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

DiST Dipartimento di Scienze e Tecnologie

# Laurea Magistrale in STN

# Applicazioni di Calcolo Scientifico e Laboratorio di ACS

## (12 cfu)


## prof. Mariarosaria Rizzardi

Centro Direzionale di Napoli – Isola C4
stanza: n. 423 – Lato Nord, 4° piano
tel.: 081 547 6545
email: mariarosaria.rizzardi@uniparthenope.it

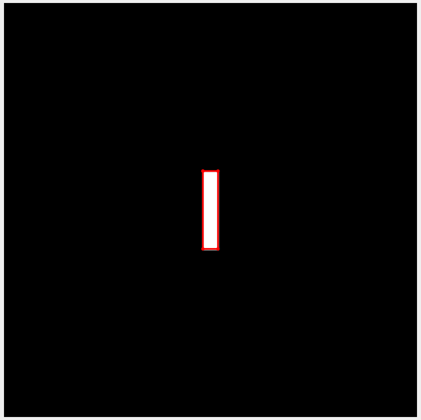
# ACS parte 2: ACS_14b

## Argomenti trattati

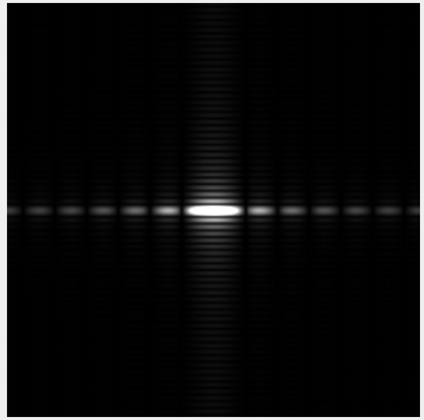
- Altre proprietà della FT 2D.
- Applicazione della FT 2D alle immagini.

# Esempi di alcune proprietà della FT 2D

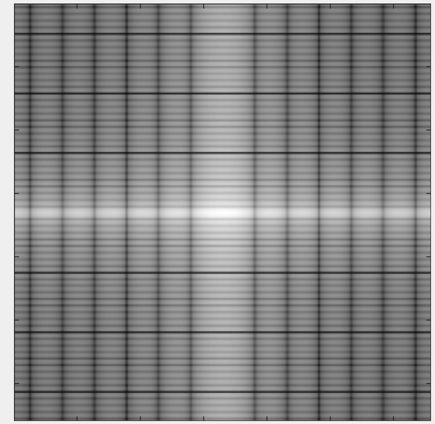
La **FT** di un'immagine è insensibile alla traslazione.



`imshow(IO)`

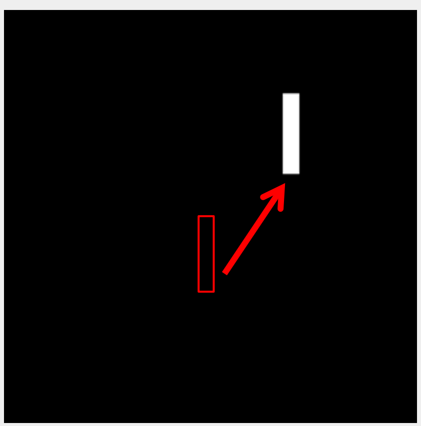


`imshow(uint8(abs(FO)))`

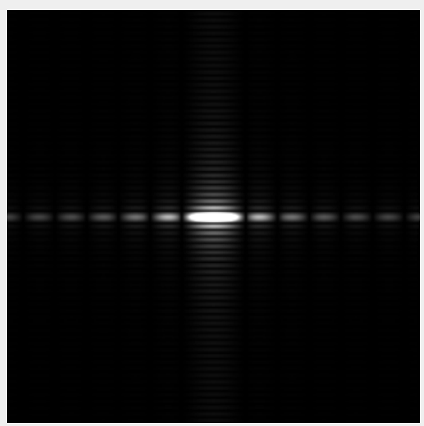


`imagesc(log10(abs(FO))); colormap(gray)`

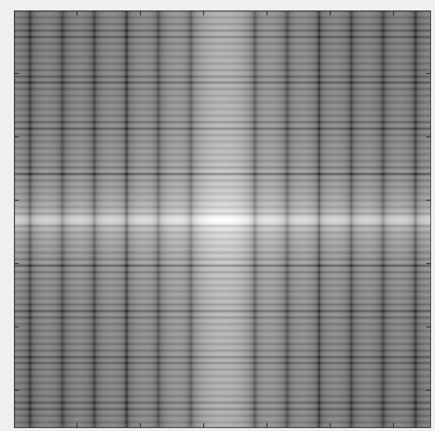
lo stesso spettro



`imshow(IT)`



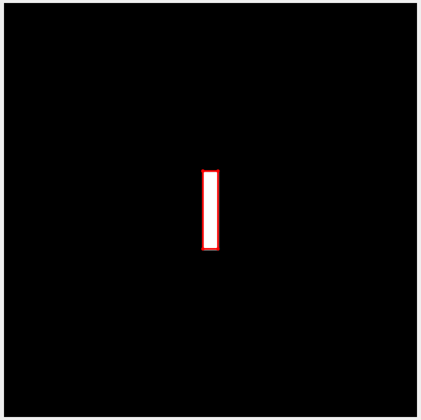
`imshow(uint8(abs(FT)))`



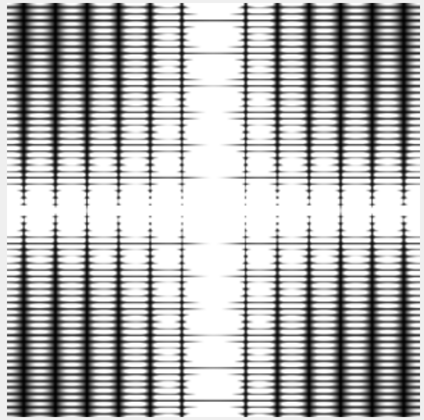
`imagesc(log10(abs(FT))); colormap(gray)`

# Esempi di alcune proprietà della FT 2D

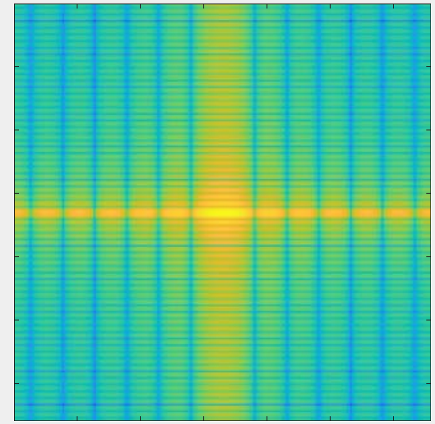
La **FT** di un'immagine è insensibile alla traslazione.



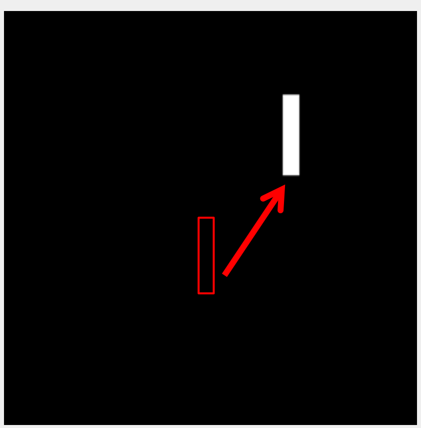
imshow(IO)



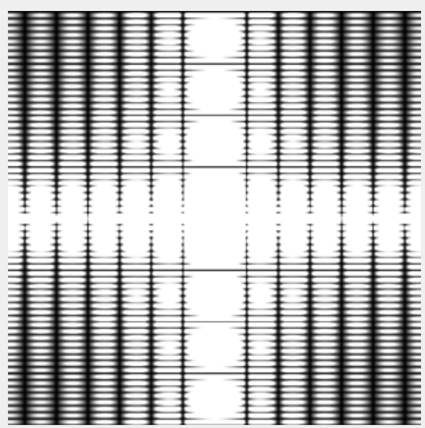
imshow(abs(FO))



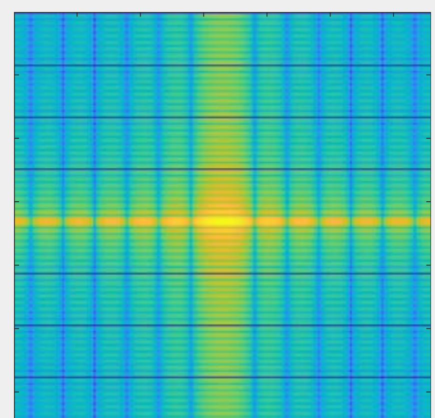
imagesc(log10(abs(FO)))



imshow(IT)



imshow(abs(FT))

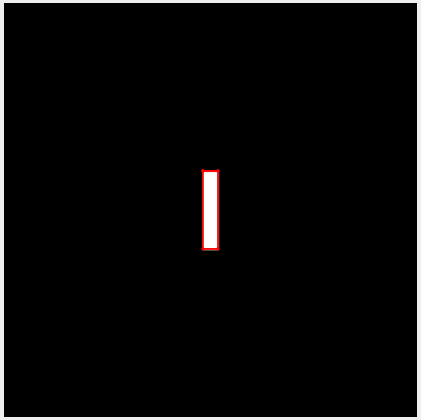


imagesc(log10(abs(FT)))

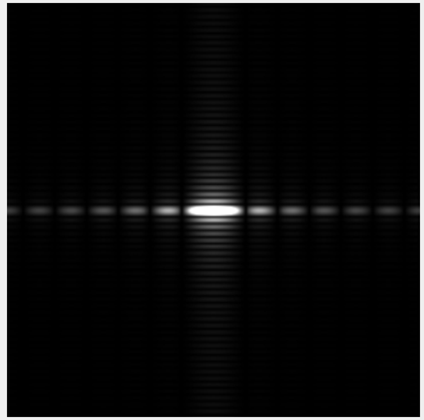
lo stesso spettro

# Esempi di alcune proprietà della FT 2D

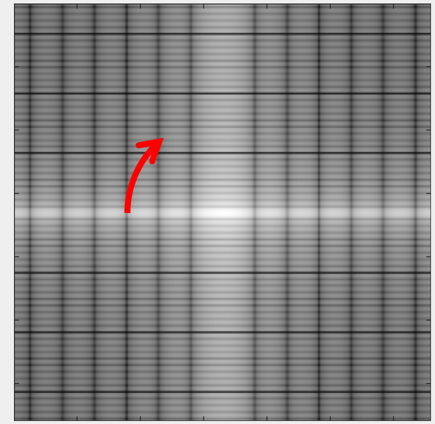
La **FT** di un'immagine è sensibile alla rotazione.



imshow(IO)

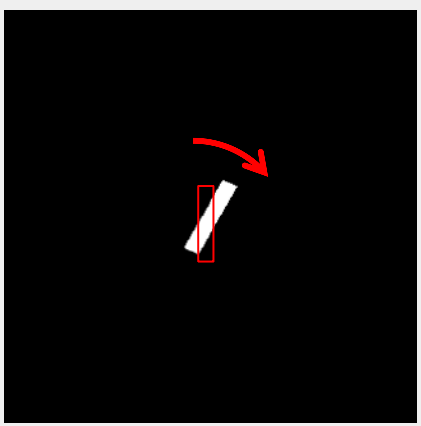


imshow(uint8(abs(FO)))

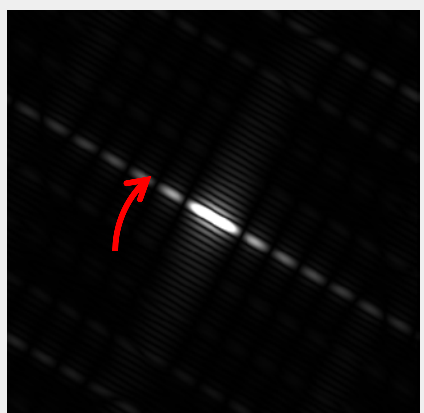


imagesc(log10(abs(FR)));
colormap(gray)

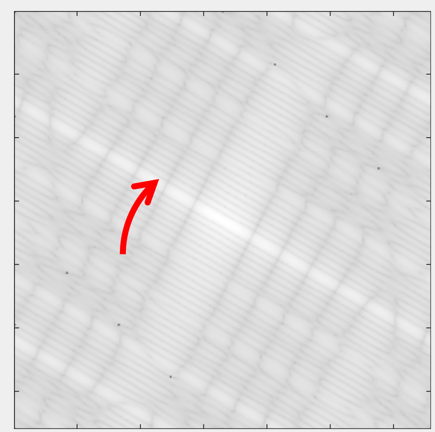
## Lo spettro ruota dello stesso angolo dell'immagine



imshow(IR)



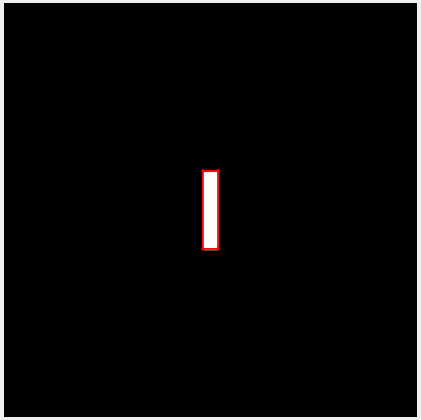
imshow(uint8(abs(FR)))



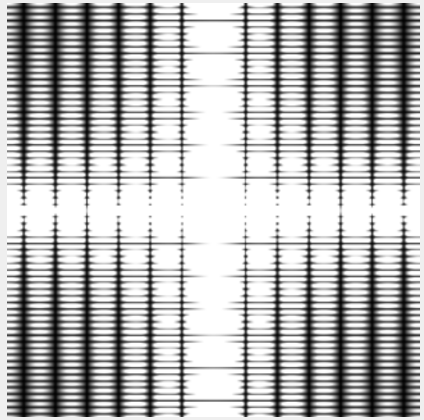
imagesc(log10(abs(FR)));
colormap(gray)

# Esempi di alcune proprietà della FT 2D

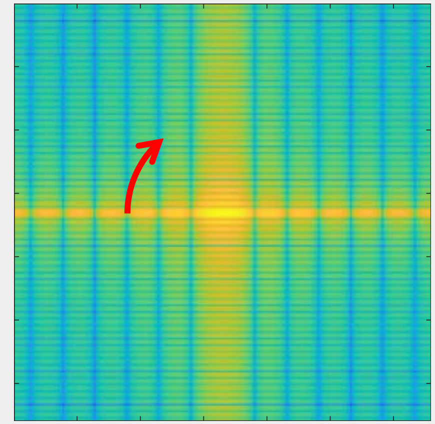
La **FT** di un'immagine è sensibile alla rotazione.



`imshow(IO)`

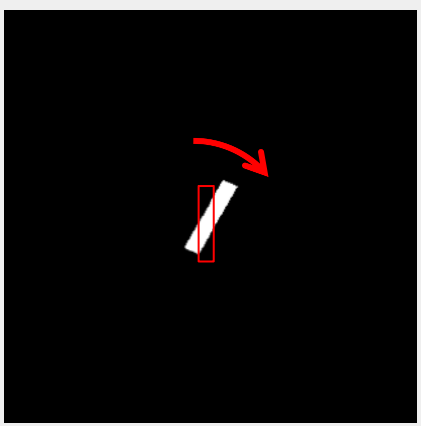


`imshow(abs(FO))`

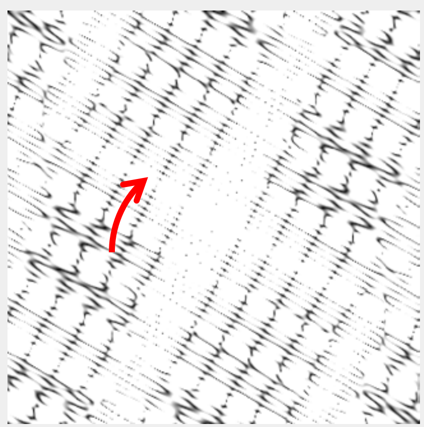


`imagesc(log10(abs(FO)))`

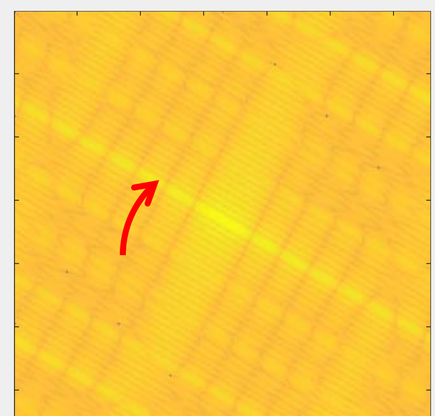
## Lo spettro ruota dello stesso angolo dell'immagine



`imshow(IR)`



`imshow(abs(FR))`



`imagesc(log10(abs(FR)))`

# **Esempio:** quale parte dello Spettro di Fourier è fondamentale?

```
f=imread('./Fourier.jpg'); [m,n]=size(f);
if rem(m,2) == 0,   f=[f;zeros(1,size(f,2))];
else                f(end,:)=zeros(1,size(f,2)); m=m-1;
end
if rem(n,2) == 0,   f=[f zeros(size(f,1),1)];
else                f(:,end)=zeros(size(f,1),1); n=n-1;
end
figure; imagesc(f); axis equal; colormap(gray); axis tight
[h,k]=meshgrid(0:n-1, 0:m-1); F=fftshift(fft2(f,m,n)).*(-1).^(h+k);
F=[F;F(1,:)]; F=[F F(:,1)];
figure; imagesc(log10(abs(F))); colormap('jet'); axis equal; axis tight
mMid=m/2+1; nMid=n/2+1;
perc=0.20; Hm=fix(m/2*perc); Hn=fix(n/2*perc);
I=mMid-Hm : mMid+Hm; J=nMid-Hn : nMid+Hn; FF=zeros(size(F)); FF(I,J)=F(I,J);
ff=fftshift(ifft2(FF,m,n)).*(-1).^(h+k); ff=[ff;ff(1,:)]; ff=[ff ff(:,1)];
figure; imagesc(real(ff)); colormap(gray); axis equal; axis tight
xlabel(['reduction to (' num2str(100*perc) '%)^2'])
title(['Reconstruction from reduced FT of size ' num2str(2*Hm+1) ' x ' num2str(2*Hn+1)])
```

rende l'input periodico e m, n pari

immagine originale con assi

algoritmo non ottimale

plot dello spettro

FT ridotta

algoritmo non ottimale

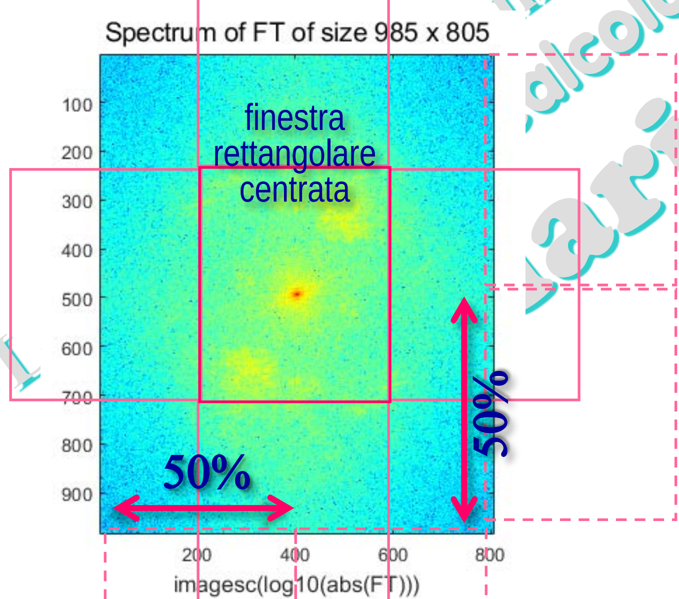
100%

# Esempio: quale parte dello Spettro di Fourier è fondamentale?


```
mMid=m/2+1; nMid=n/2+1;
perc=0.20;
Hm=fix(m/2*perc); Hn=fix(n/2*perc);
I=mMid-Hm : mMid+Hm;
J=nMid-Hn : nMid+Hn;
FF=zeros(size(F)); FF(I,J)=F(I,J);
```

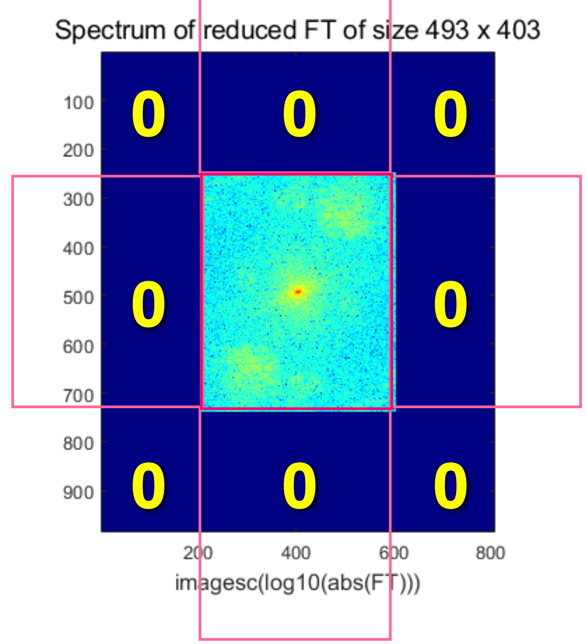
FT ridotta

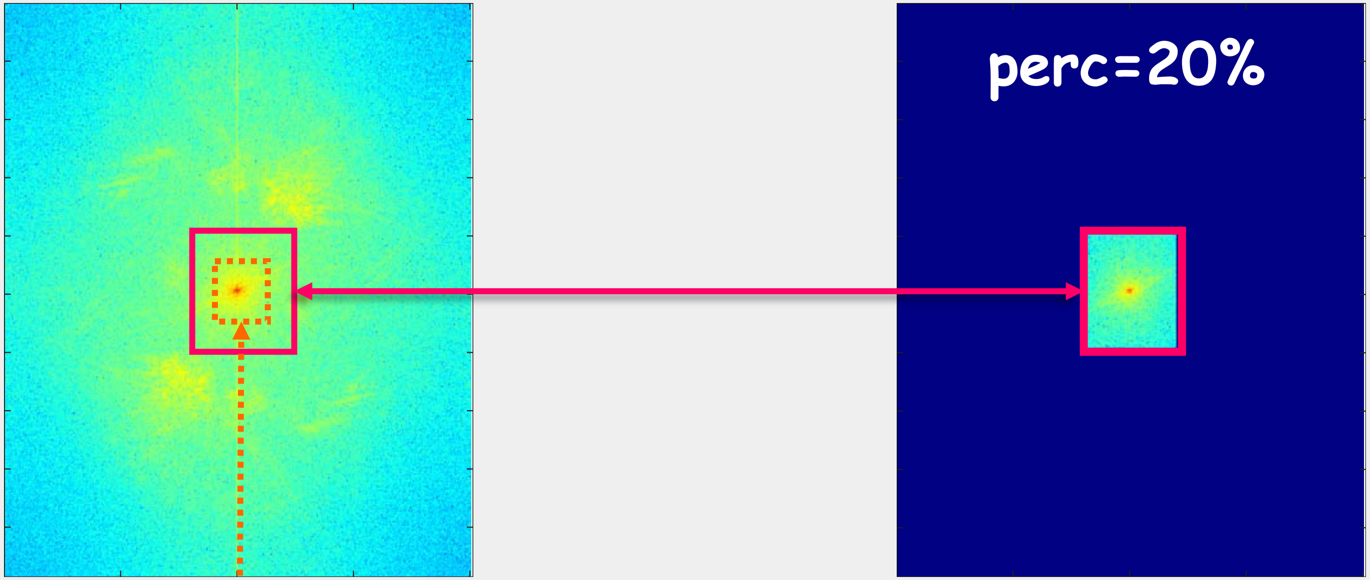

**Cos'è perc?**

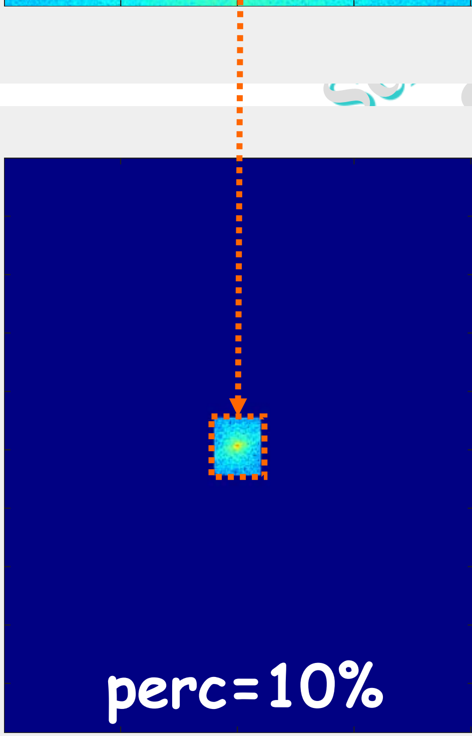



**perc=0.50**

corrisponde al **25%** della parte centrale dello Spettro



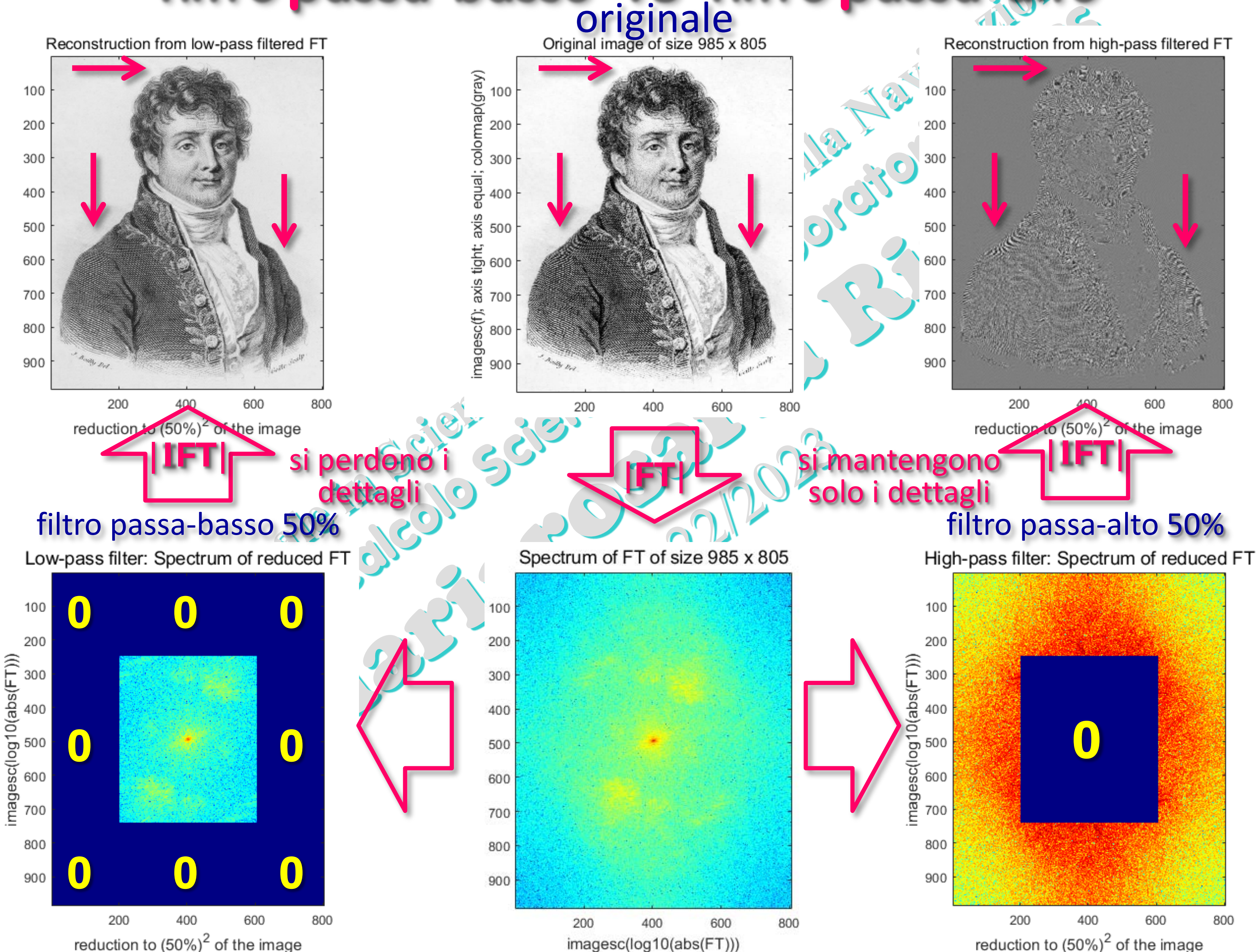



# La parte fondamentale dello Spettro di Fourier è quella centrale!

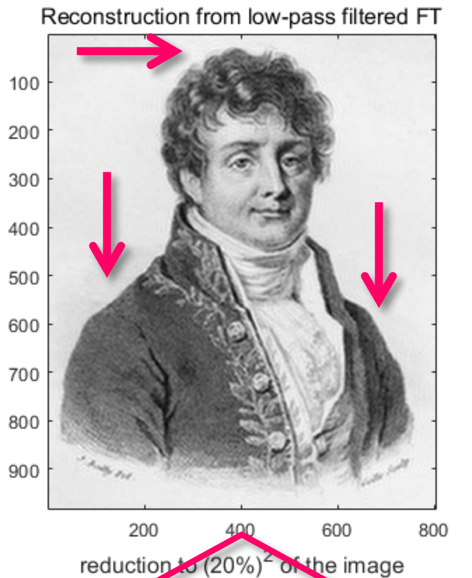
Ma, per ricostruire l'immagine, è necessaria l'intera grande matrice, poiché la FT ridotta deve essere posizionata al centro.

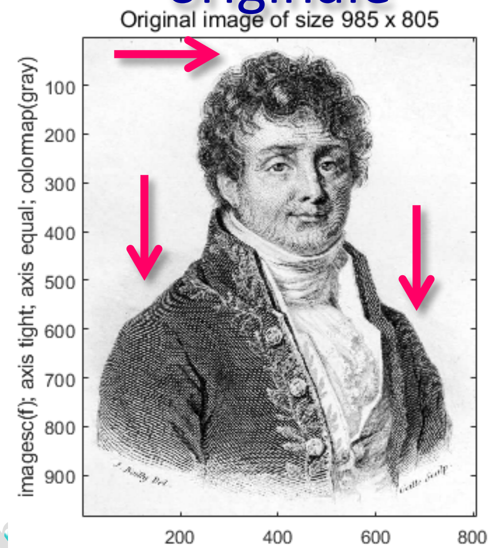
# filtro passa-basso VS filtro passa-alto

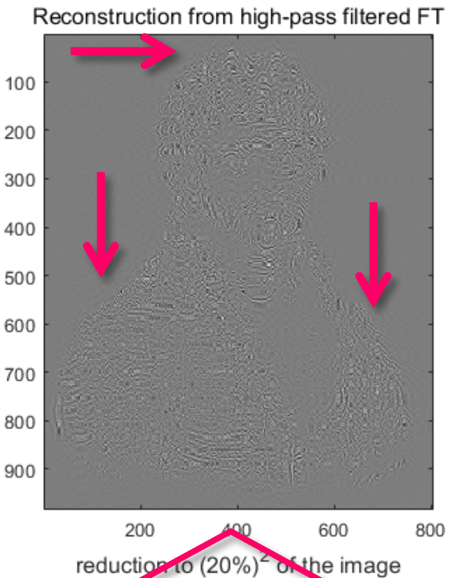


# filtro passa-basso VS filtro passa-alto

originale




|IFT|

si perdono i dettagli

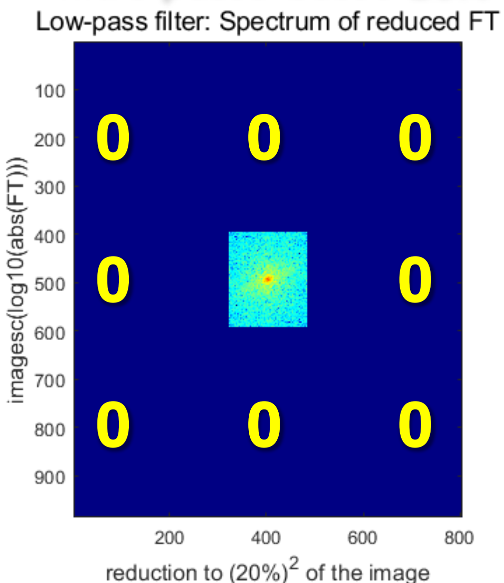
|FT|

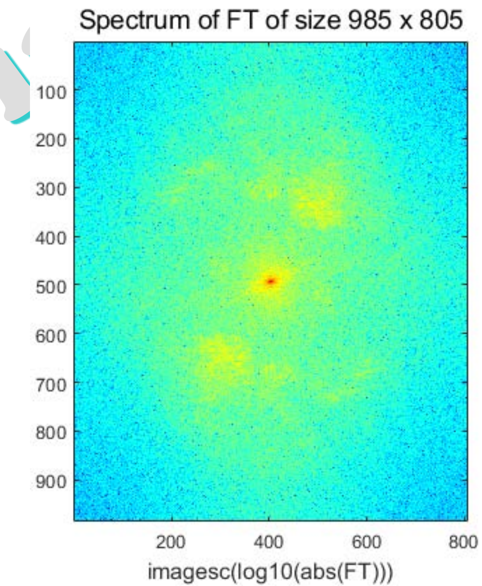
si mantengono solo i dettagli

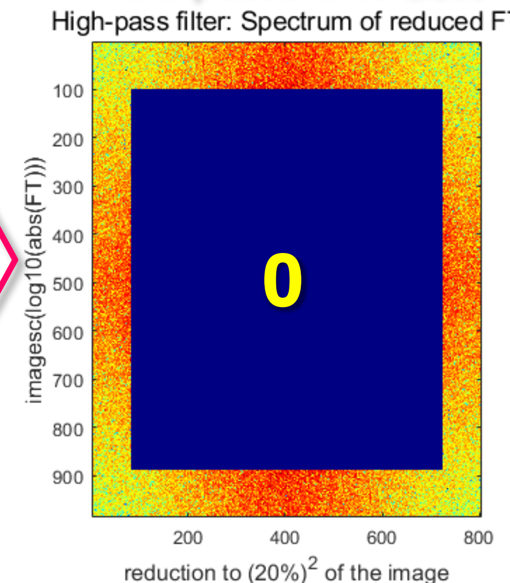
|IFT|

filtro passa-basso 20%

filtro passa-alto 20%



# Esempio: compressione di un'immagine

```
f= ... ;   F= ... ; nMid= ... ; mMid= ... ; perc=0.20;
Hn= ... ; Hm= ... ; i= ... ; j= ... ; FF=zeros(size(F));
FF=F(i,j); ff=fftshift(ifft2(FF,m,n)).*(-1).^(h+k);
figure; imagesc(real(f)); axis equal; colormap(gray); axis tight
...
figure; imagesc(real(ff)); axis equal; colormap(gray) ; axis tight
...
C=dct2(f);  CC=C(1:2*Hm+1,1:2*Hn+1);
cc=idct2(CC,m,n);
figure; imagesc(cc); axis equal; colormap(gray); ...
```

zero padding

2D DCT e DCT ridotta

ricostruzione dell'imagine dalla DCT ridotta

**DCT = Discrete Cosine Transform**

**usata nella compressione jpeg**

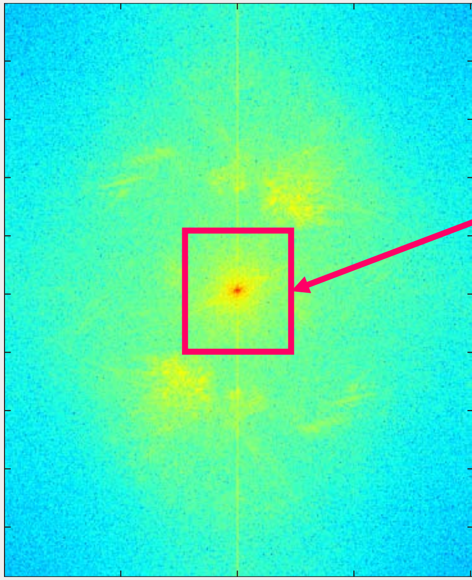
(compressione di dati con perdita di informazioni)

in MATLAB per la DCT 2D:

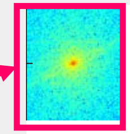
**dct2()** e **idct2()**

100%

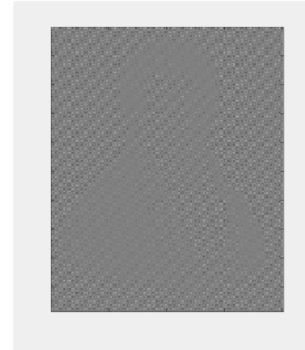
## Perché non c'è compressione?



Fourier Trasform



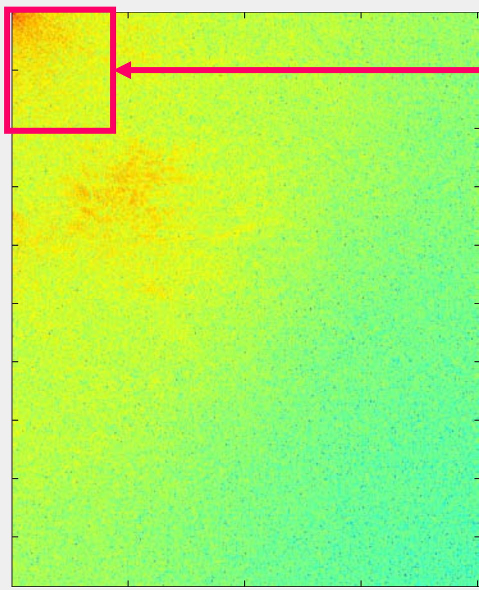
**non funziona!**



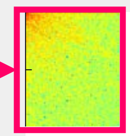
si memorizza solo una parte della FT

**perc=20%**

## Perché compressione?



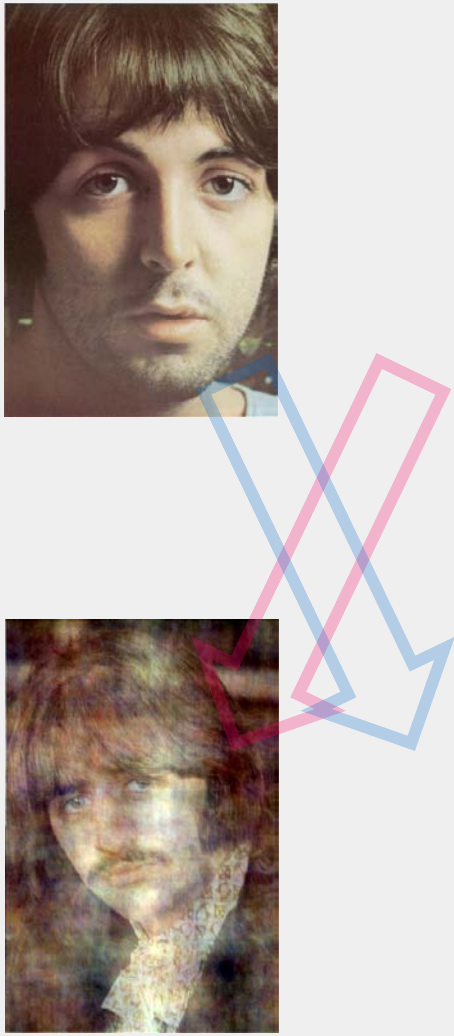
Cosine Transform



**funziona!**

**Esempio:** la FT restituisce, in generale, valori complessi.
Nel ricostruire un'immagine dalla sua FT, quale parte della FT è più importante: l'argomento (o angolo di fase) o il modulo (o magnitude)?

```
fig1=imread('McCartney.jpg');
fig2=imread('Starr.jpg');
figure(1); clf
subplot(1,2,1); imshow(fig1)
subplot(1,2,2); imshow(fig2)
sgtitle('Original images')
% mescola le due Trasformate di Fourier
Ffig1=fft2(fig1);
Ffig2=fft2(fig2);
G1=abs(Ffig1).*exp(1i*angle(Ffig2));
G2=abs(Ffig2).*exp(1i*angle(Ffig1));
% inverte la nuova Trasformata di Fourier
g1=ifft2(G1);
g2=ifft2(G2);
subplot(1,2,1); imshow(uint8(real(g1)))
subplot(1,2,2); imshow(uint8(real(g2)))
```



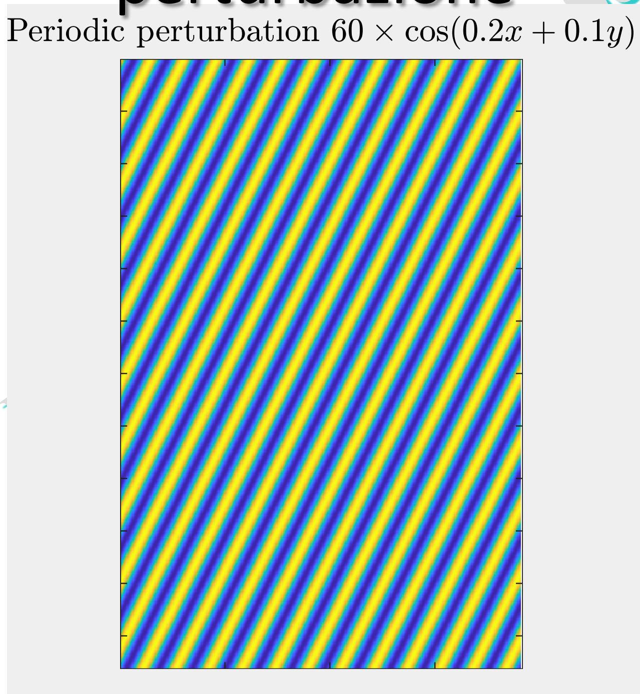
Nel ricostruire un'immagine, l'argomento della FT predomina rispetto al modulo.

**Esempio:** si introduca una perturbazione periodica sulla componente rossa di un'immagine RGB

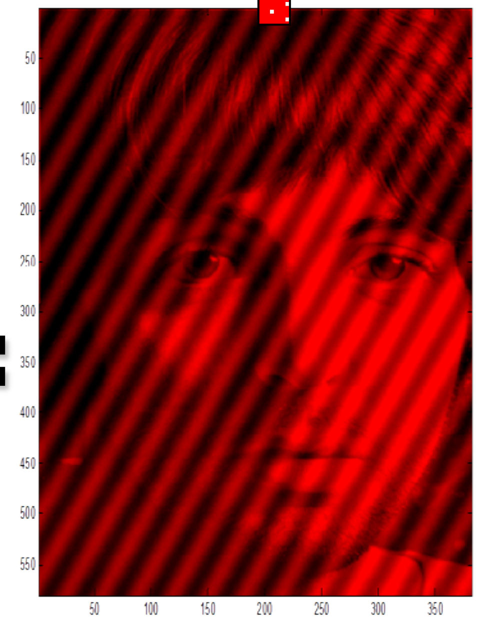
```
f=imread('McCartney.jpg');
[m,n,~]=size(f); [X,Y]=meshgrid(1:n,1:m);
p=60*cos(.2*X+.1*Y); imagesc(p); axis tight; axis equal
pf=f; pf(:,:,1)=pf(:,:,1)+uint8(p);
imagesc(pf); axis equal
fred=uint8(zeros(size(f)));  fred(:,:,1)=f(:,:,1);
imagesc(fred); axis equal; axis tight
```

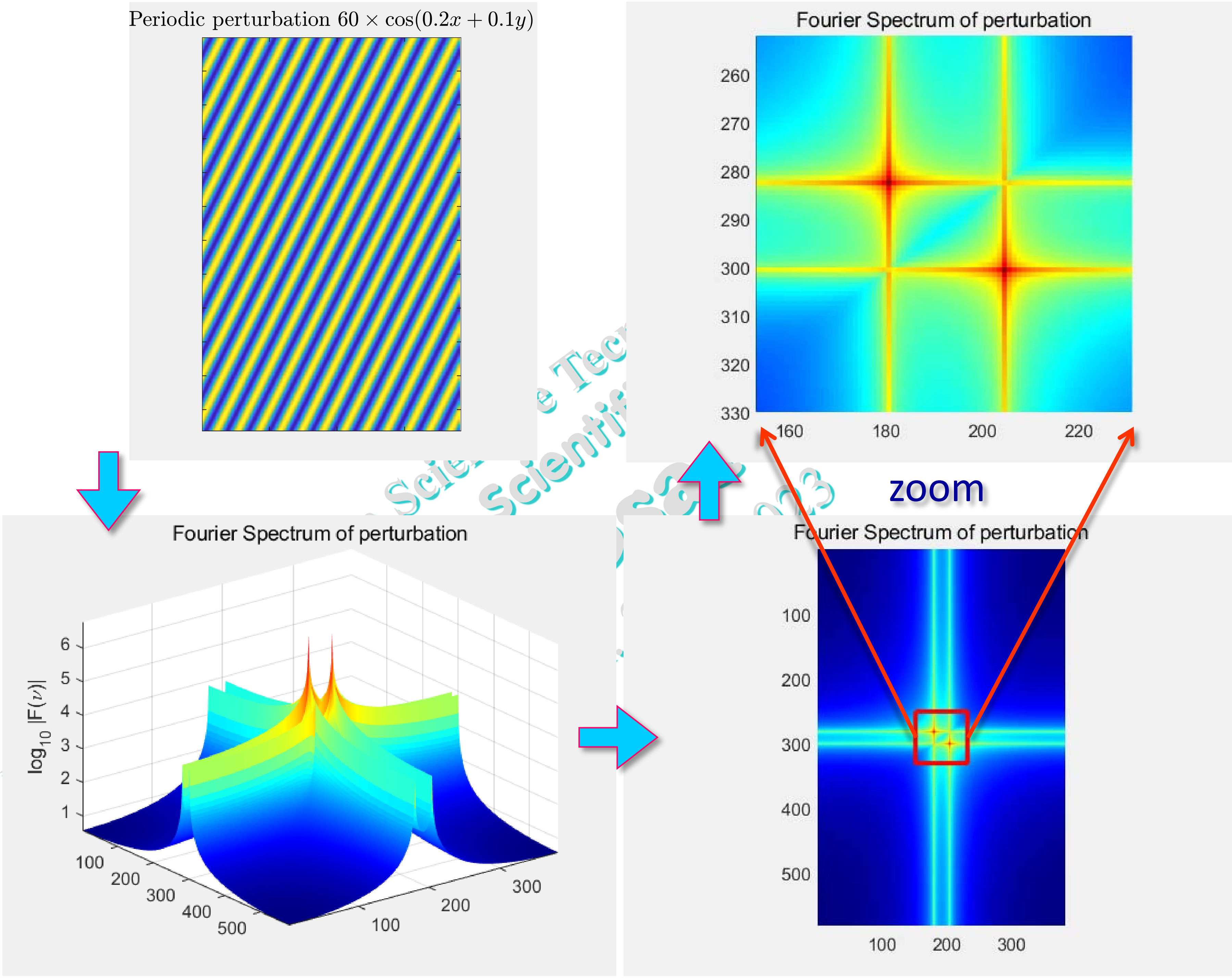
immagine RGB

perturbazione



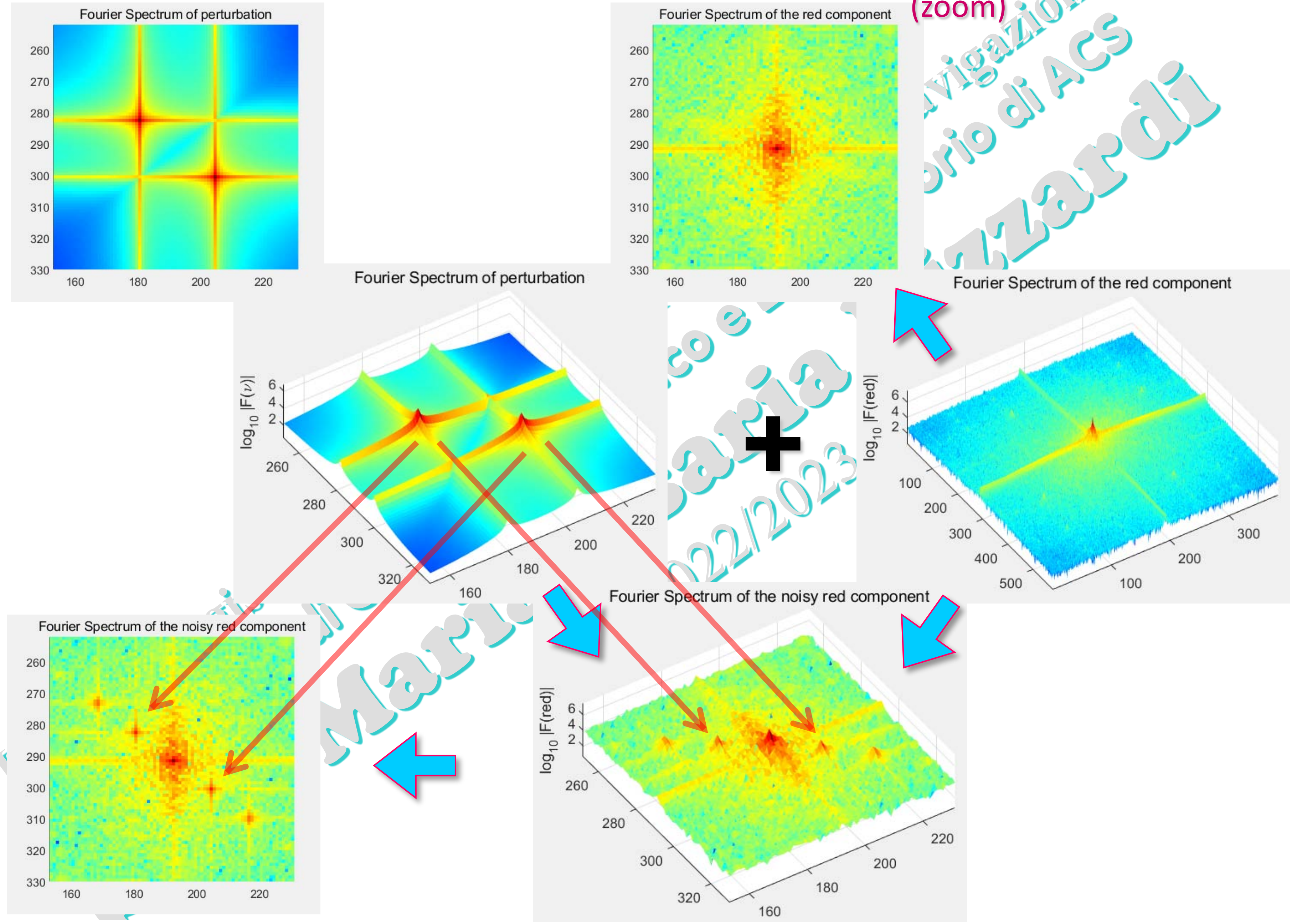

componente rossa




Periodic perturbation $60 \times \cos(0.2x + 0.1y)$
Fourier Spectrum of perturbation
260
270
280
290
300
310
320
330
160
180
200
220
zoom
Fourier Spectrum of perturbation
$\log_{10} |F(\nu)|$
6
5
4
3
2
1
100
200
300
400
500
100
200
300
Fourier Spectrum of perturbation
100
200
300
400
500
100
200
300

spettro della perturbazione

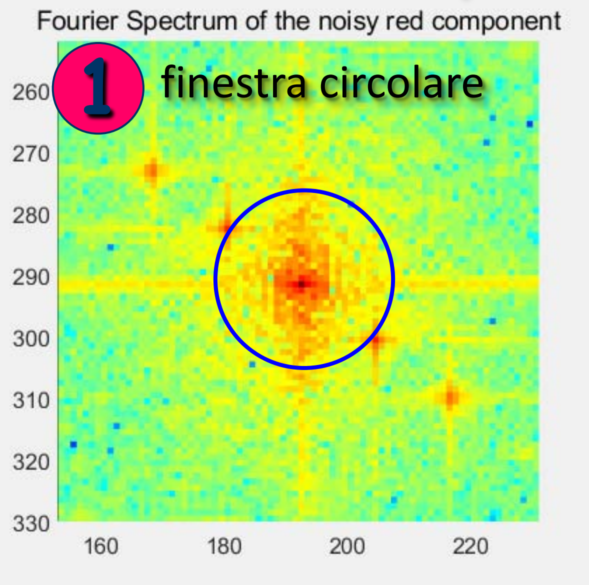
Spettro della componente rossa (zoom)

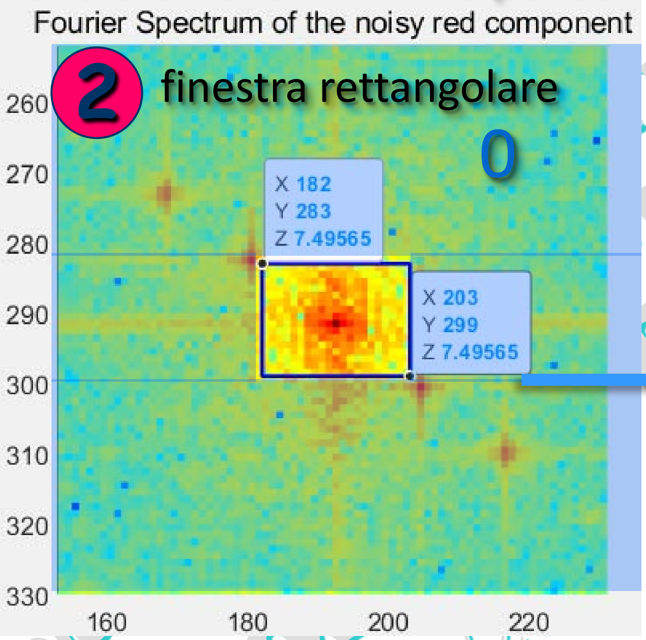


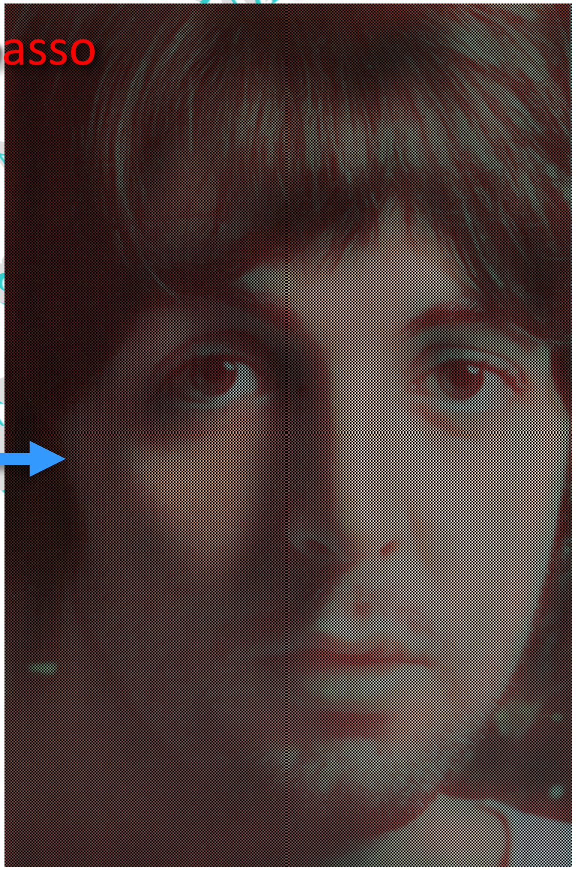

Spettro della componente rossa perturbata

# Per rimuovere le perturbazioni si può filtrare la FT nelle frequenze ...

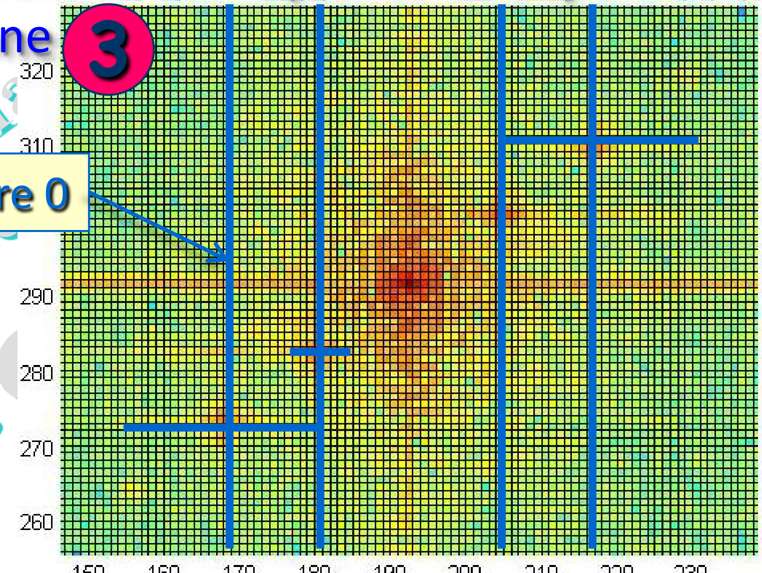
mantenendo solo la parte centrale della FT: filtro passa-basso



azzerando nella FT le frequenze corrispondenti alla perturbazione




**Esercizio**

Implementare in MATLAB tali filtri, e confrontare i loro risultati