BME I5000: Biomedical Imaging

2D Fourier Transform

Lucas C. Parra, parra@ccny.cuny.edu

Fourier Transform (1D)

The **Fourier Transform** (FT) is defined as*

$$H(k) = FT[h(x)] = \int_{-\infty}^{\infty} dx\, h(x)\, e^{-i2\pi kx}$$

The FT is an invertible transformation

$$h(x) = FT^{-1}[H(k)] = \int_{-\infty}^{\infty} dk\, H(k)\, e^{i2\pi kx}$$

We can show this using $\int_{-\infty}^{\infty} dk\, e^{-i2\pi kx} = \delta(x)$

$$\int_{-\infty}^{\infty} dk\, H(k)\, e^{i2\pi kx} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dk\, dx'\, h(x')\, e^{-i2\pi k(x'-x)} = h(x)$$

* Notational convention: Use *k* for spacial, and ω for temporal frequency.

Fourier Transform – Frequency Domain

$H(k)$ is in general complex valued

$$H(k) = R(k) + i\,I(k) = A(k)\,e^{i\phi(k)}$$

with $R(k)$ = Re(H(k)), $I(k)$ = Im($H(k)$) and

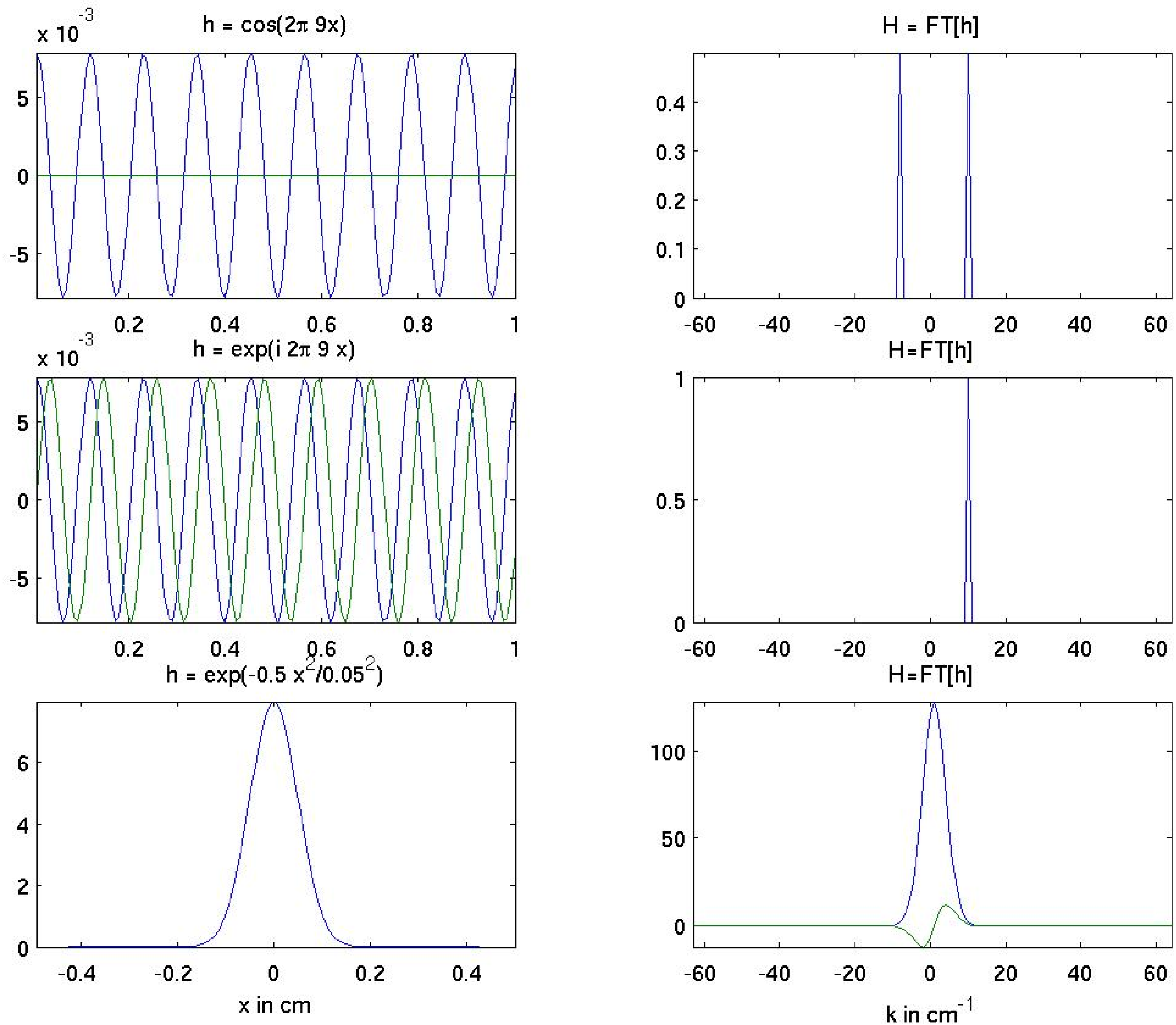
Amplitude: $$A(k) = |H(k)| = \sqrt{R(k)^2 + I(k)^2}$$

Phase: $$\phi(k) = \arctan\left(\frac{I(k)}{R(k)}\right)$$

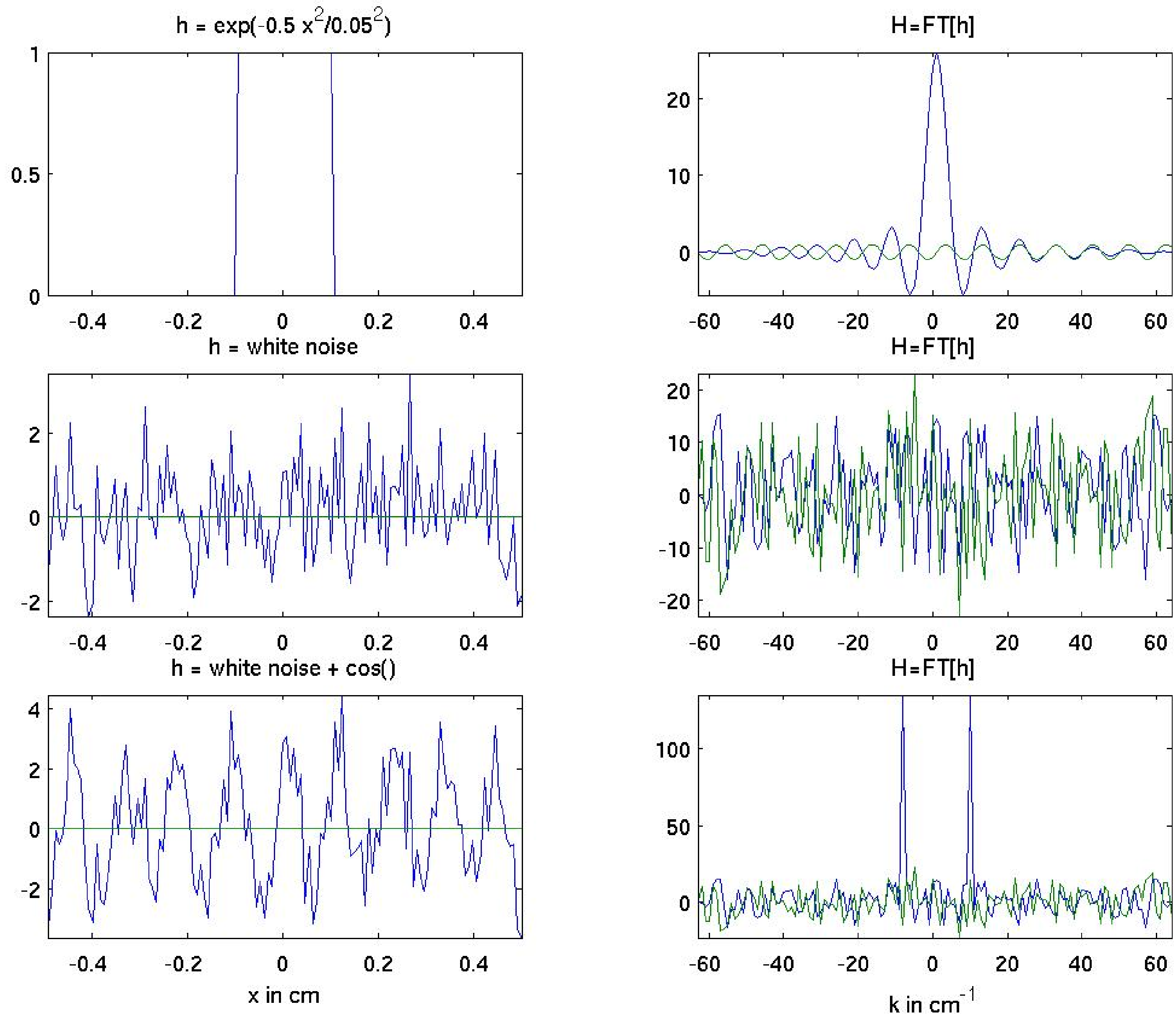
Positive frequencies $k>0$: counterclockwise rotation
Negative frequencies $k<0$: clockwise rotation

$$H(k) = \delta(k - k_0), \qquad h(x) = e^{i2\pi k_0 x}$$

Fourier Transform - Examples



Fourier Transform - Examples



Fourier Transform – Convolution Theorem

Convolution is defined as

$$b(x) = \int_{-\infty}^{\infty} dx' \, h(x') \, g(x - x') = h(x) * g(x)$$

Convolution Theorem states that

$$B(k) = FT[h(x) * g(x)] = H(k) G(k)$$

Note that with the convolution theorem we can implement convolution as a multiplication in the frequency domain.

$$\begin{tikzcd}
h(x) \arrow[r, "FT"] & H(k) \arrow[dr] & & & \\
& & \times \arrow[r] & B(k) \arrow[r, "FT^{-1}"] & b(x) \\
g(x) \arrow[r, "FT"] & G(k) \arrow[ur] & & &
\end{tikzcd}$$

Fourier Transform – Convolution Theorem

Proof

$$FT\left[h(x)*g(x)\right]=\int_{-\infty}^{\infty} dx\, h(x)*g(x)\, e^{-i2\pi kx}=$$

$$=\int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dx'\, h(x')\, g(x-x')\, e^{-i2\pi kx}$$

$$=\int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dx'\, h(x')\, g(x)\, e^{-i2\pi k(x+x')}$$

$$=\int_{-\infty}^{\infty} dx'\, h(x')\, e^{-i2\pi kx'} \int_{-\infty}^{\infty} dx\, g(x)\, e^{-i2\pi kx}$$

$$=H(k)\,G(k)$$

Fourier Transform - Inverse Filter

With the Convolution Theorem we can derive the inverse convolution (or inverse filter)

$$b(x) = g(x) * h(x) \Leftrightarrow B(k) = G(k)\,H(k)$$

Therefore

$$G(k) = \frac{B(k)}{H(k)}$$

And the inverse filter is given by the inverse FT of $H^{-1}(k)$:

$$g(x) = FT^{-1}\left[\frac{1}{H(k)}\right] * b(x)$$

Fourier Transform – 2D and higher

The Fourier transform in 2D is defined as

$$H(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} dx\, dy\, h(x, y)\, e^{-i 2\pi (k_x x + k_y y)}$$

The inverse transform is defined correspondingly.

Note that the exponent is **separable** and therefor the Fourier transform can be applied **sequentially in each dimension**!

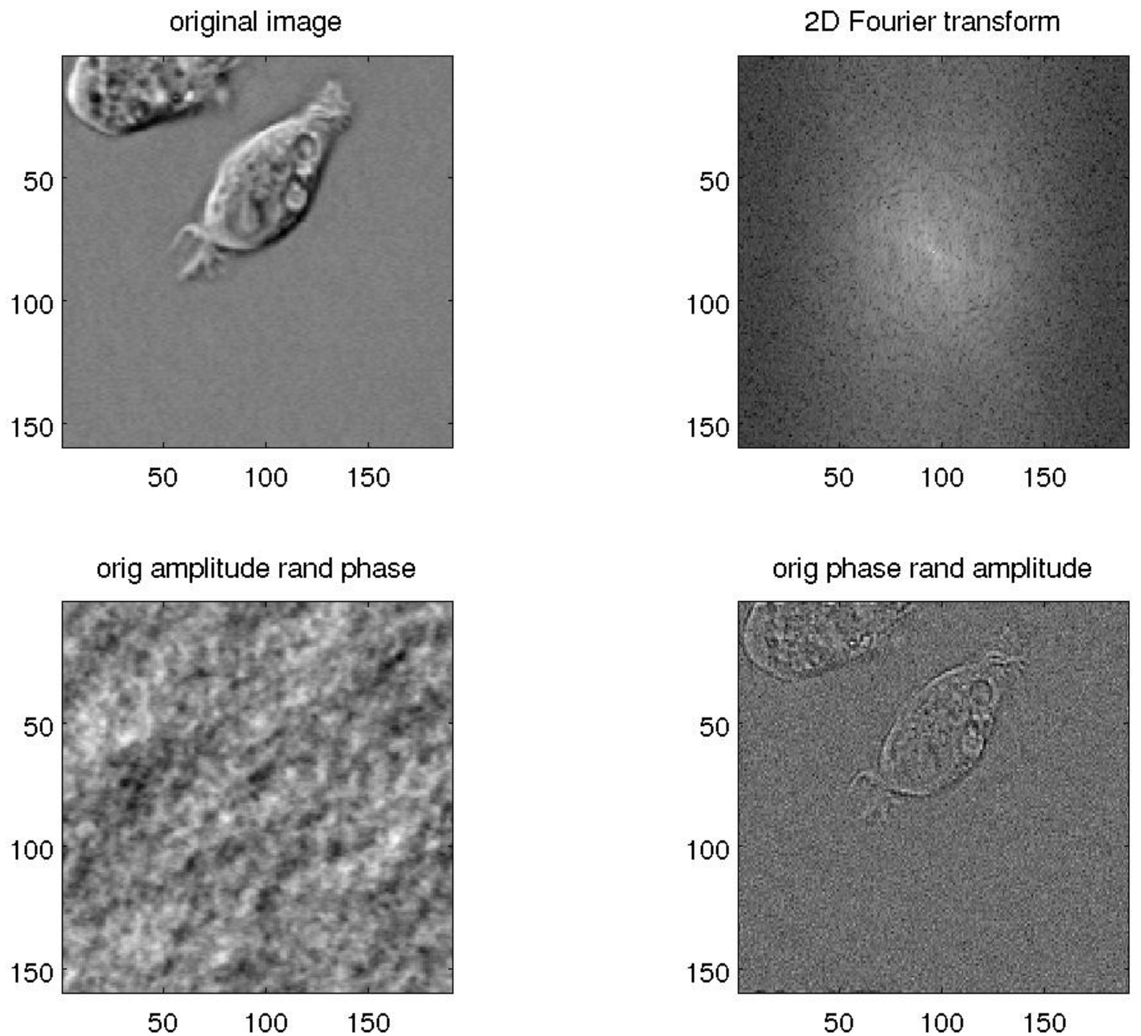
$$H(k_x, k_y) = \int_{-\infty}^{\infty} dy\, e^{-i 2\pi k_y y} \int_{-\infty}^{\infty} dx\, h(x, y)\, e^{-i 2\pi k_x x}$$

In multiple dimensions with $\boldsymbol{k} = [k_x, k_y, k_z, \ldots]^T$, $\boldsymbol{r} = [x, y, z, \ldots]^T$

$$H(\boldsymbol{k}) = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} d\boldsymbol{r}\, h(\boldsymbol{r})\, e^{-i 2\pi \boldsymbol{k} \cdot \boldsymbol{r}}$$

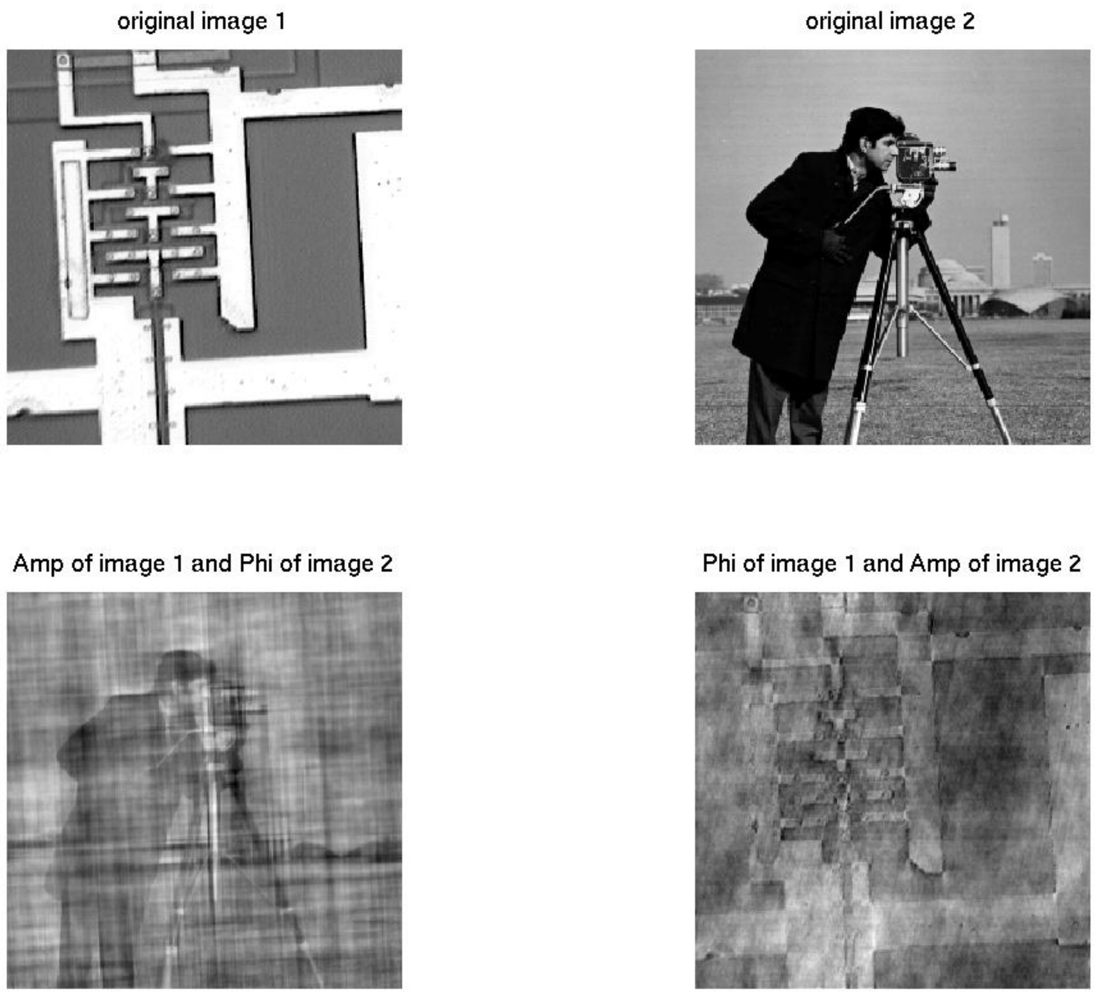
2D Fourier Transform – Phase and Amplitude

Note that the in images the phase carries most of the information

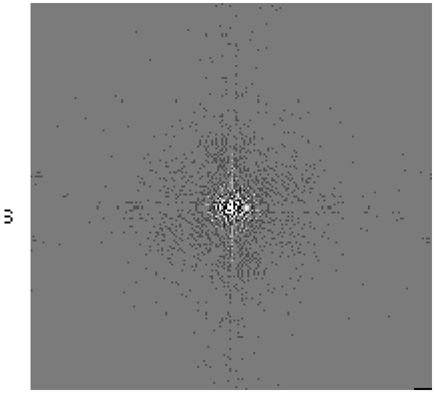


2D Fourier Transform – Phase and Amplitude

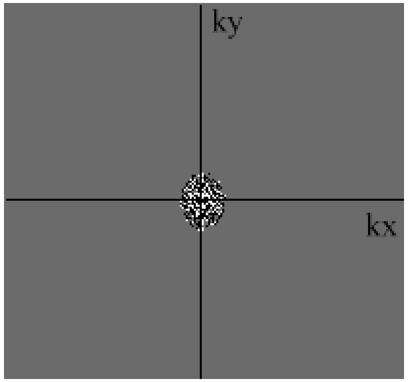
Note that the in images the phase carries most of the information



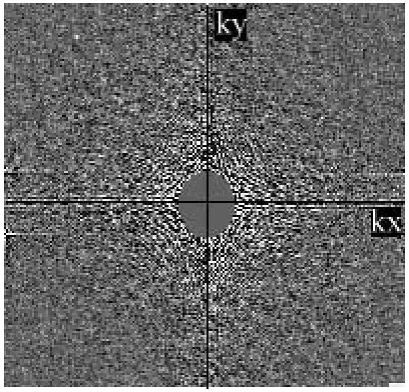
Fourie Transform k-space



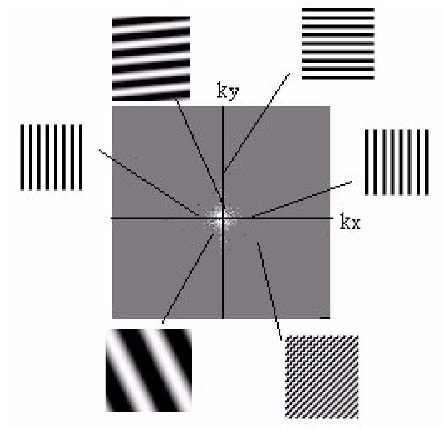
The Mona Lisa in k-Space



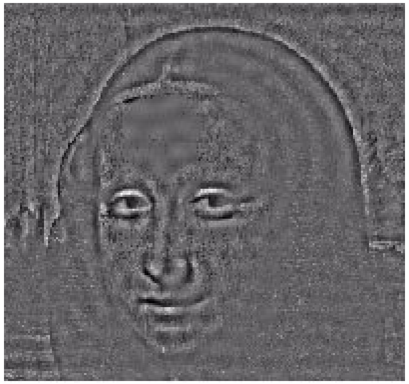
k-Space



k-Space



Low Frequency Mona



High Frequency Mona

Source: (C.A. Mistretta)

Fourier Transform – 2D Convolution

The 2D convolution with a PSF $h(x,y)$ is

$$b(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dx'dy'h(x',y')g(x-x',y-y')$$
$$=h(x,y)*g(x,y)$$

The Convolution Theorem applies in higher dimensions as well.

$$B(k_x,k_y)=G(k_x,k_y)H(k_x,k_y)$$

Even though $h(x,y)$ may not be separable ($h(x,y) \neq h(x)*h(y)$) with the convolution theorem we never have to truly compute a 2D convolution.

Fourier Domain System Response

Consider stationary oscillatory input to a LSI system $h(x)$:

$$g(x) = e^{i2\pi kx}$$

$$b(x) = h(x) * g(x) = \int_{-\infty}^{\infty} dx' h(x') e^{i2\pi k(x-x')} = H(k) e^{i2\pi kx}$$

The output is the input times the FT of the impulse response

$$e^{i2\pi kx} \longrightarrow \boxed{H(k)} \longrightarrow H(k)e^{i2\pi kx}$$

The oscillation with frequency k has been modified in phase ϕ and amplitude A

$$A = |H(k)| \qquad \phi = arg(H(k))$$

$$H(k) = A e^{i\phi}$$

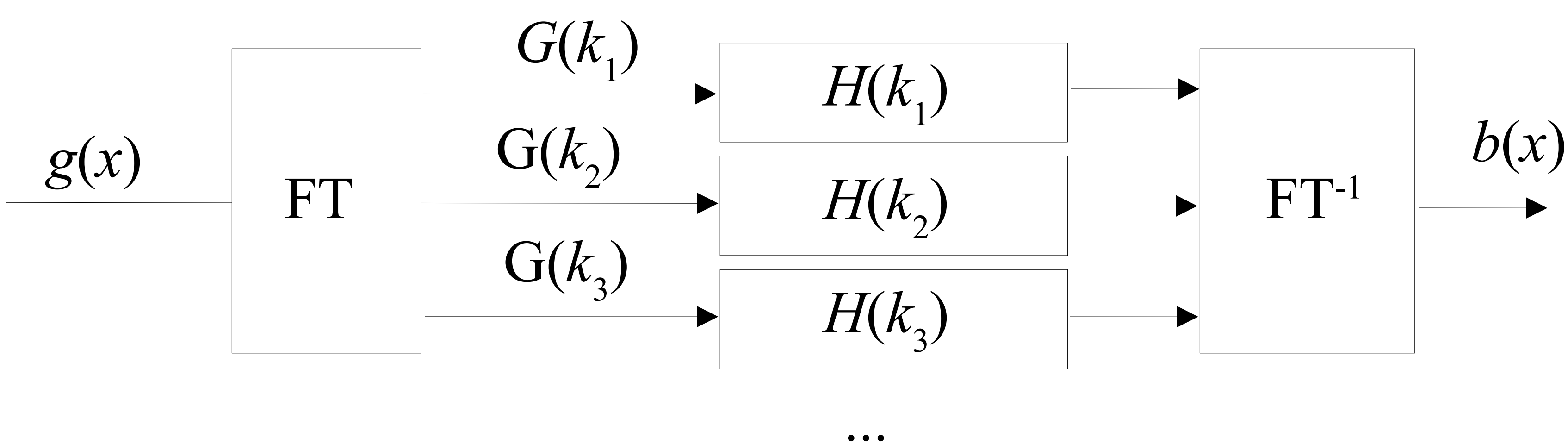
Fourier Domain System Response

FT inversion formula tells us that arbitrary input g(x) can be decomposed into sum of oscillations weighted by $G(k)$

$$g(x)=\int_{-\infty}^{\infty} dk\, G(k)\, e^{i2\pi kx}$$

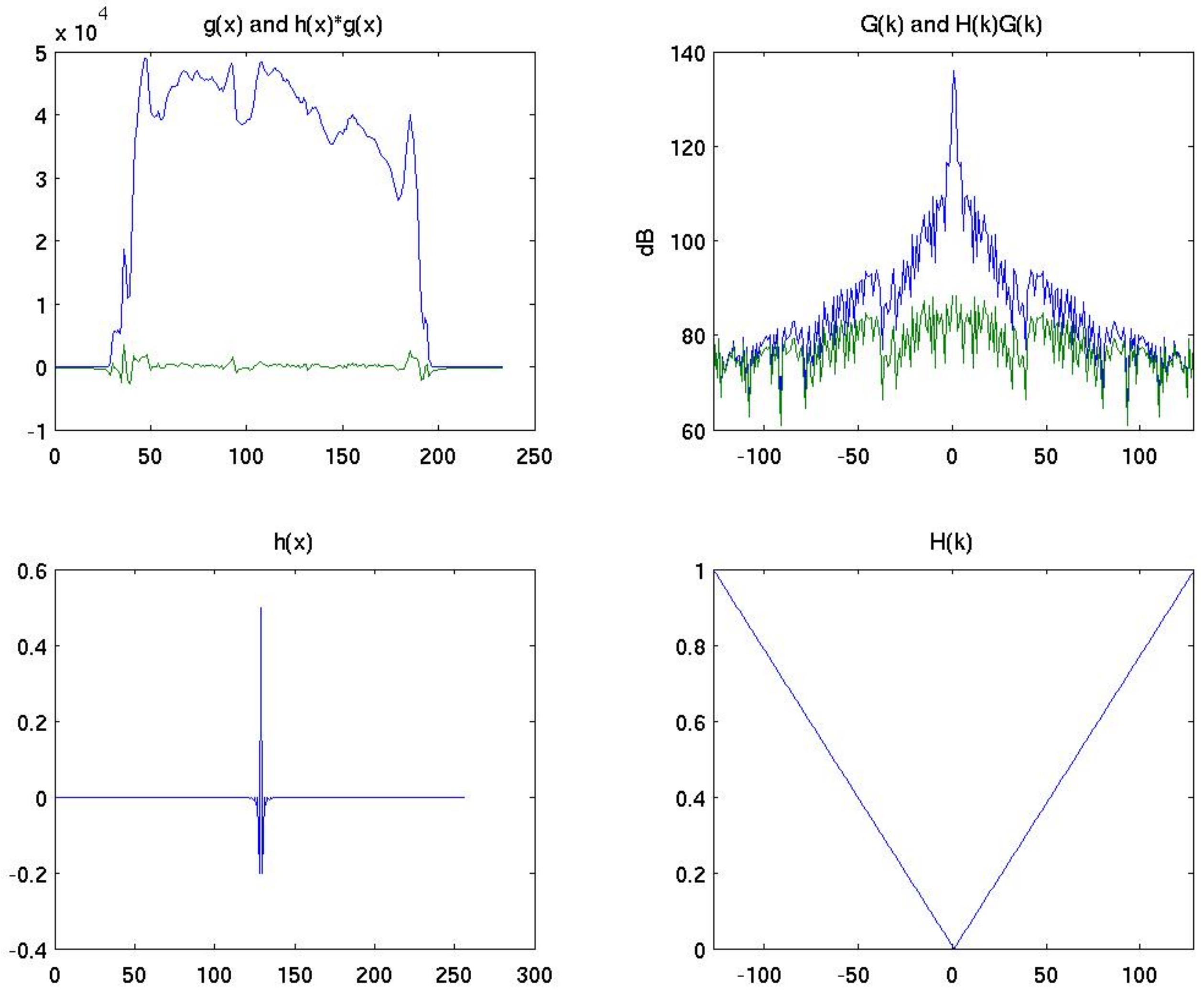
The system response to that is given by the convolution theorem

$$B(k)=H(k)\,G(k)$$



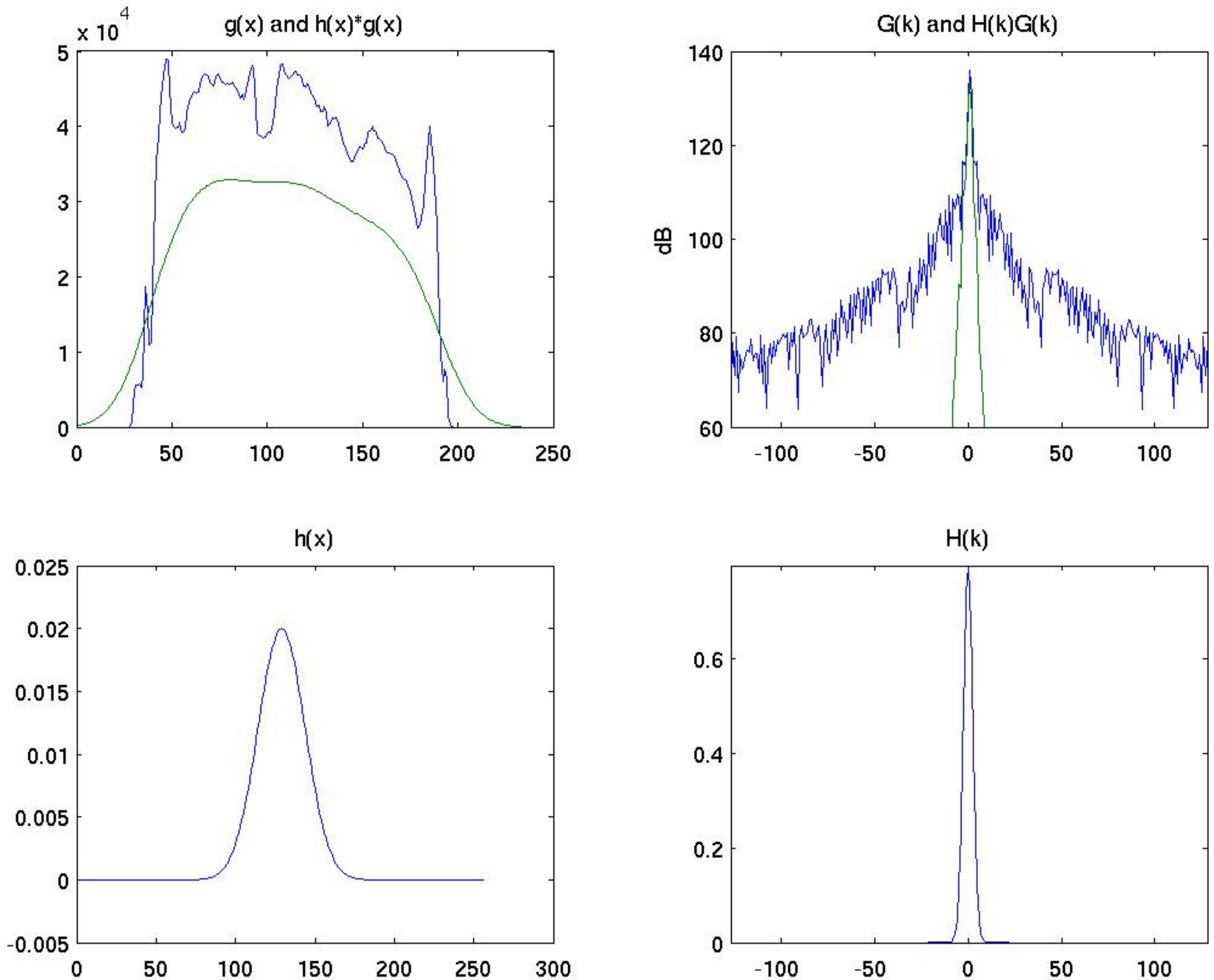
Fourier Domain System Response

Radon inverse filter kernel, $H(k) = |\,k\,|$ (high pass filter)

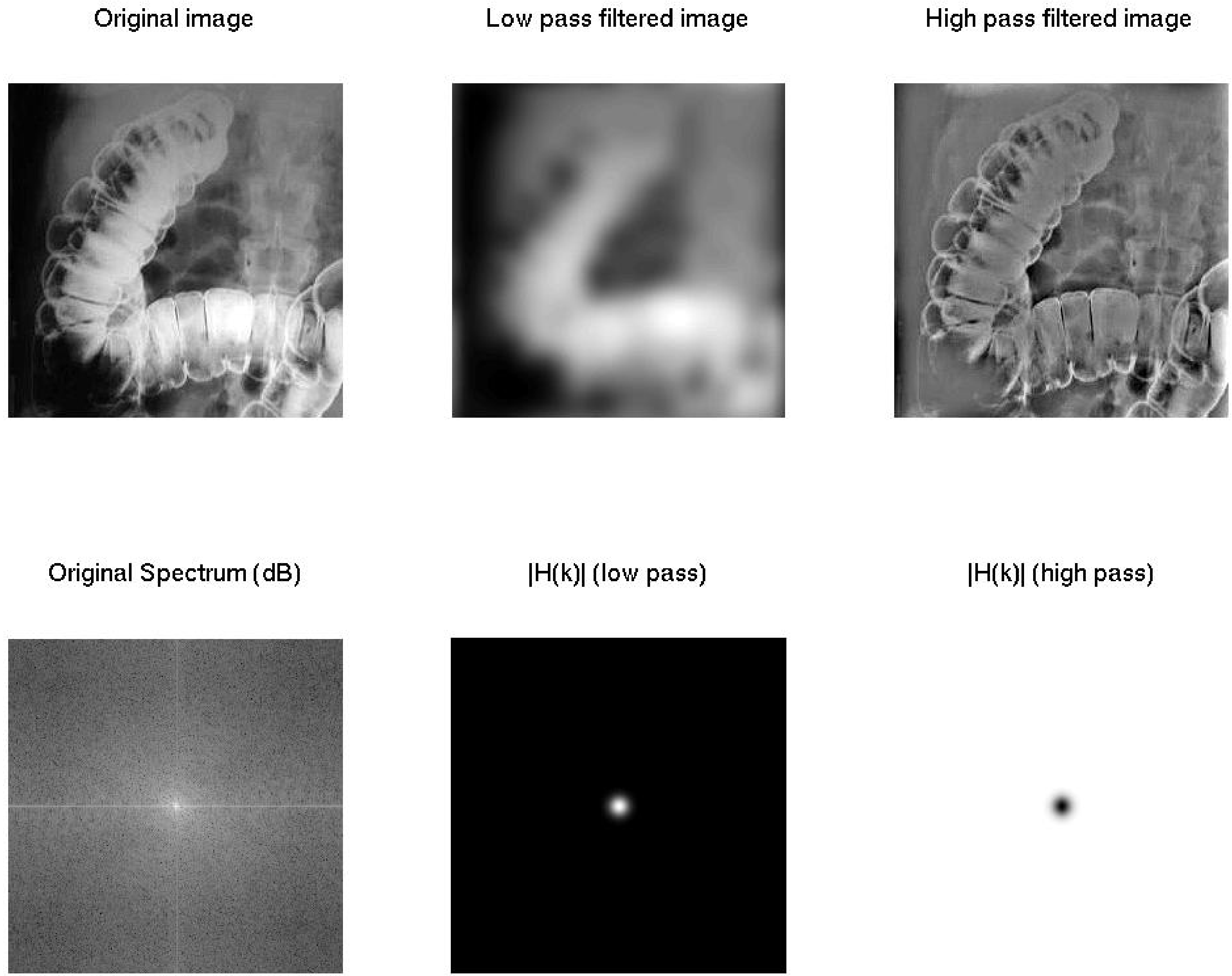


Fourier Domain System Response

Gaussian low-pass filter kernel, $H(k) = \exp\left(\frac{-k^2}{2\sigma^2}\right), \quad \sigma = 1.27$



Fourier Domain System Response 2D



Fourier Transform – FFT

The numerical implementation of the FT is discrete in x and k is referred to as Discrete Fourier Transform (DFT).

$$G[k] = \sum_{x=0}^{N-1} g[x] e^{-j2\pi kx/N}$$

$$g[x] = \frac{1}{N} \sum_{k=0}^{N-1} G[k] e^{j2\pi kx/N}$$

- A fast algorithm is available (FFT) to compute the DFT in only $N \log_2 N$ operations instead of N^2.
- Convolution with long filters is therefore often implemented using the FFT.
- FFT requires N to be a power of 2.

Fourier Transform – Filter with FFT

To filter signals with lengths not a power of 2:

- Pad zeros at the end up to next power of 2
- FFT
- multiply each frequency
- inverse FFT
- keep only fist L values.

Matlab $b(x,y) = h(x,y) * g(x,y)$ using fft (assumes h smaller than g):

```
L = size(g);
N = 2.^ceil(log2(L));
b = ifft2(fft2(g,N(1),N(2)).*fft2(h,N(1),N(2)));
if isreal([g;h]) b = real(b); end
b = b(1:L(1),1:L(2));
```

Demonstrate: `fftshift`, inverse filtering, separability.

Assignment FT: Generate image with randomized phase and original amplitude, and image with random amplitude and original phase.

Fourier Transform – Filter with FFT

Assignment HP filter in frequency domain:

Apply a high-pass filter to an image using an appropriate gain H(k) in in the Fourier domain. Select an image in which it makes sense to enhance edges. Send your image along with the program.

To get the frequency values correctly for the use of the fft2() function these lines of code may be helpful:

```
[Ny,Nx]=size(img);
[kx,ky]=meshgrid((0:Nx-1)-Nx/2,(0:Ny-1)-Ny/2);
kx = fftshift(kx);
ky = fftshift(ky);
```