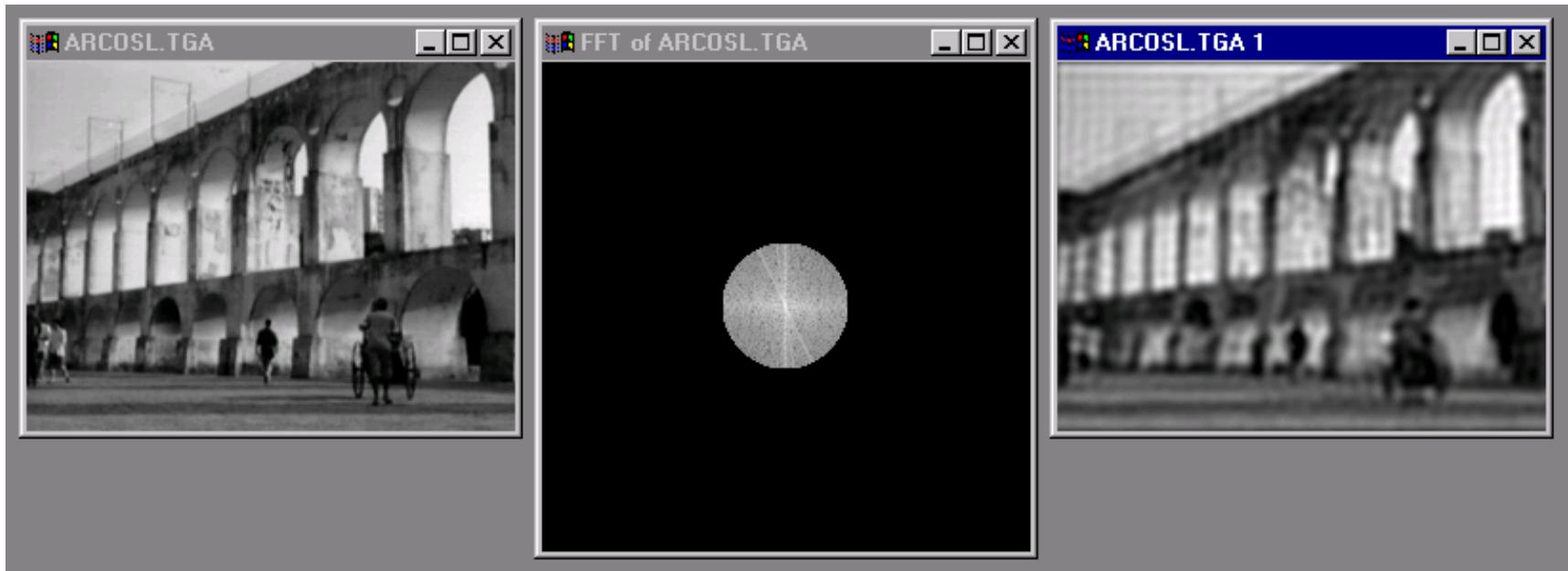
# FFT



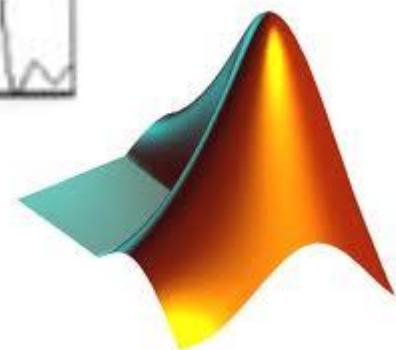
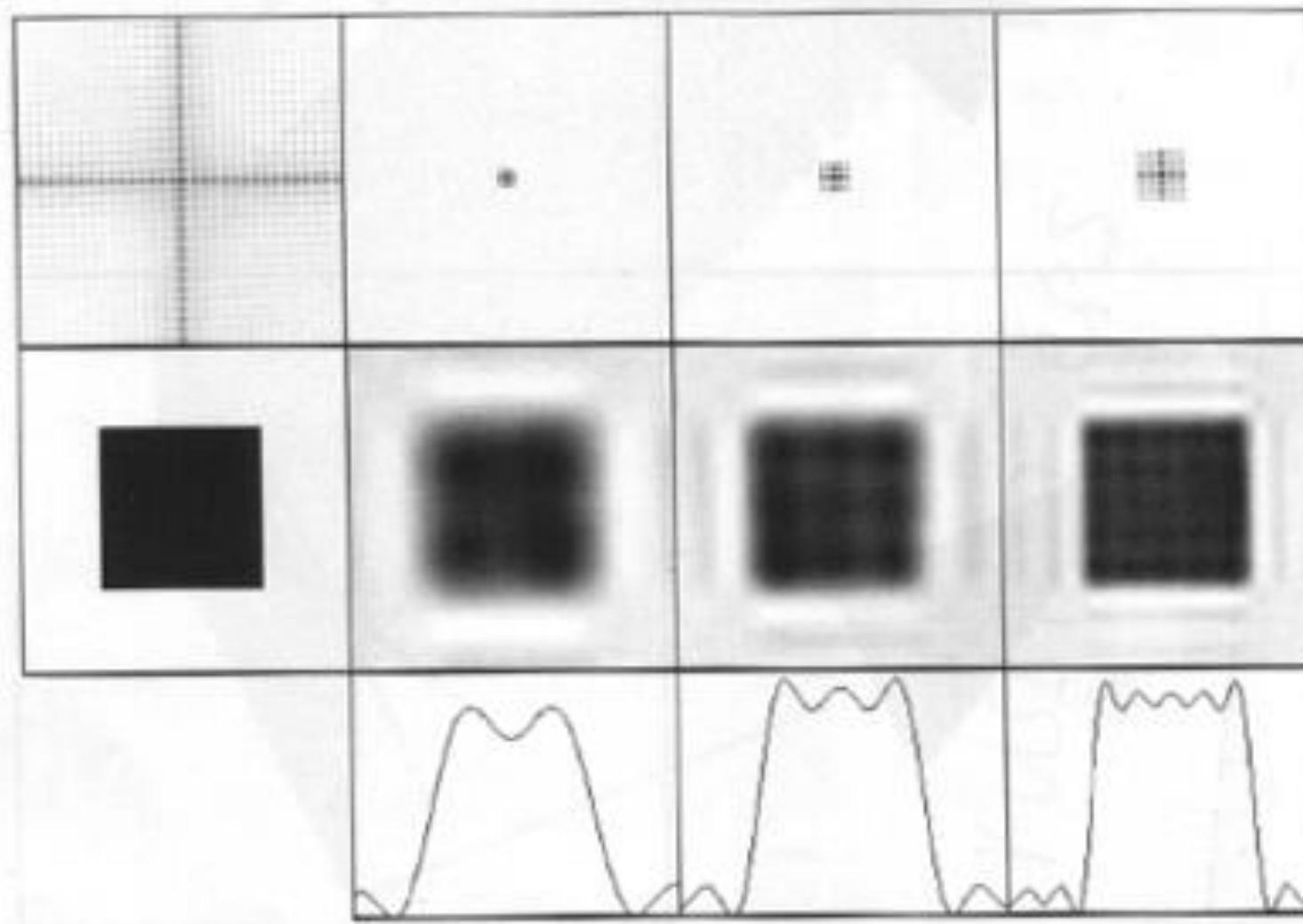
# FFT



# FFT

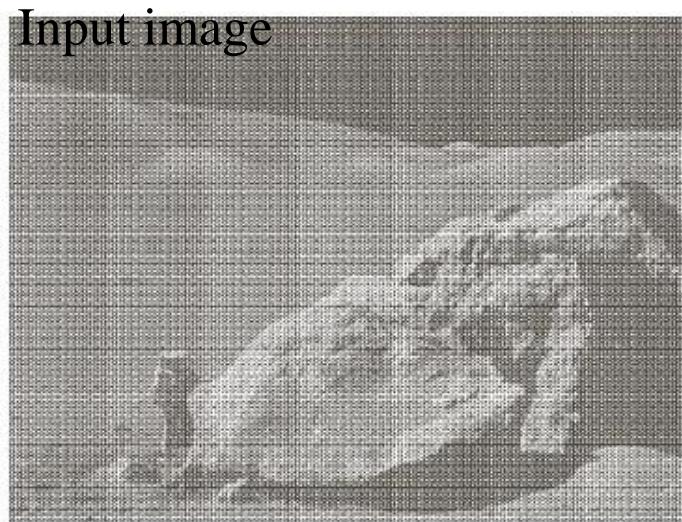


# FFT

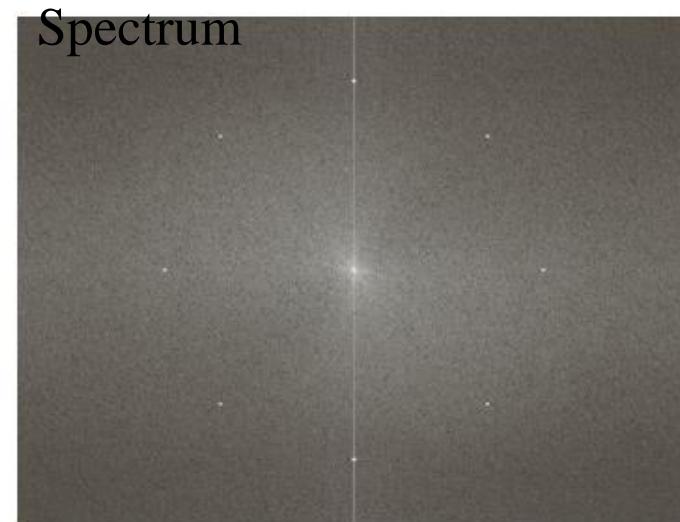


# FFT

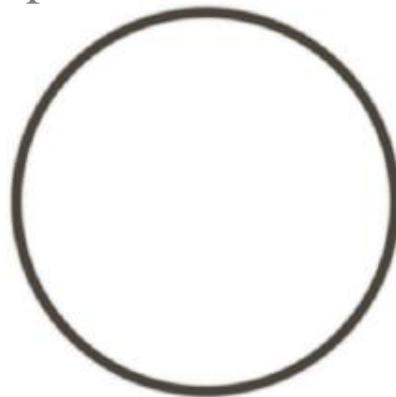
Input image



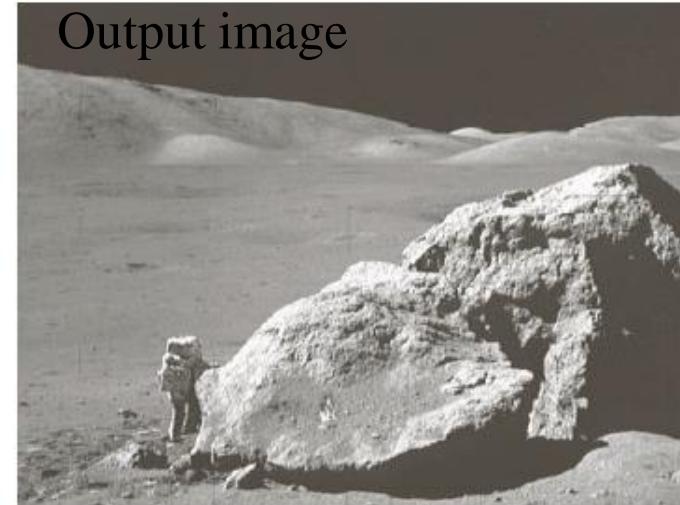
Spectrum



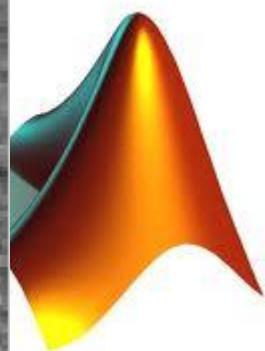
Band-pass  
filter



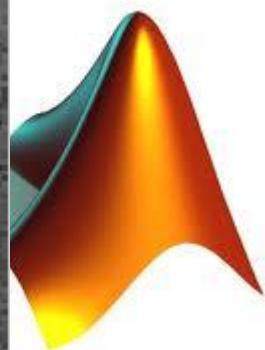
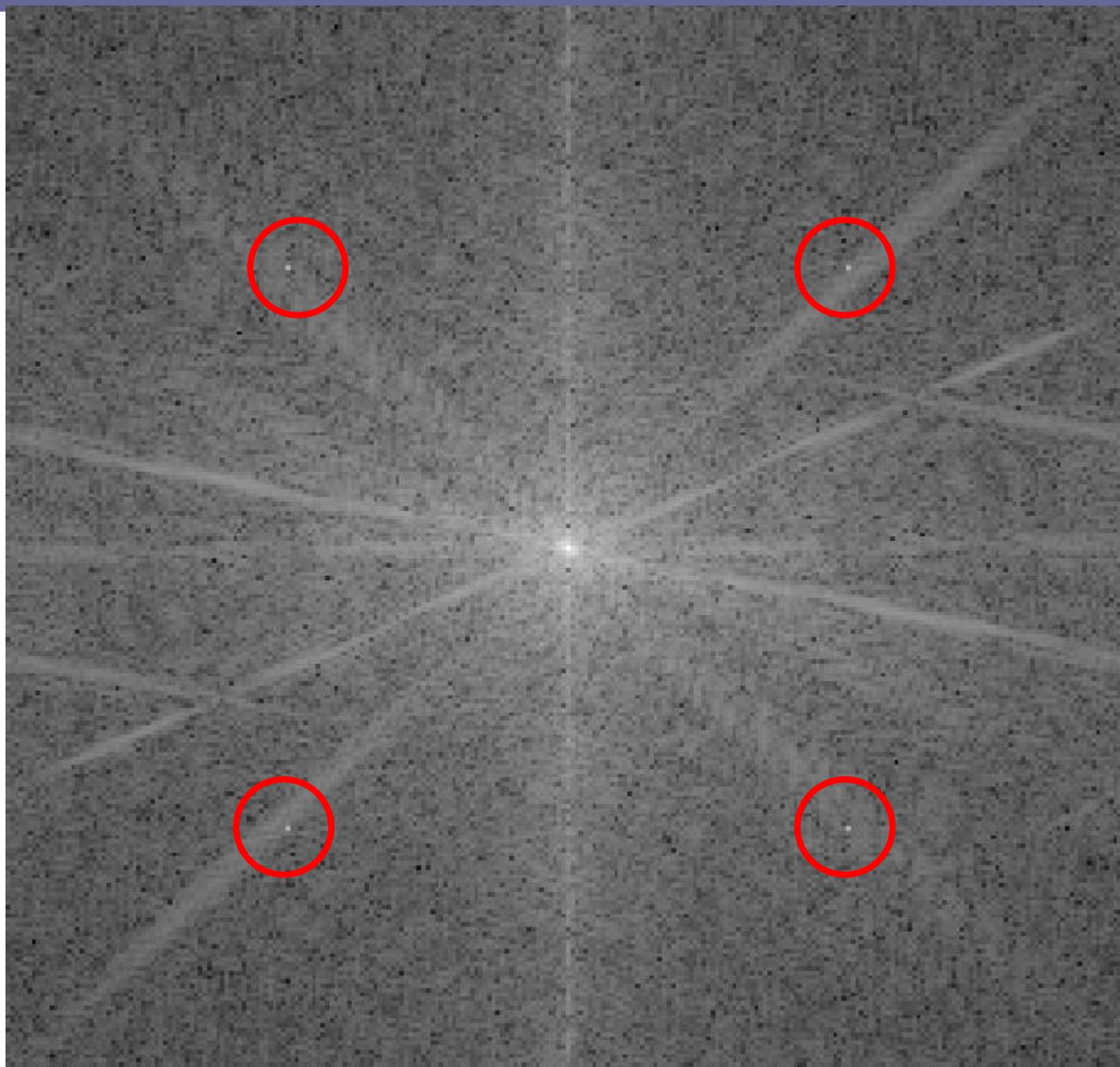
Output image



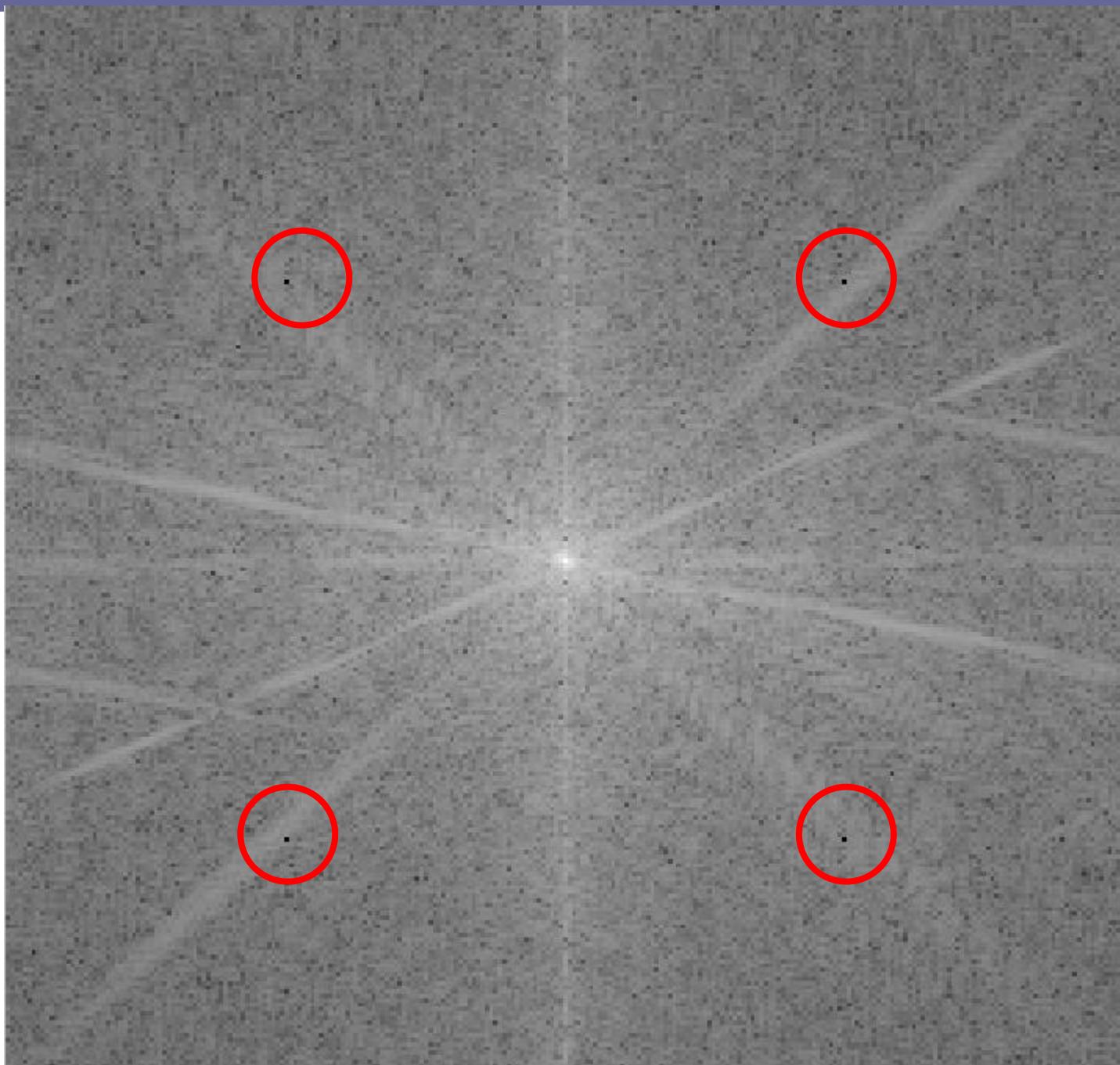
# FFT



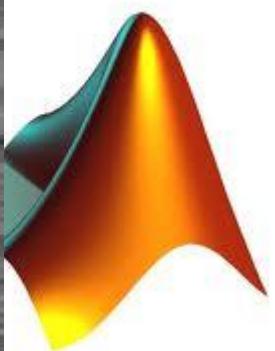
FFT



FFT



# FFT



# FFT

```
y=double(checkerboard(2,64,64)>0.5);  
  
a=double(imread('cameraman.tif'))+10*y;  
  
F = fft2(a);  
FF=fftshift(F);  
  
FF(?,?)=0;  
  
IF=fftshift(FF);  
b=ifft2(IF);  
figure, imshow(real(b),[])
```

