



Tratamiento de Señales

Version 2024-I

Implementación de Filtros usando Fourier en 2D

[Capítulo 4]

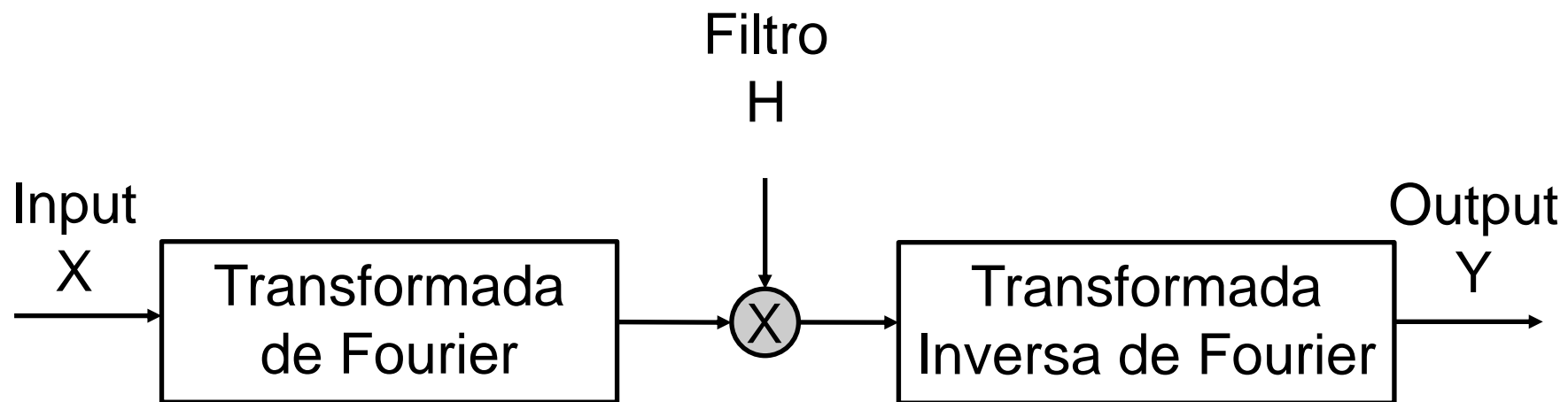
Dr. José Ramón Iglesias

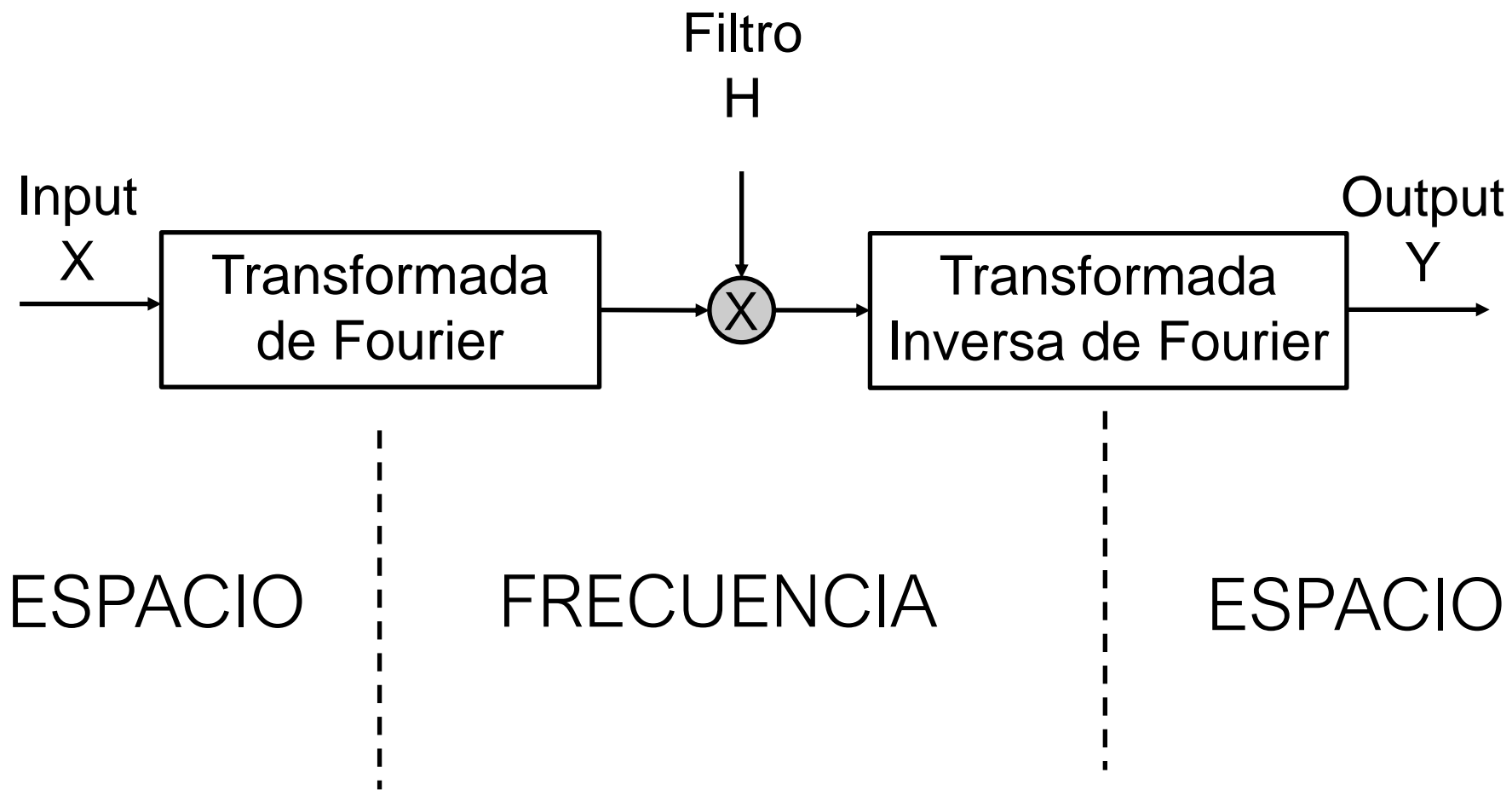
DSP-ASIC BUILDER GROUP

Director Semillero TRIAC

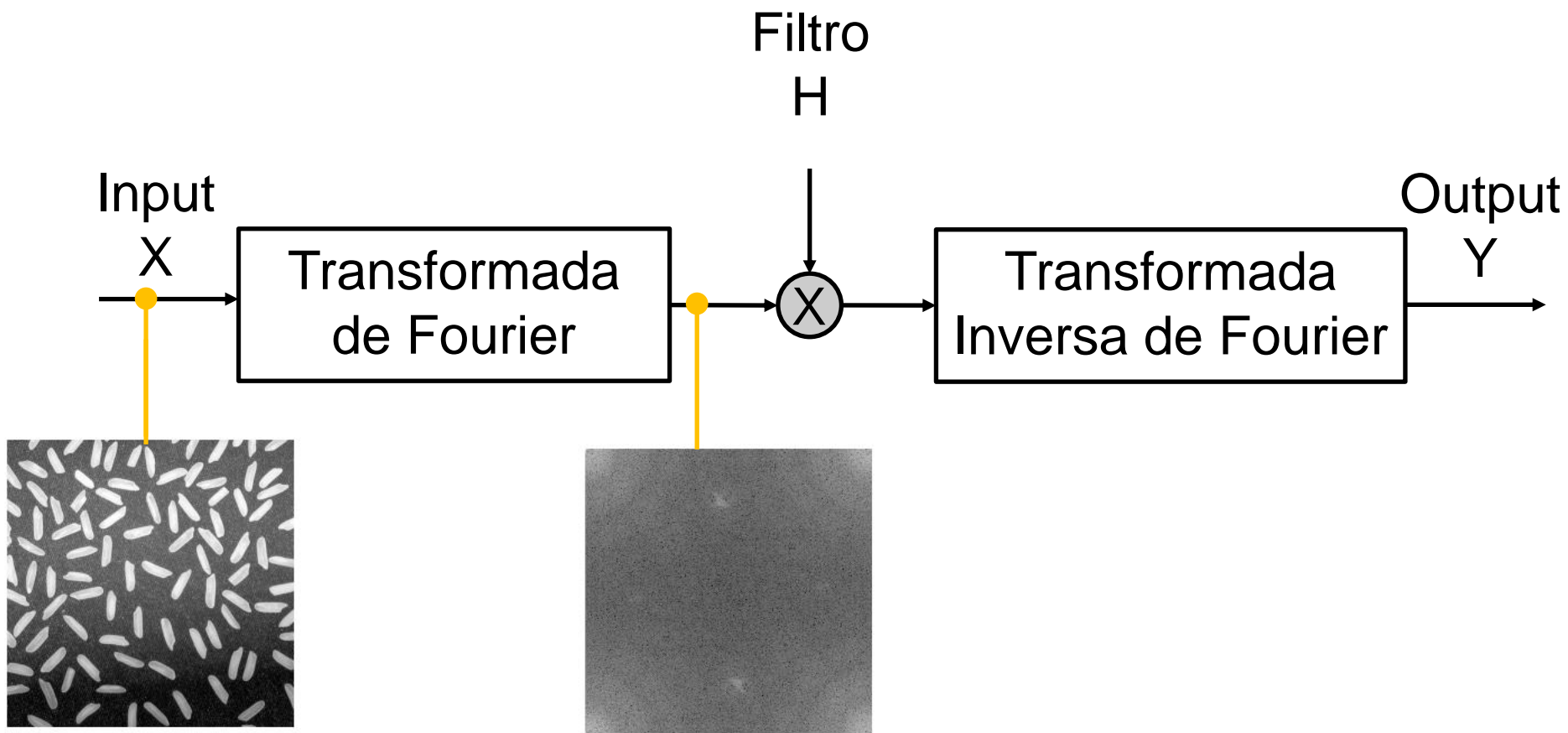
Ingeniería Electronica

Universidad Popular del Cesar

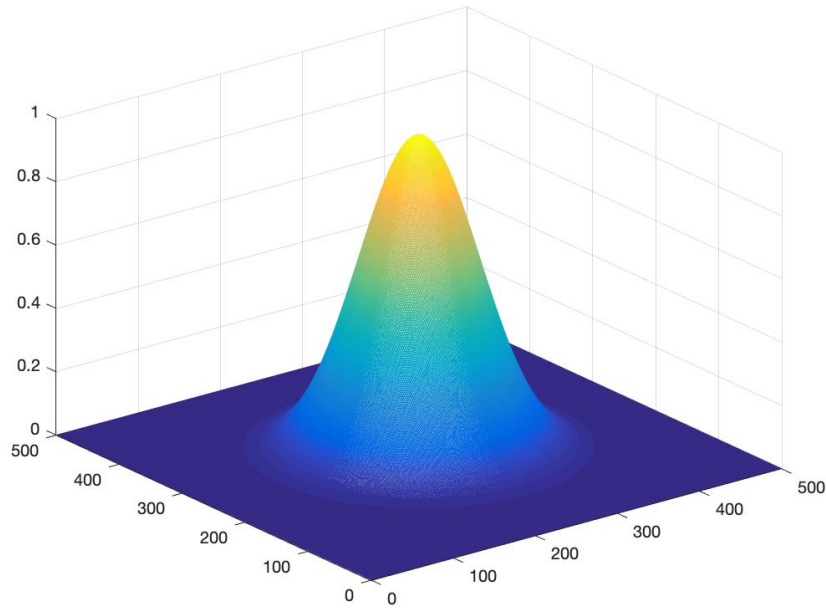




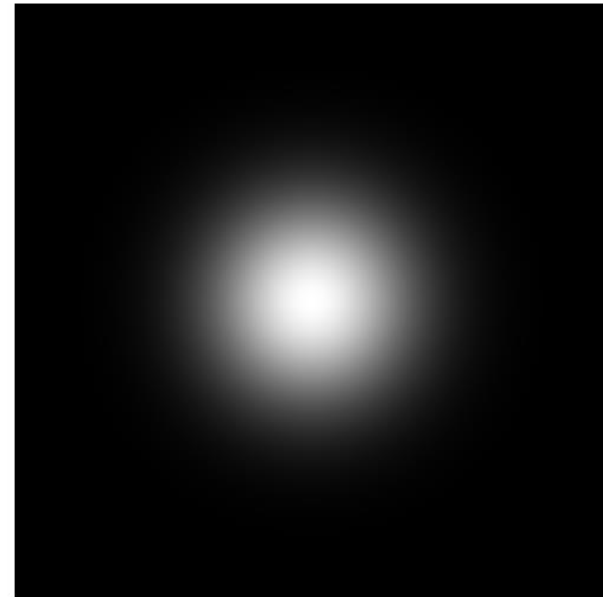
DISEÑO DEL FILTRO EN EL DOMINIO DE LA FRECUENCIA



Si H es un filtro pasa bajos (por ejemplo Gaussiana):

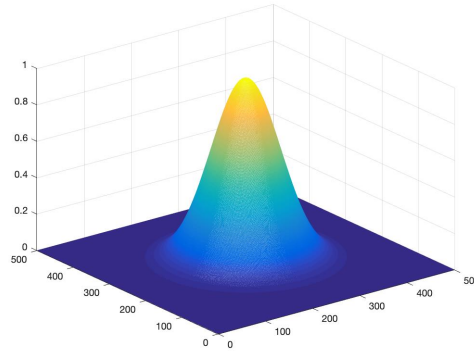


$H(u,v)$
(en el dominio de Fourier)



Top View
(la frecuencia central está en
el centro)

Filtro
H



Input

X



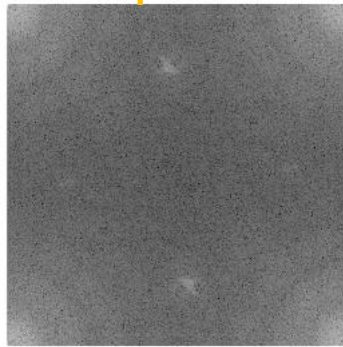
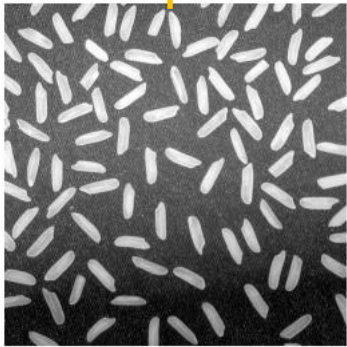
Transformada
de Fourier



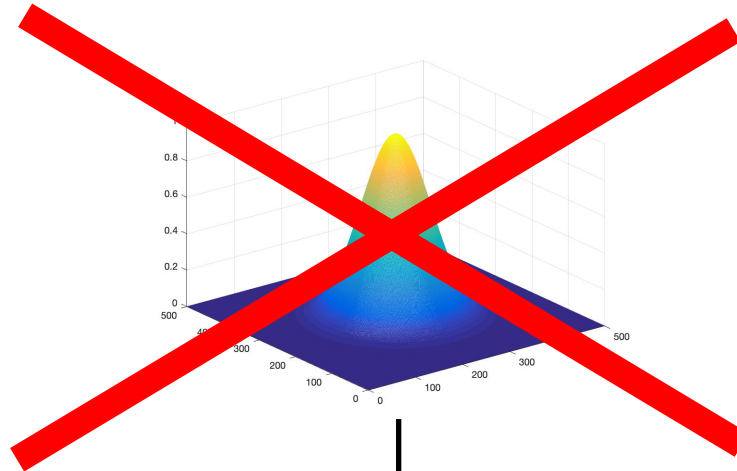
Transformada
Inversa de Fourier

Output

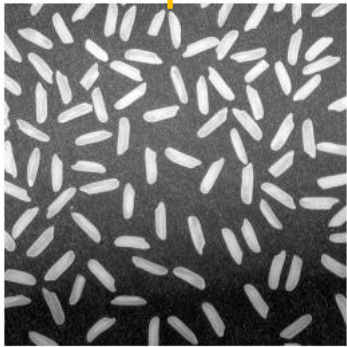
Y



Filtro
H



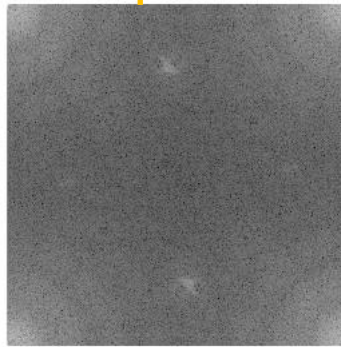
Input
X



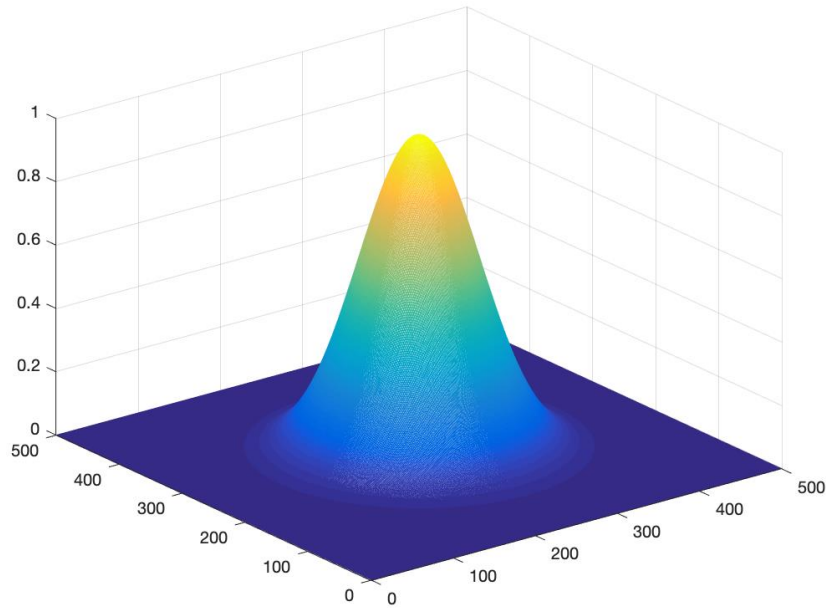
Transformada
de Fourier

Transformada
Inversa de Fourier

Output
Y



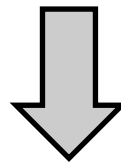
Si H es un filtro pasa bajos (por ejemplo Gaussiana):



$H(u,v)$
(en el dominio de Fourier)

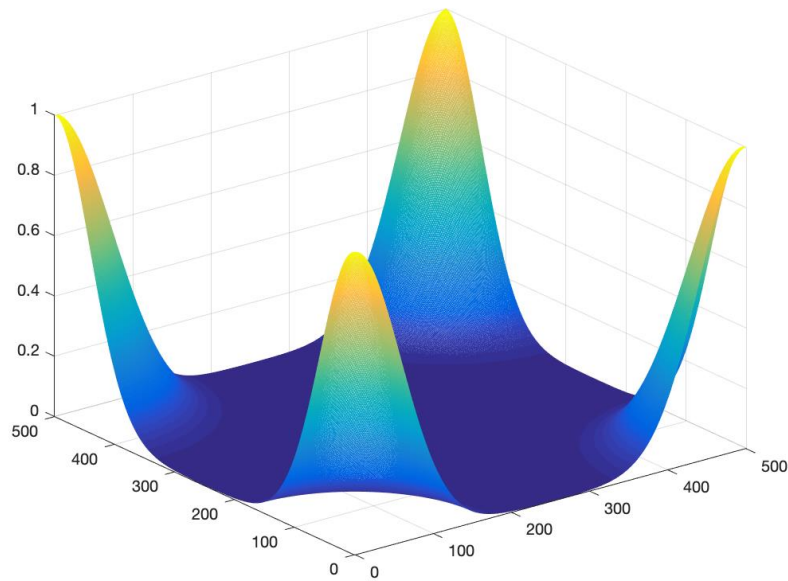


Top View
(la frecuencia central está en
el centro)



fftshift

Si H es un filtro pasa bajos (por ejemplo Gaussiana):

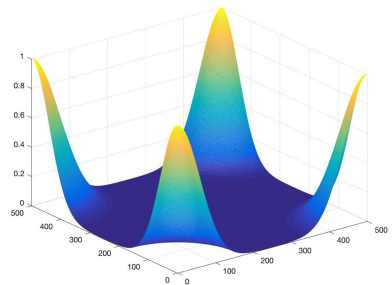
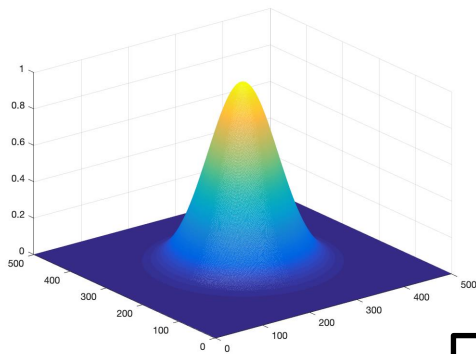


$\text{fftshift}(H)$
(en el dominio de Fourier)

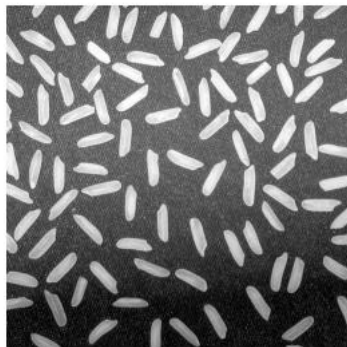


Top View
(la frecuencia central está en
esquina superior izquierda)

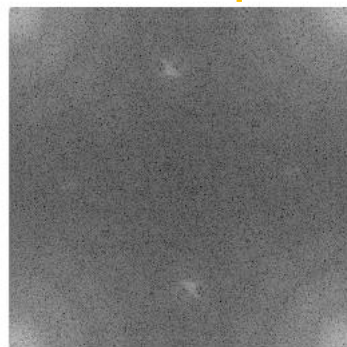
Filtro
H



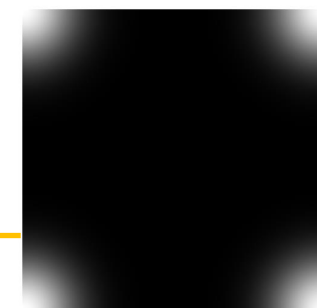
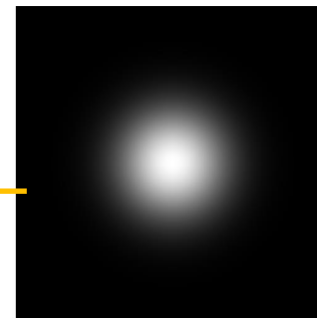
Input
X



Transformada
de Fourier

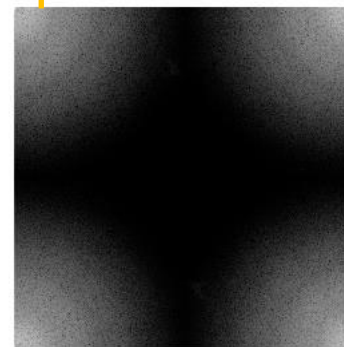


fftshift



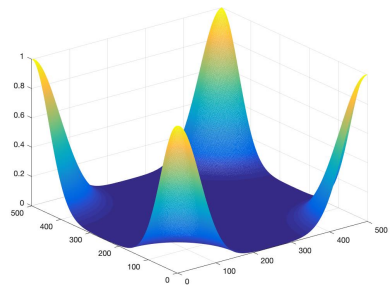
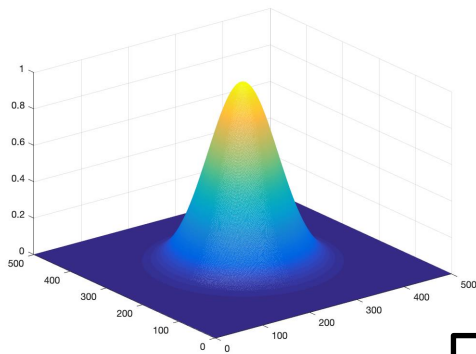
X

Transformada
Inversa de Fourier

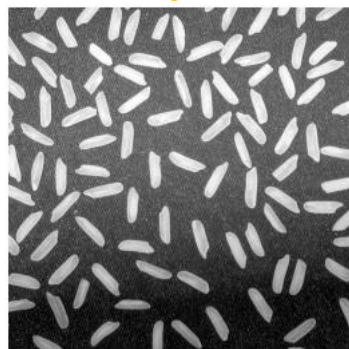


Output
Y

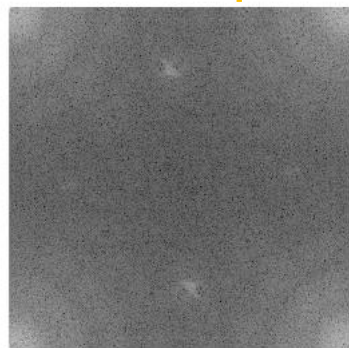
Filtro
H



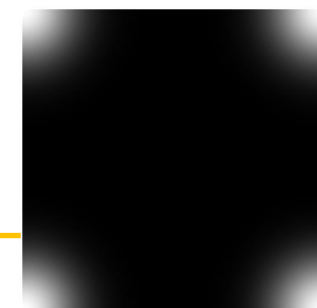
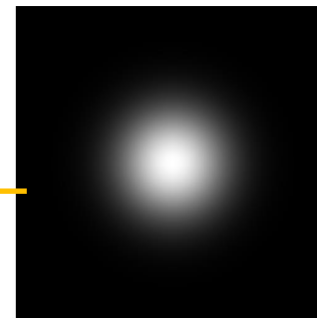
Input
X



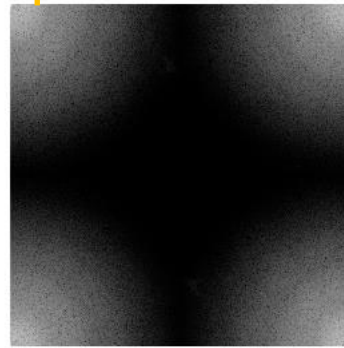
Transformada
de Fourier



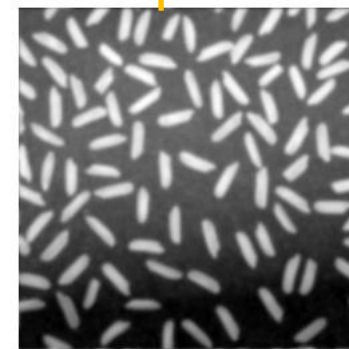
fftshift



X

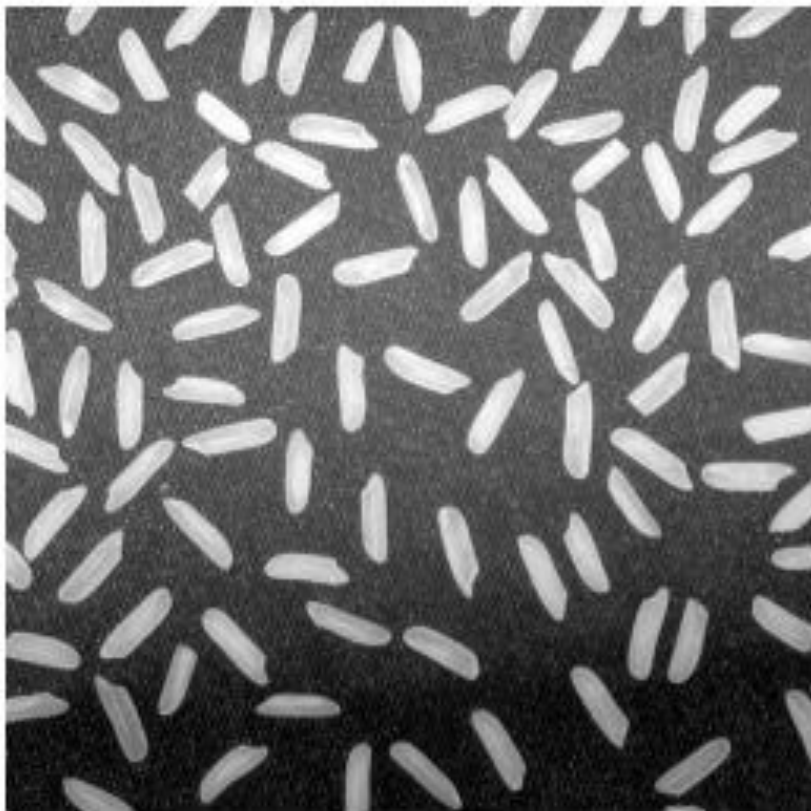


Transformada
Inversa de Fourier

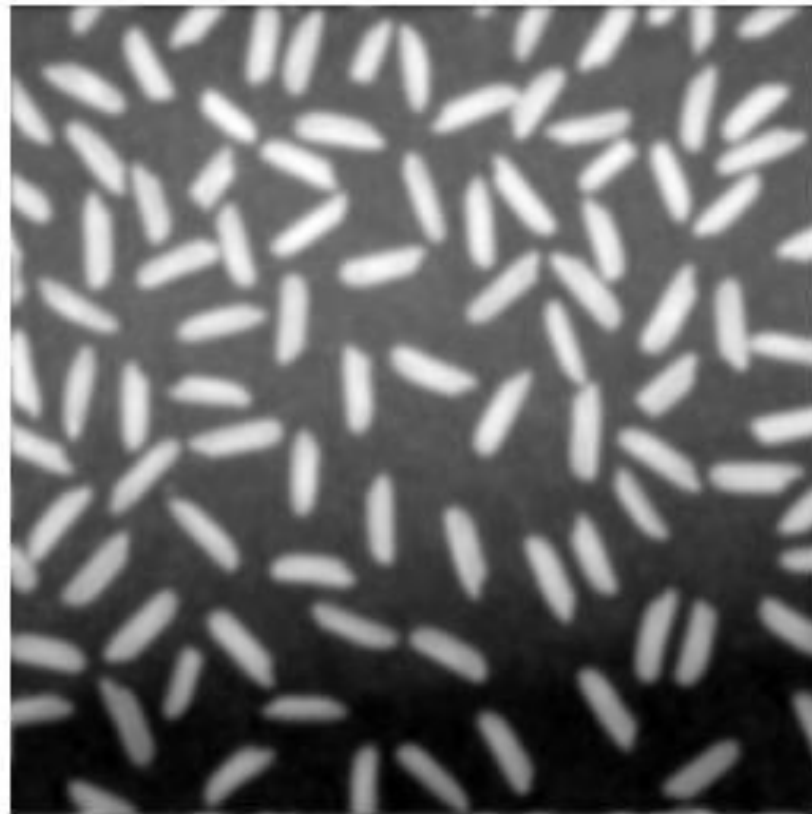


Output
Y

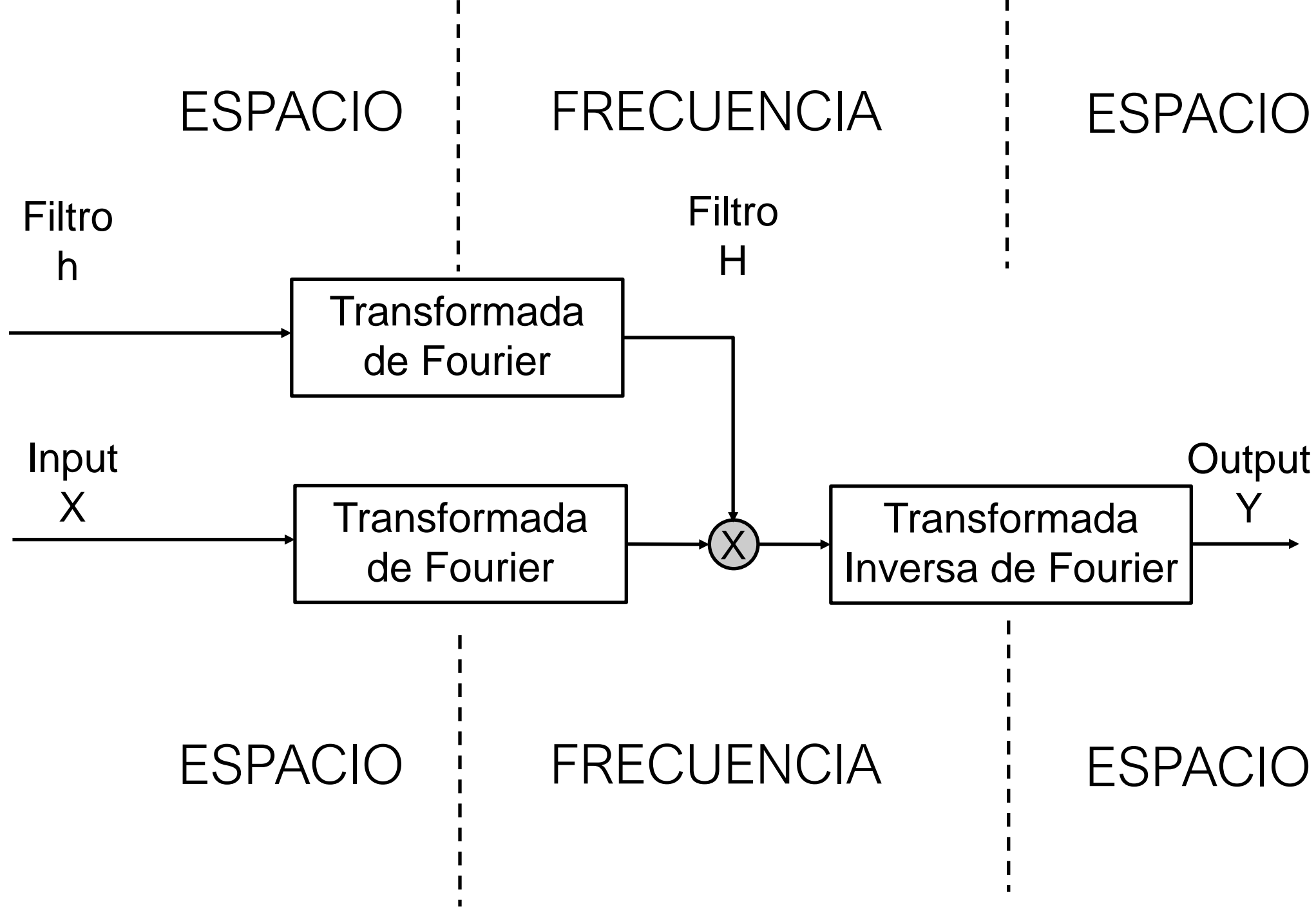
Input
X

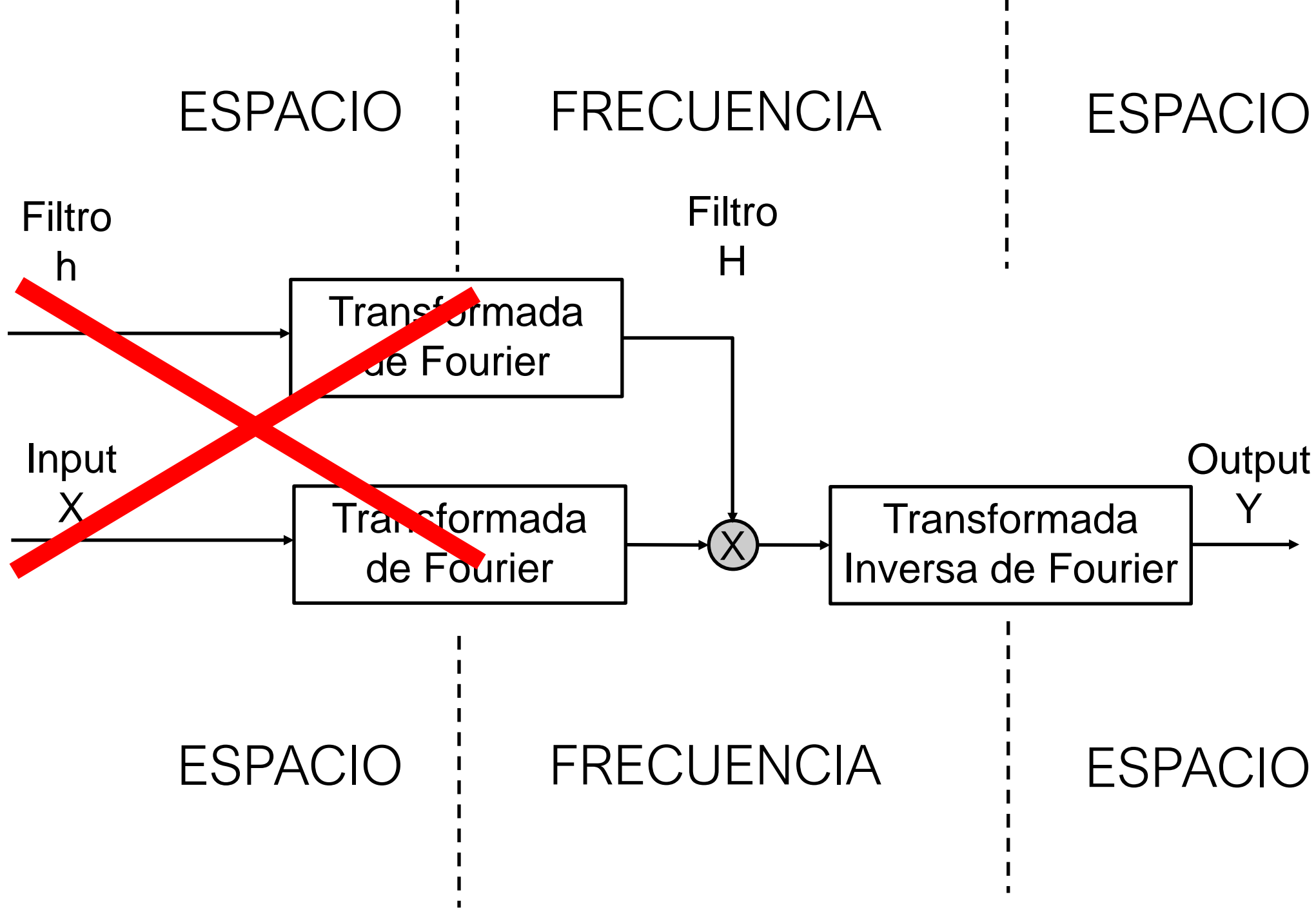


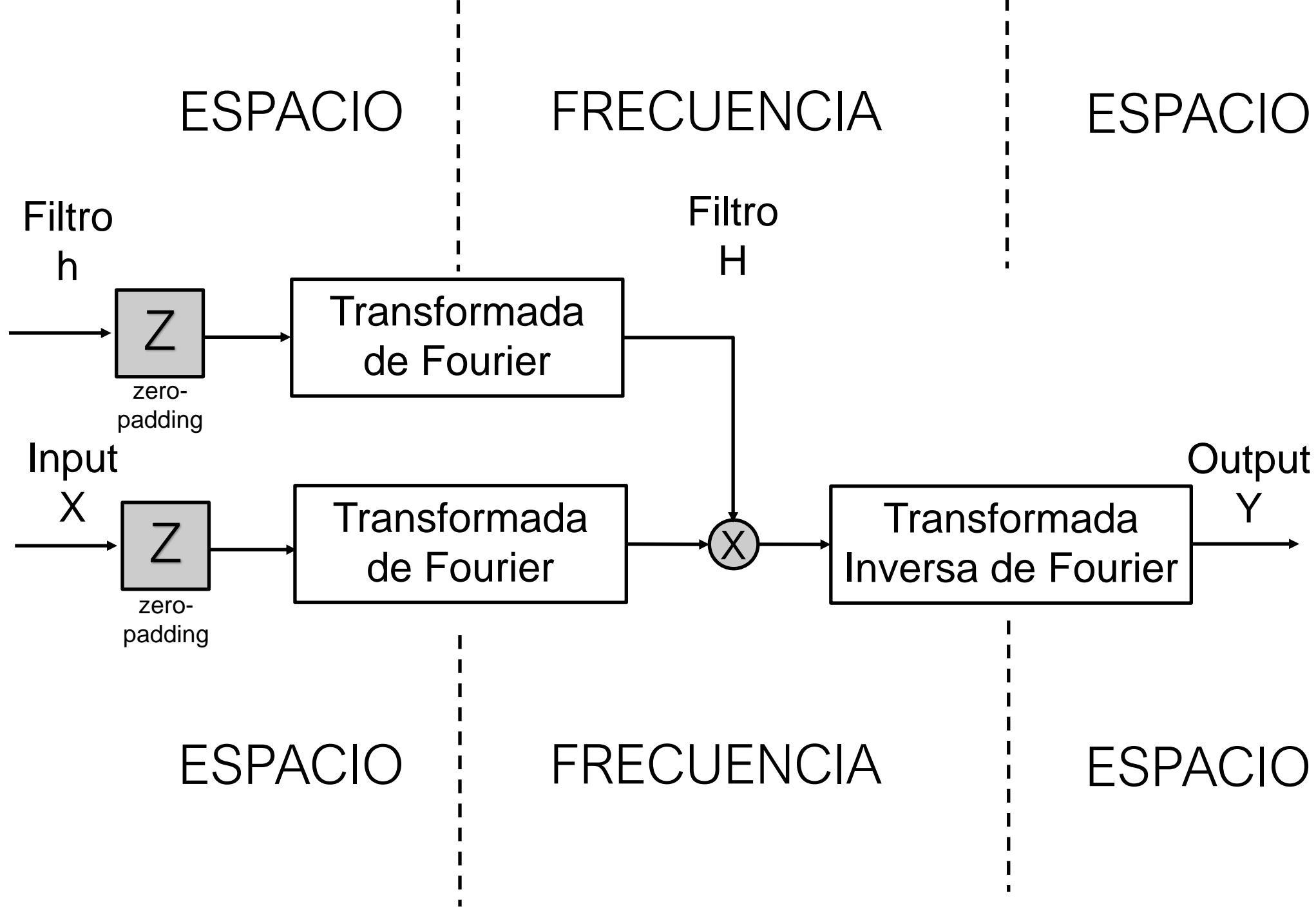
Output
Y

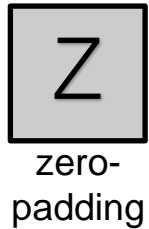


DISEÑO DEL FILTRO EN EL DOMINIO DEL ESPACIO









La convolución en 2D es similar a la convolución en 1D: es necesario hacer un *zero padding* para evitar el traslape.

Este programa obtiene el mismo resultado de
`J=conv2(X,h,'same')`.

```
function J = conv2fft(X,h) ;  
[A,B] = size(X) ;  
[C,D] = size(h) ;  
P = A + C - 1 ;  
Q = B + D - 1 ;  
Xp = zeros(P,Q) ;  
Xp(1:A,1:B) = X ;  
hp = zeros(P,Q) ;  
hp(1:C,1:D) = h ;  
  
Xpf = fft2(Xp) ;  
Hpf = fft2(hp) ;  
Ypf = Xpf.*Hpf ;  
Yp = ifft2(Ypf) ;  
m1 = (C+1)/2 ;  
m2 = (D+1)/2 ;  
Y = Yp(m1:m1+A-1,m2:m2+B-1) ;
```

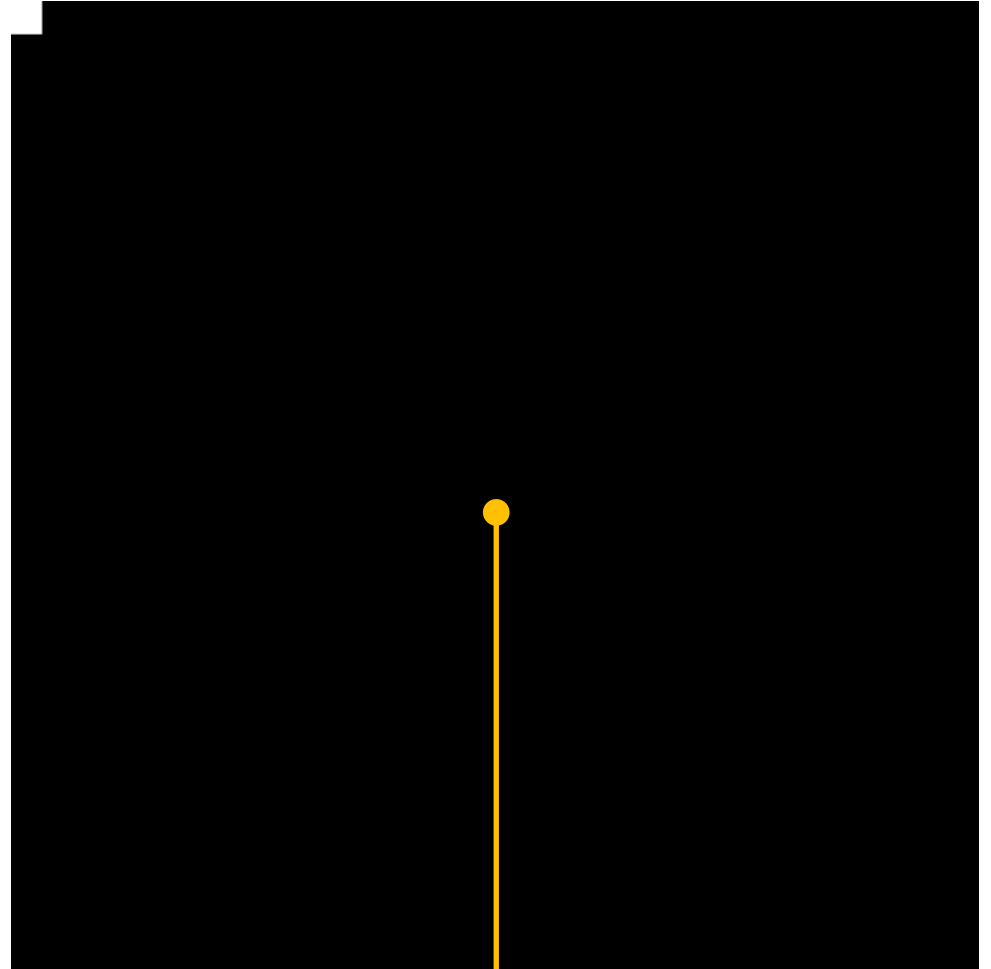
X_p :
Input (X) con zero padding



Size: $P \times Q$

zero-
padding

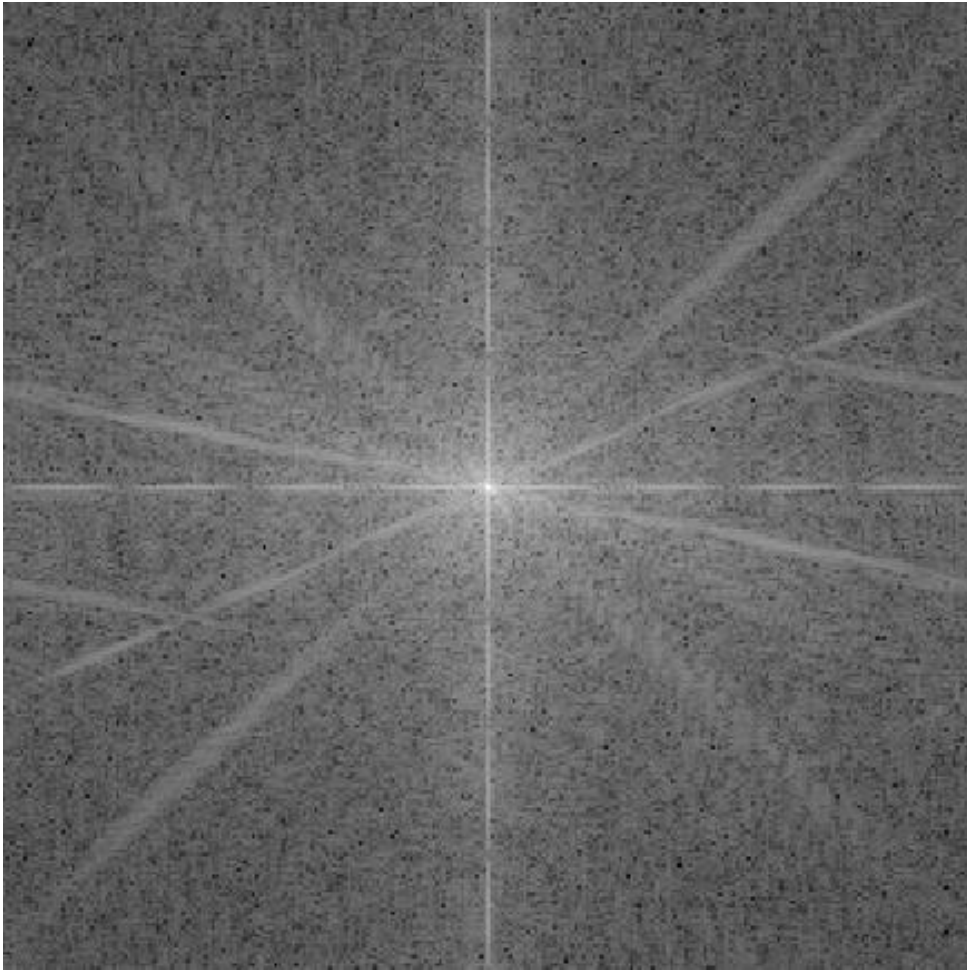
h_p :
Filtro Promedio (h) con zero padding



Size: $P \times Q$

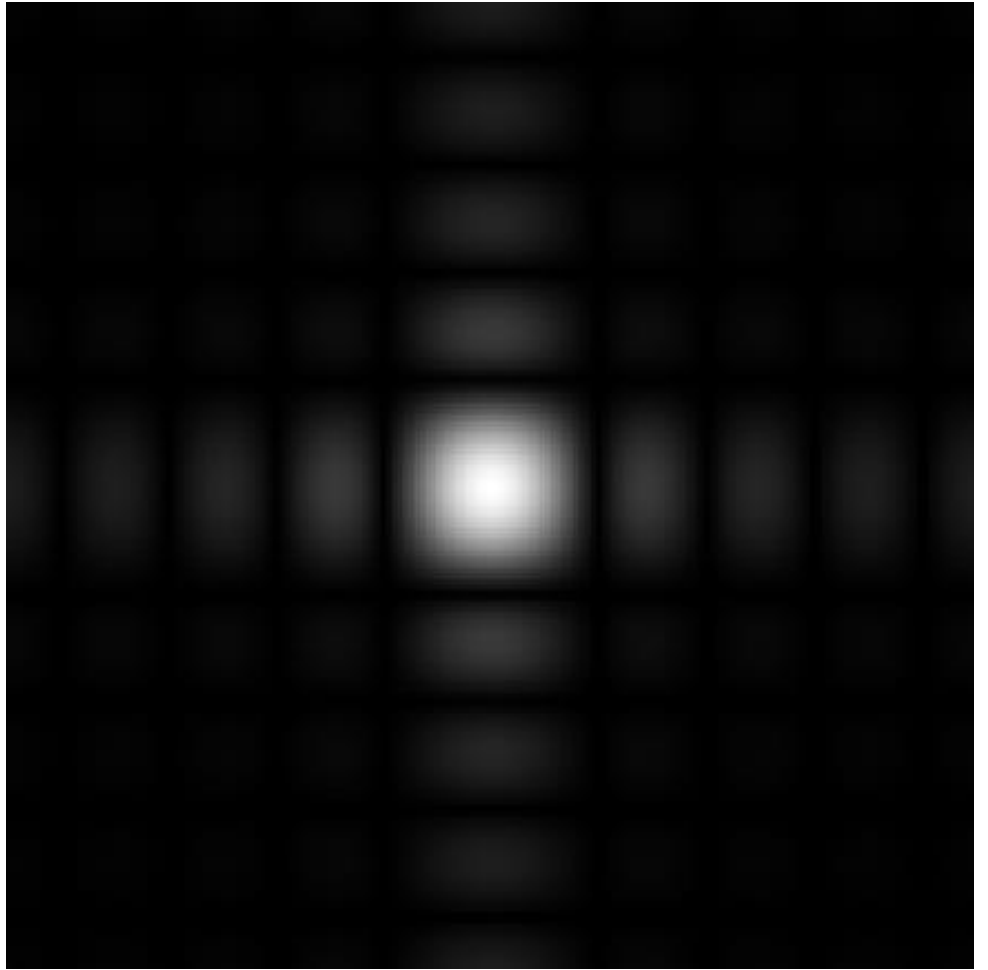
zero-
padding

Transformada de Fourier de X_p



Size: P x Q

Transformada de Fourier de h_p



Size: P x Q

(resultados con `fftshift`)

Transformada Inversa de la multiplicación



Transformada Inversa de la multiplicación



(con recorte)

