SIS Scuola Interdipartimentale delle **Scienze**, dell'**Ingegneria** e della **Salute**

DiST Dipartimento di Scienze e Tecnologie

L. Magistrale in IA (ML&BD)

Scientific Computing (part 2 – 6 credits)

prof. Mariarosaria Rizzardi

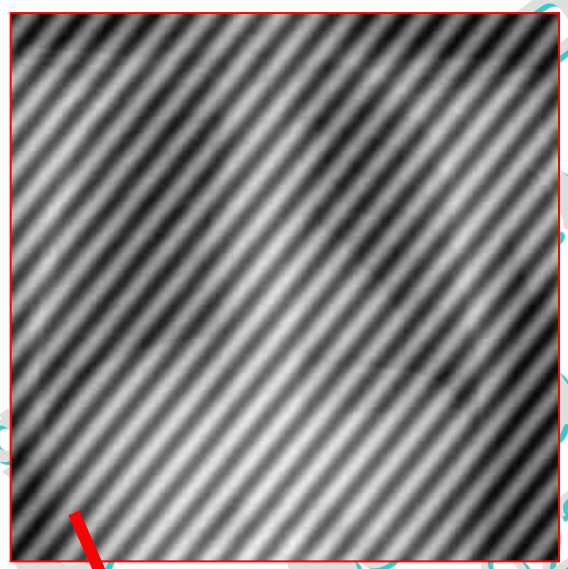
Centro Direzionale di Napoli – Bldg. C4
room: n. 423 – North Side, 4th floor
phone: 081 547 6545
email: mariarosaria.rizzardi@uniparthenope.it

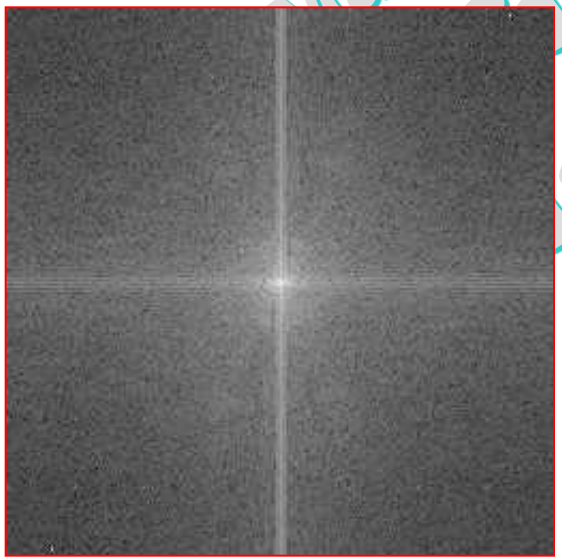
Contents

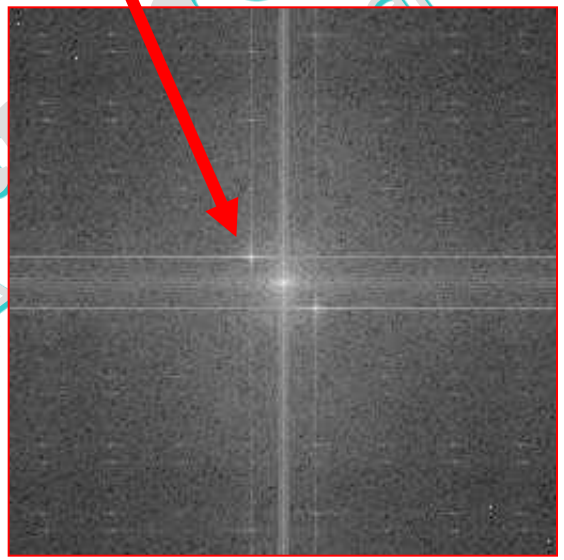
- **Other applications of 2D FT to images.**

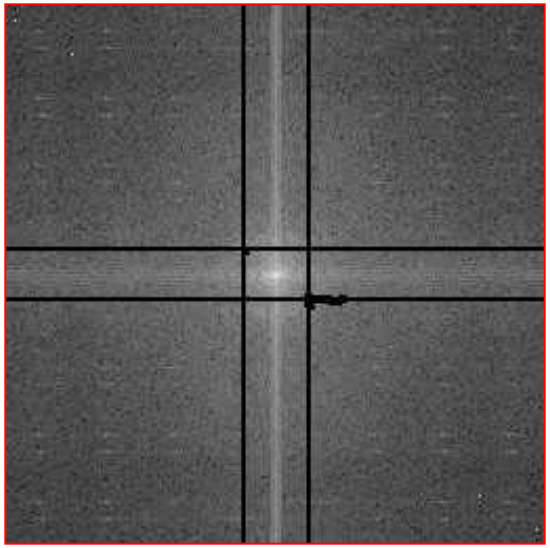
from the previous MATLAB examples

APPLICATION: remove periodic noise



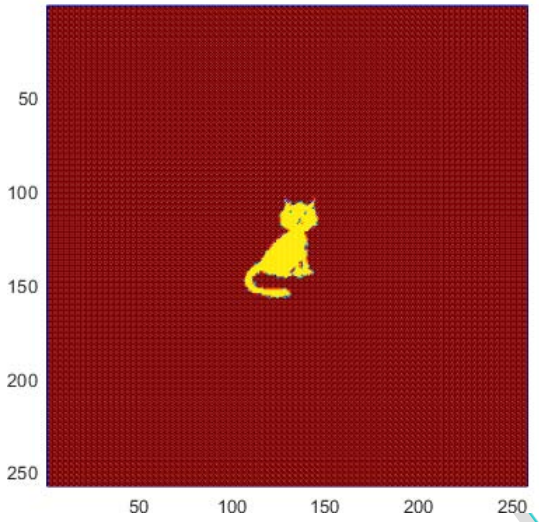




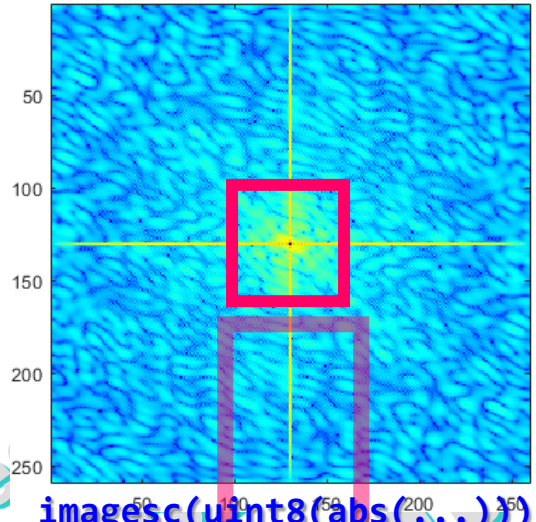


Example: windowing in frequency domain

Original image

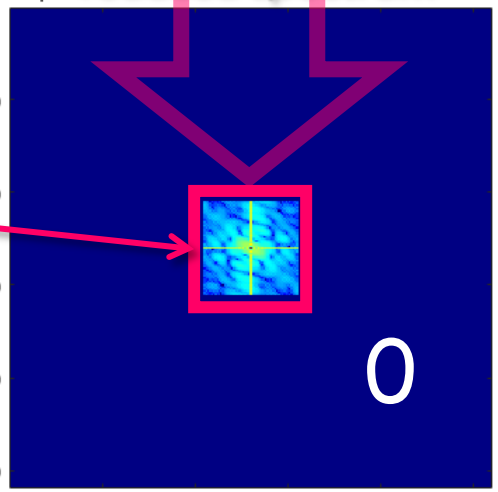


Fourier spectrum



imagesc(uint8(abs(...)))

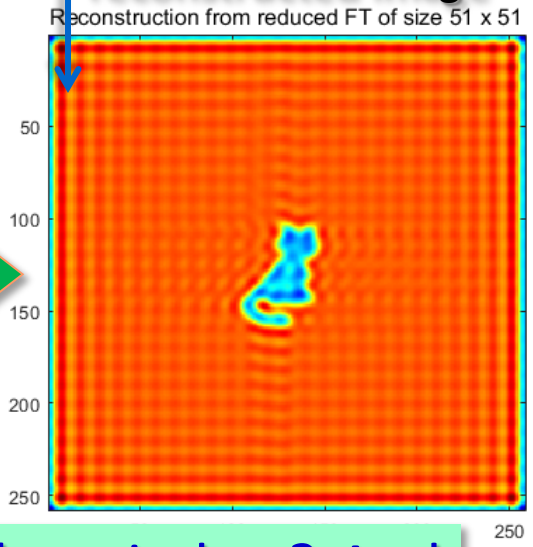
reduced spectrum



window

secondary oscillations

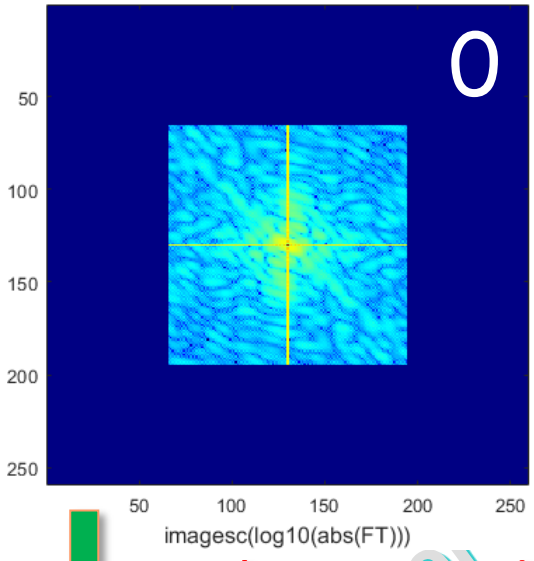
reconstructed image



Question: What do you expect with a smaller window? And with a larger one?

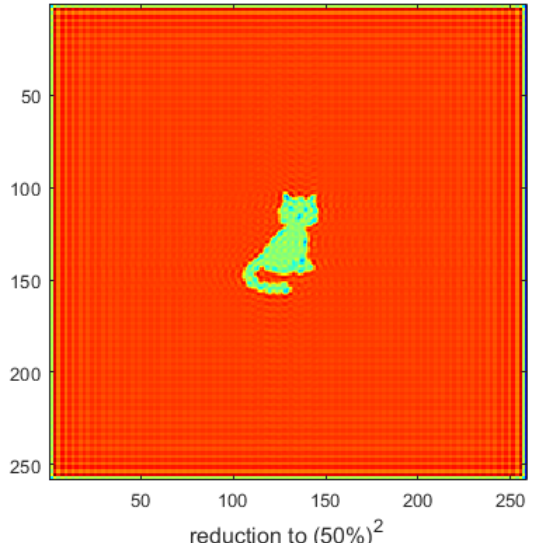
Example: windowing in frequency domain

reduced Spectrum

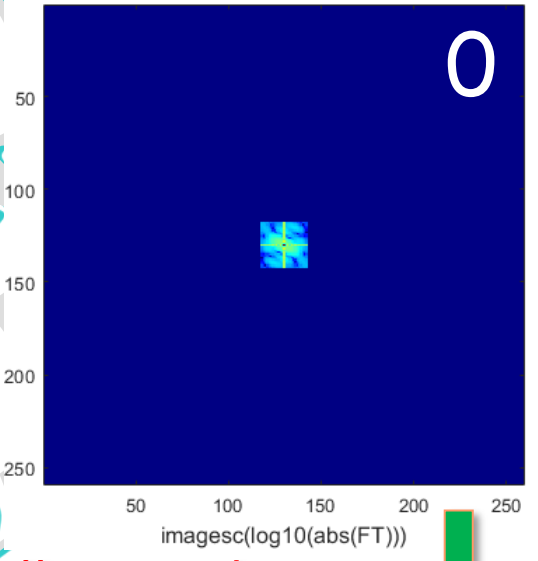


larger window

reconstructed image

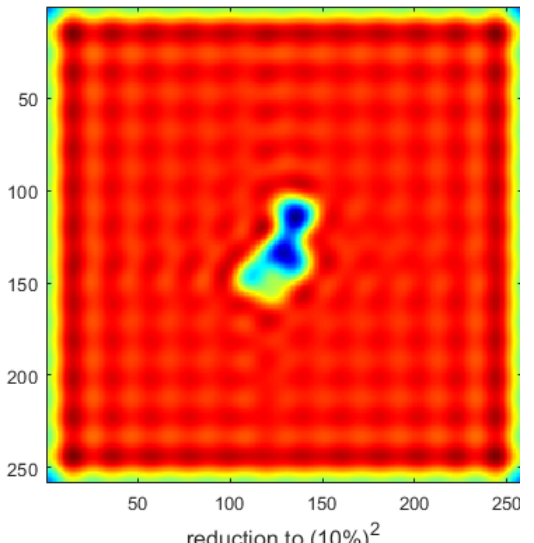


reduced Spectrum



smaller window

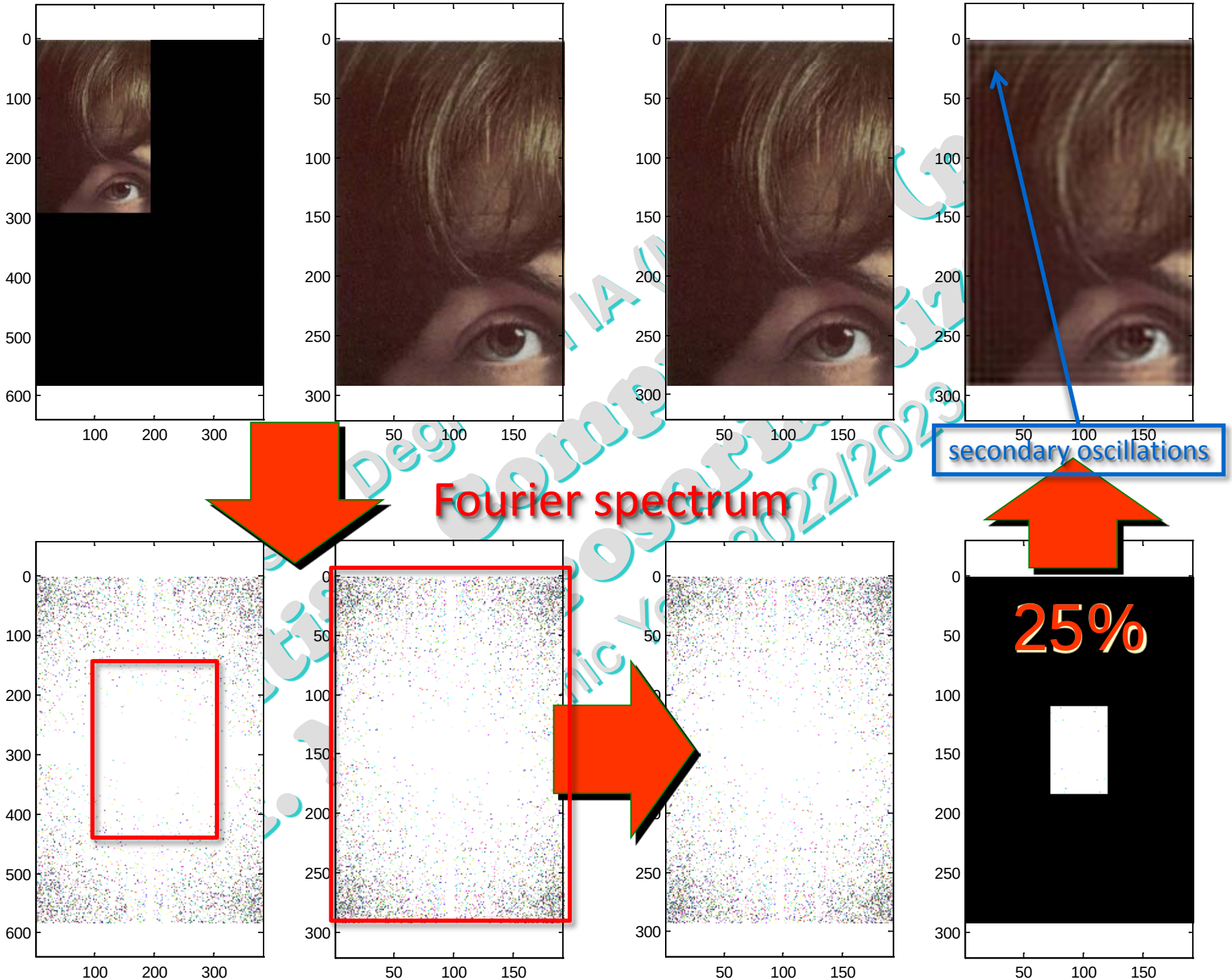
reconstructed image



Example: windowing in frequency domain

If the image is very large, you can divide it* up into "small chunks", then process on every chunk (even in parallel), and at last reconstruct it from chunks by putting together results as a mosaic.

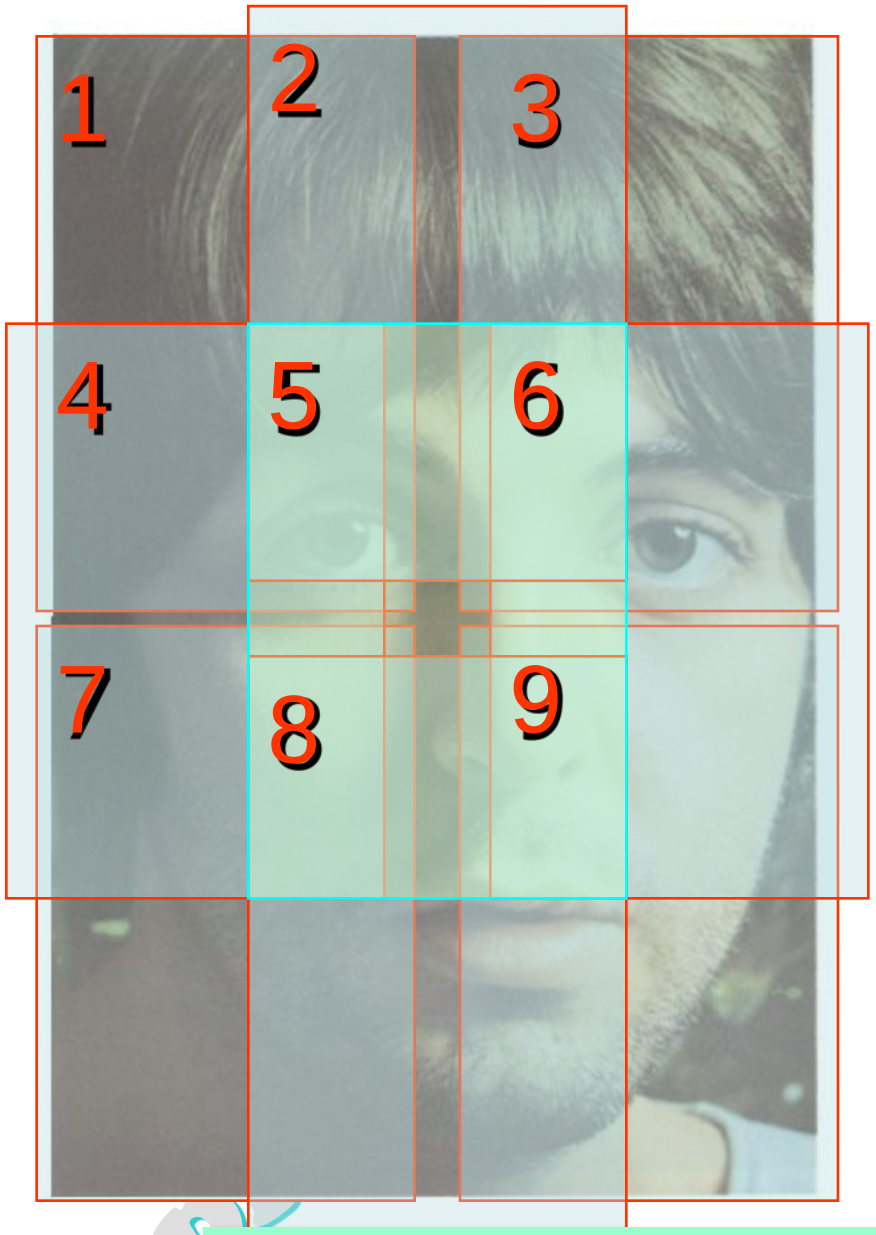
* low-pass filtering


secondary oscillations
Fourier spectrum
25%

Effects of windowing error

How can it be removed ?

Windowing error: how can it be removed ?



The image is divided into 3x3 subimages that overlap (in the example by 50%)

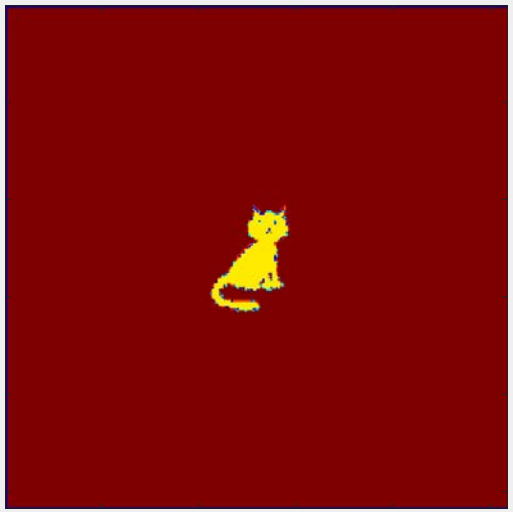
The reconstructed parts of the image after filtering can be cropped keeping only the central portion, free of errors and eliminating the border where the "windowing effect" occurs.

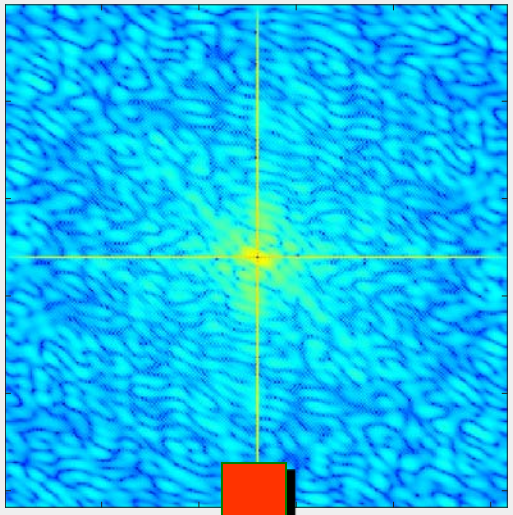
Exercise: Implement this algorithm in MATLAB.

Example: aliasing in frequency domain

```
f=imread('piccat.gif');
[m,n]=size(f);
F=fftshift(fft2(double(f)));
FF=zeros(size(F));
h=6; k=1:h:m;
FF(k,k) = F(k,k);
imagesc(log10(abs(FF)))
axis ij; axis tight; axis equal
colormap('jet')
ff=fftshift(ifft2(FF));
imagesc(real(ff)); view(2)
axis ij; axis tight; axis equal
colormap('jet')
```

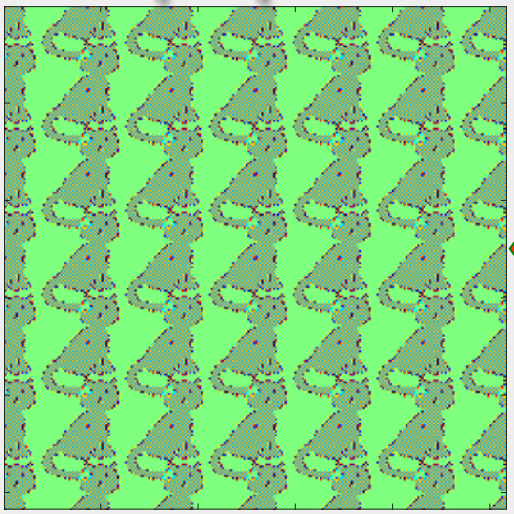
FT subsampling

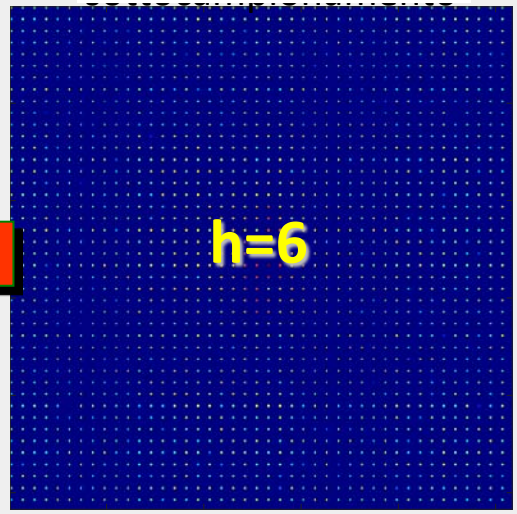




FT subsampling

superposition





Aliasing effect

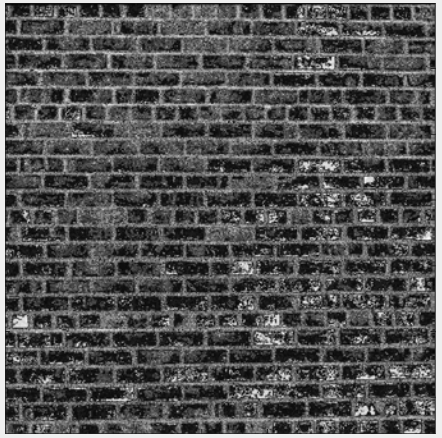
What is expected for h=3? … and for h=12?

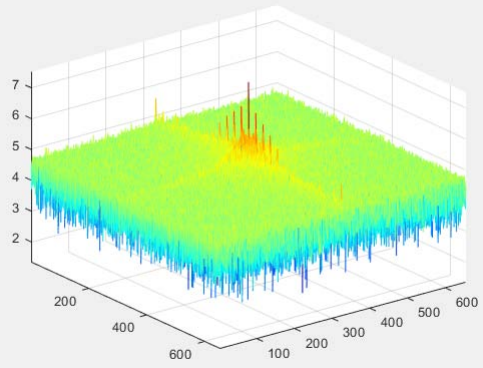
Only the values of FT with step=h in the two dimensions remain; the others are set to zeros.

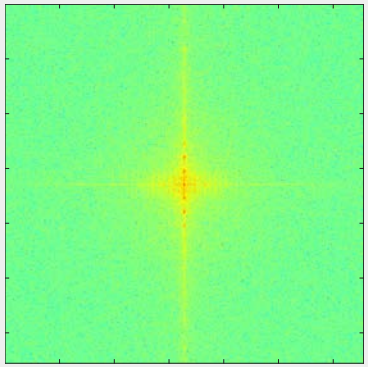
Example: periodic textures

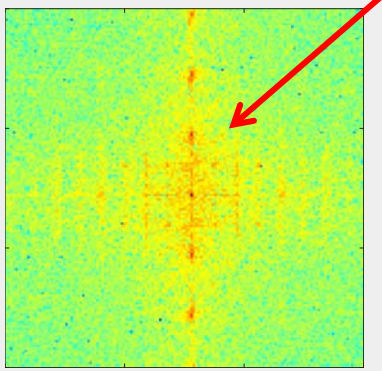
```
f=imread('../images/brick_wall.png');
figure(1); clf; imagesc(f)
axis tight; axis equal; colormap('jet')
F=fftshift(fft2(f));
figure(2); clf; mesh(log10(abs(F))); axis ij;
axis tight; colormap('jet')
figure(3); clf; imagesc(log10(abs(F)));
axis tight; axis equal; colormap('jet')
figure(2); AX=axis; % ZOOM
AX(1:4)=[250 400 250 400]; axis(AX)
figure(3); axis(AX(1:4))
```

brick_wall.png





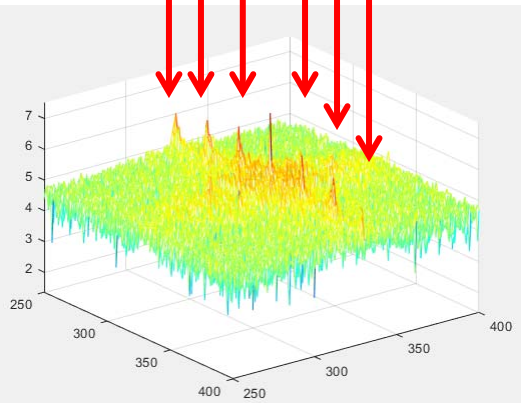




peaks

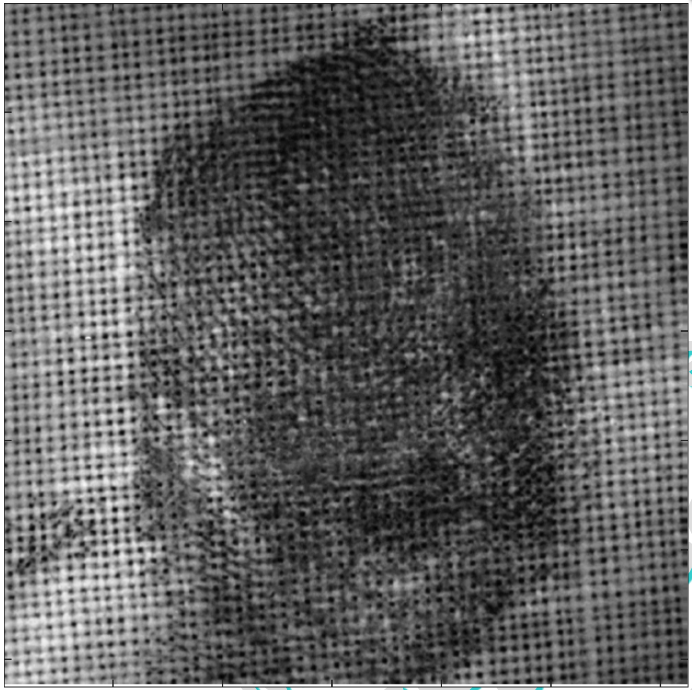
zoom

horizontal and vertical
equispace peaks

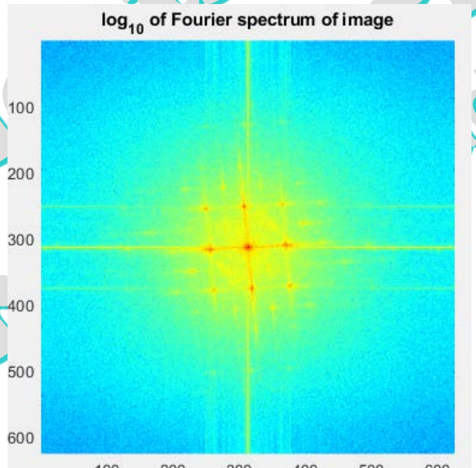


Example: application of FT to images

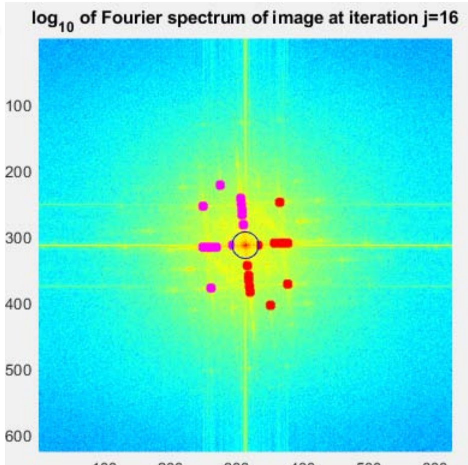
Remove a periodic background

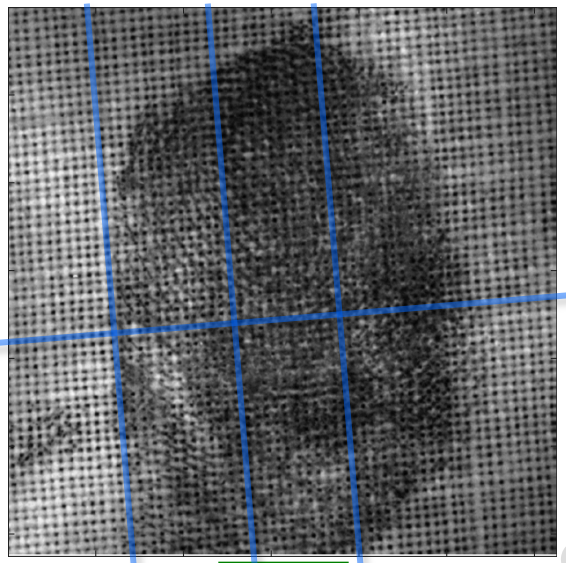


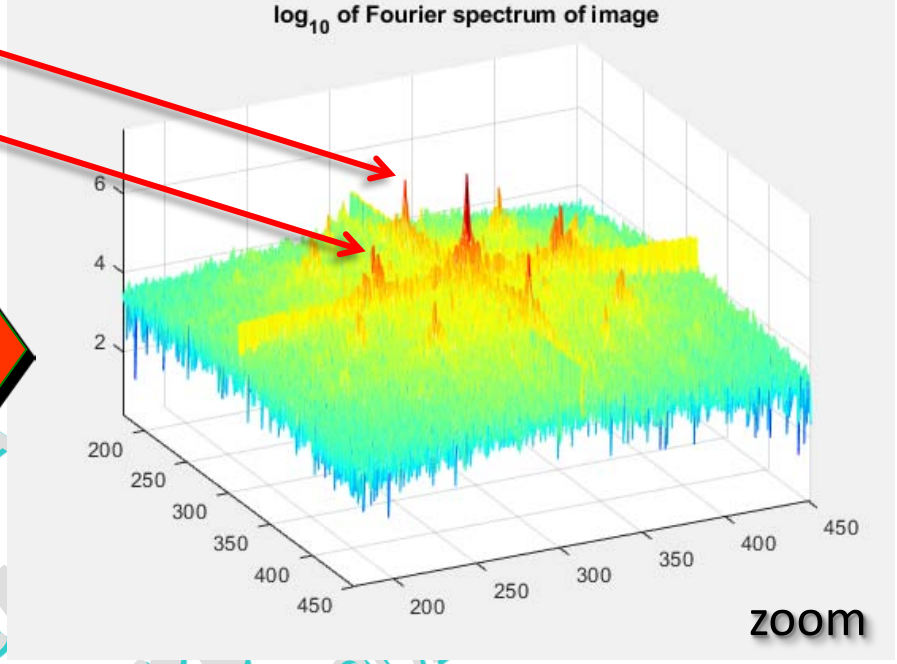
The background fabric texture interferes with the fingerprint



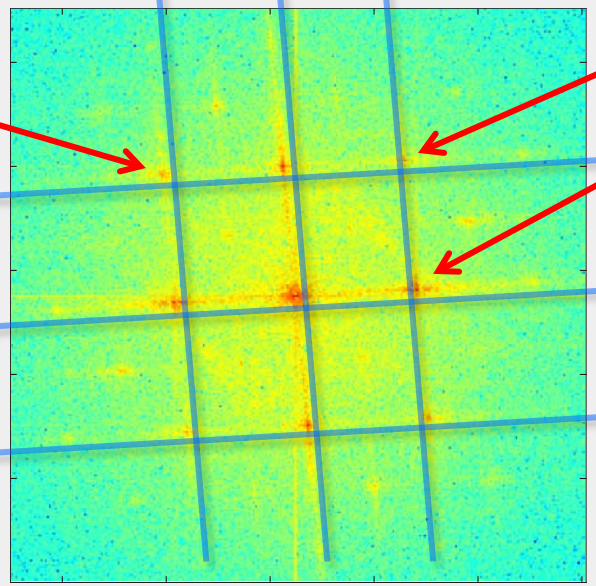
The fabric texture is now less visible, and the fingerprint stands out.





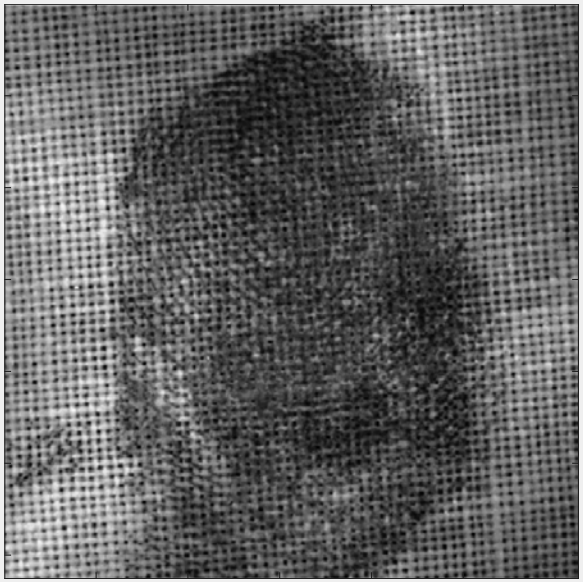


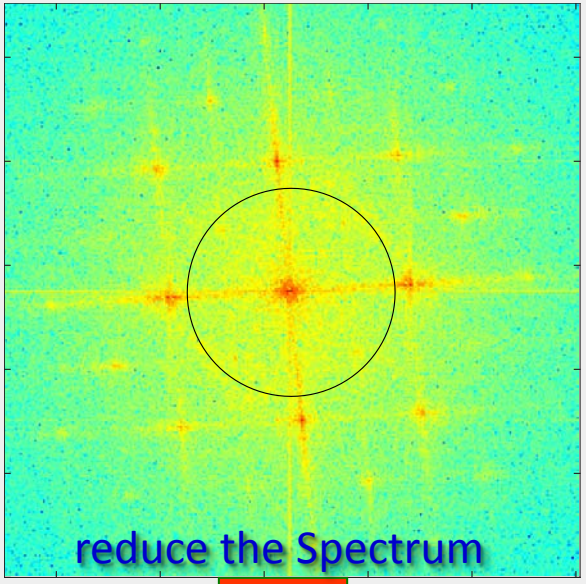
zoom



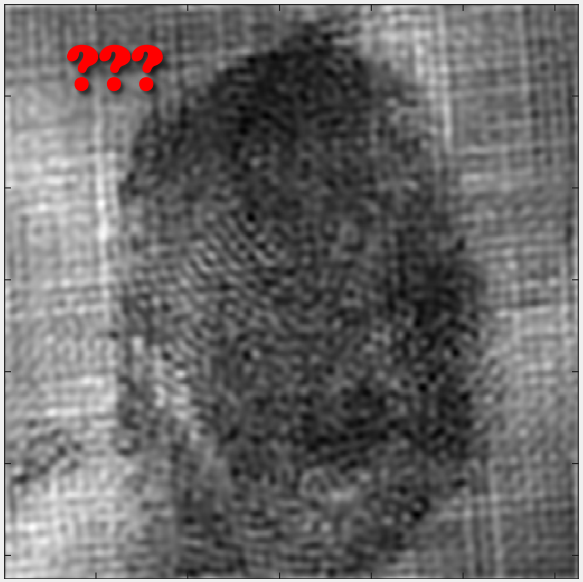
The **pulses** come from an oblique* periodic signal: in the input image the **fabric texture** under the finger-print is periodic. We need to remove these pulses.

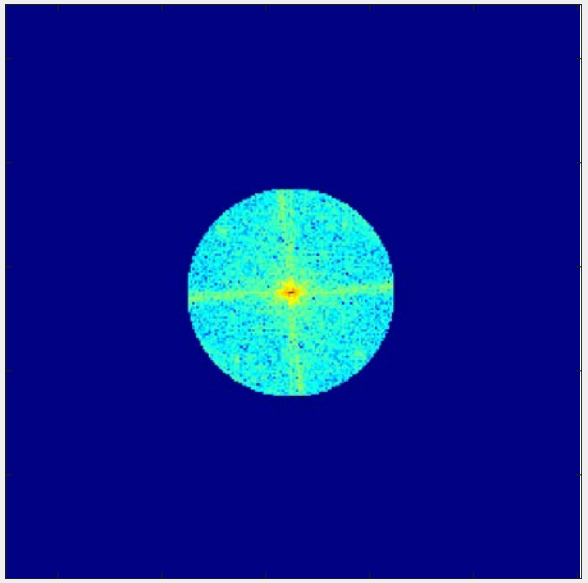
* see the rotation of the Spectrum equal to the rotation of the fabric texture.

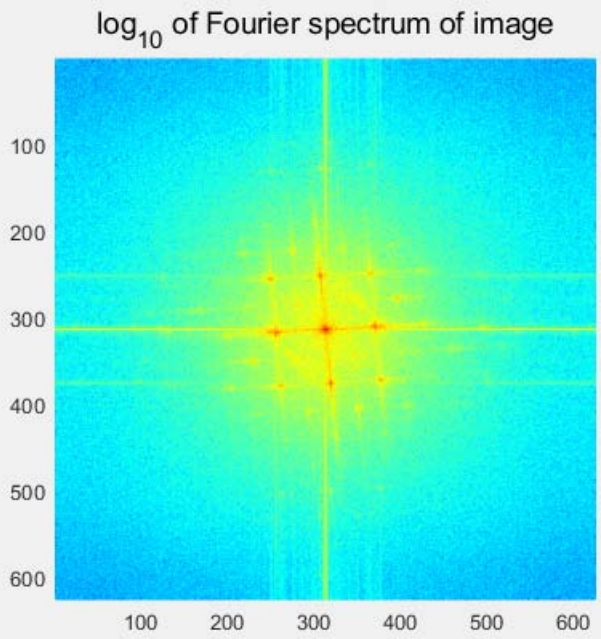


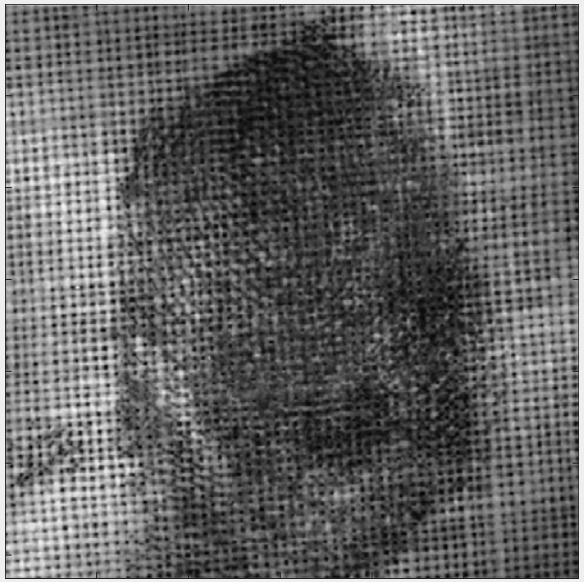


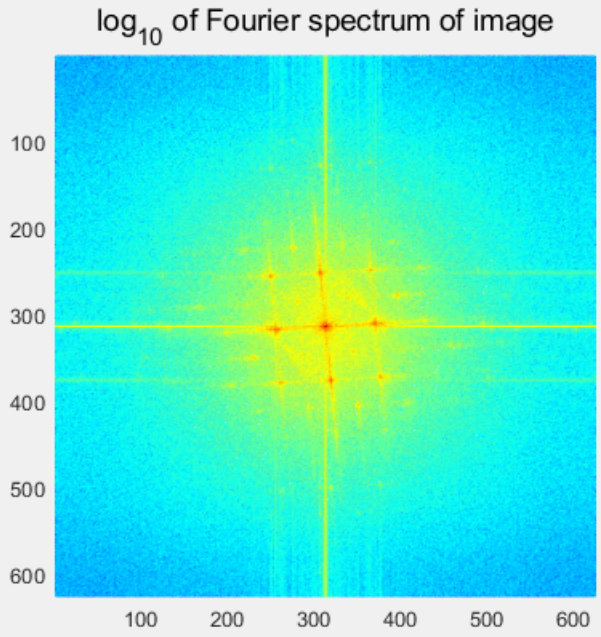
The **reduced Fourier spectrum** produces an image where both the fingerprint and the texture have lost their details

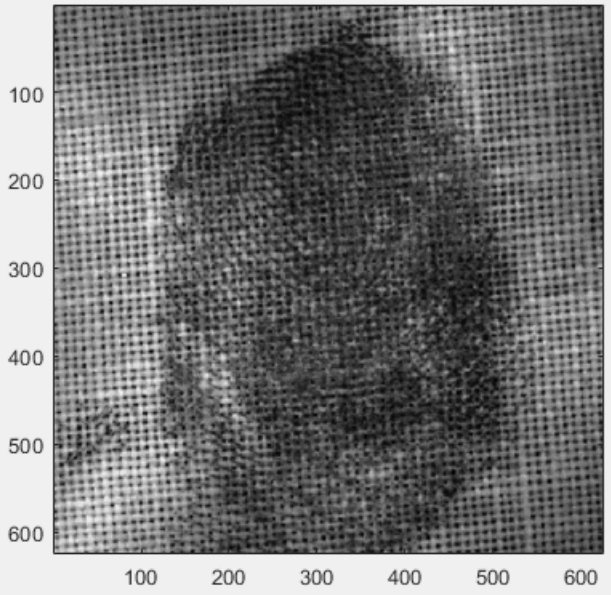




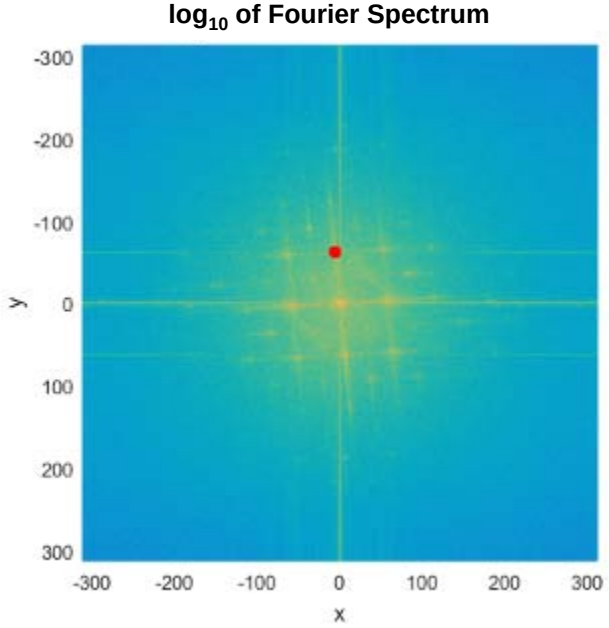


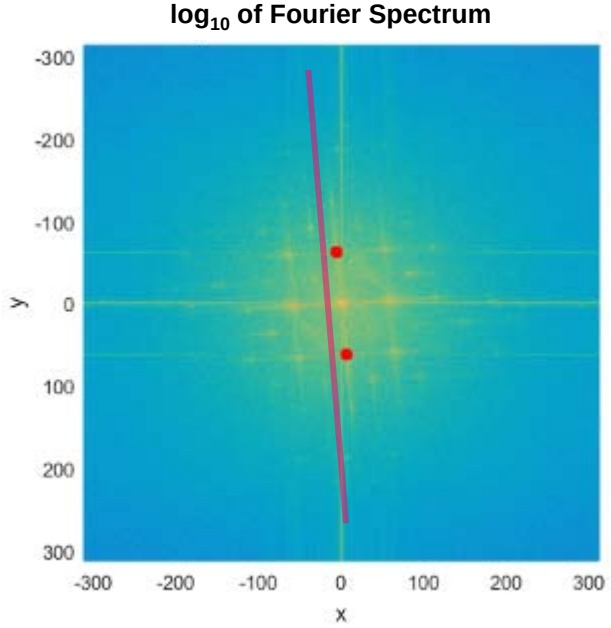






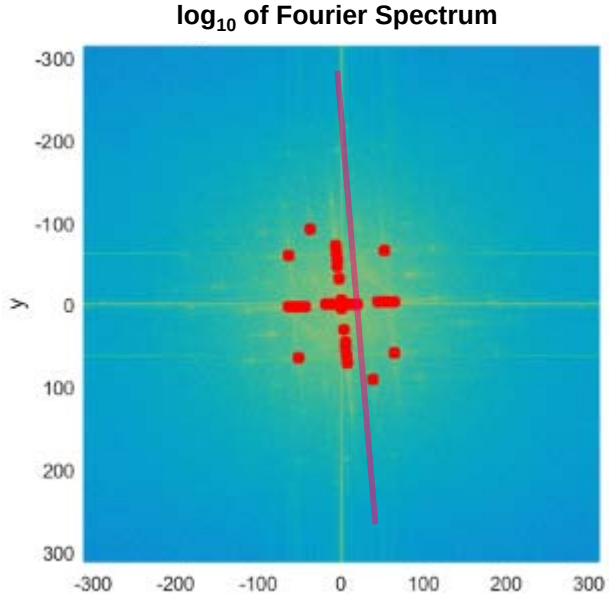
search peak points and set to zero the FT in the small disks

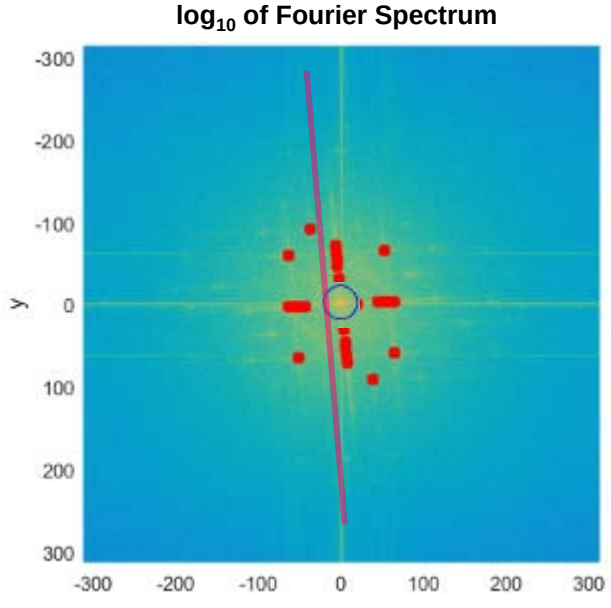




1) Remove frequencies within small circles around the pulse peak.

2) Restore the central area of the spectrum.





Exercise: Implement this algorithm in MATLAB.