Dominio de la frecuencia

Temario

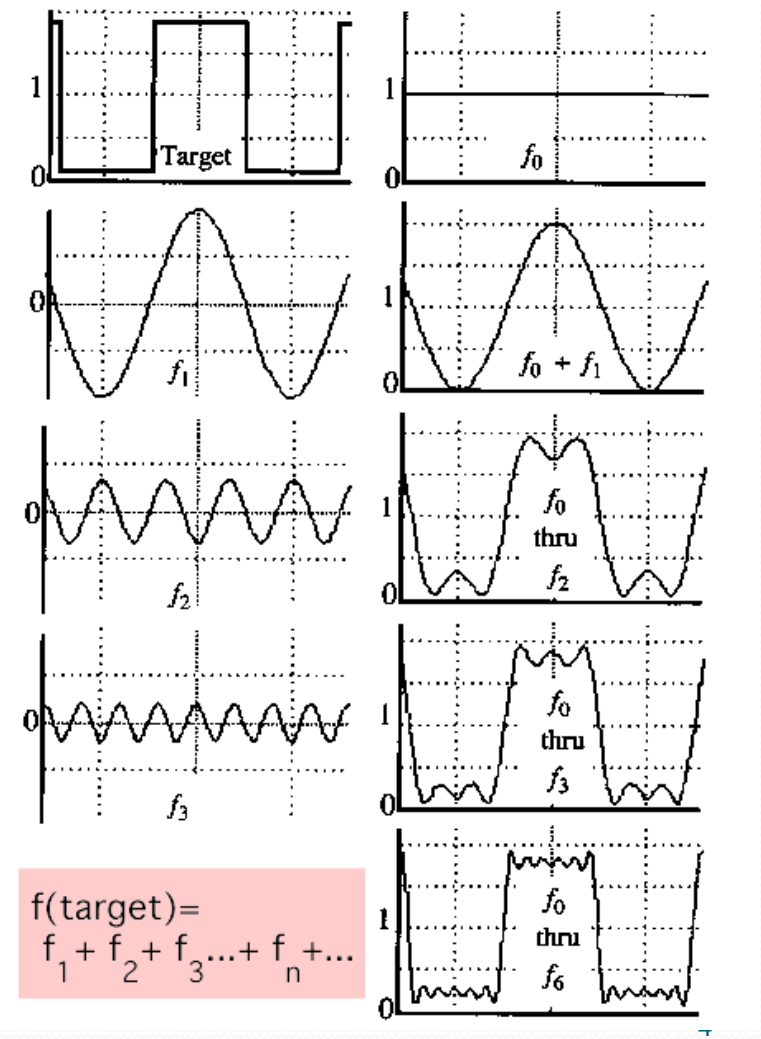
- Transformada de Fourier y dominio de la frecuencia.
- Filtrado de frecuencia.
- Otra mirada a las imágenes híbridas.
- Sampling (muestreo).

Pensando en términos de frecuencia

- Joseph Fourier (1768 – 1830):
- Cualquier función de una variable se puede reescribir como una suma ponderada de senos y cosenos de diferentes frecuencias.

Suma de senos

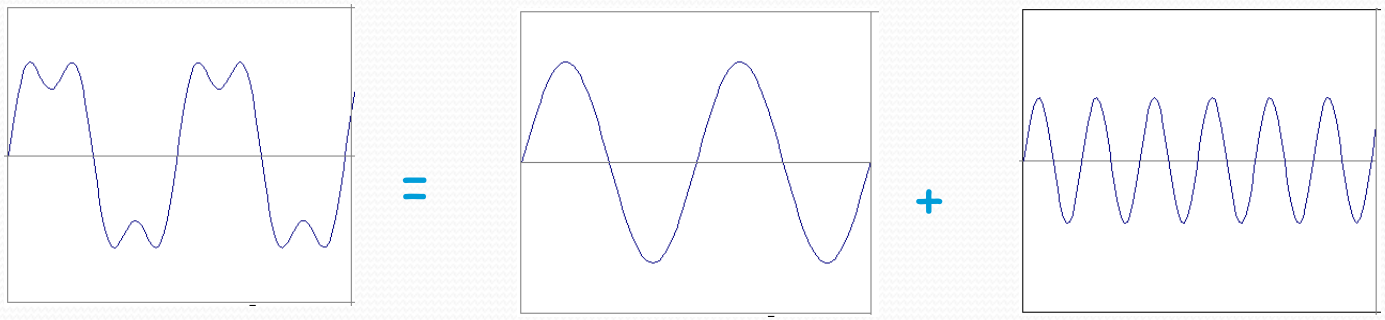
- Bloque de construcción:
- $A\sin(\omega x + \varphi)$
- Agregar suficientes de ellos para obtener cualquier señal $f(x)$ deseada.



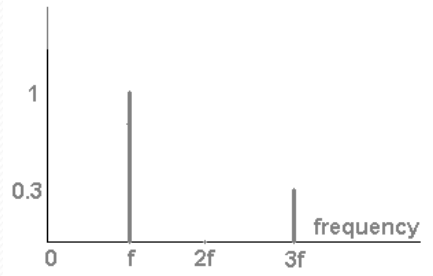
Fuente: Derek Hoiem (UIUC)

Espectro de frecuencias

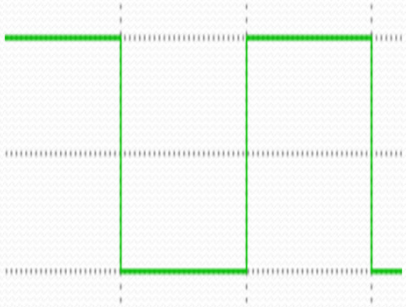
- Ejemplo: $g(t) = \sin(2\pi f t) + (1/3)\sin(2\pi(3f)t)$



Fuente: Derek Hoiem (UIUC)

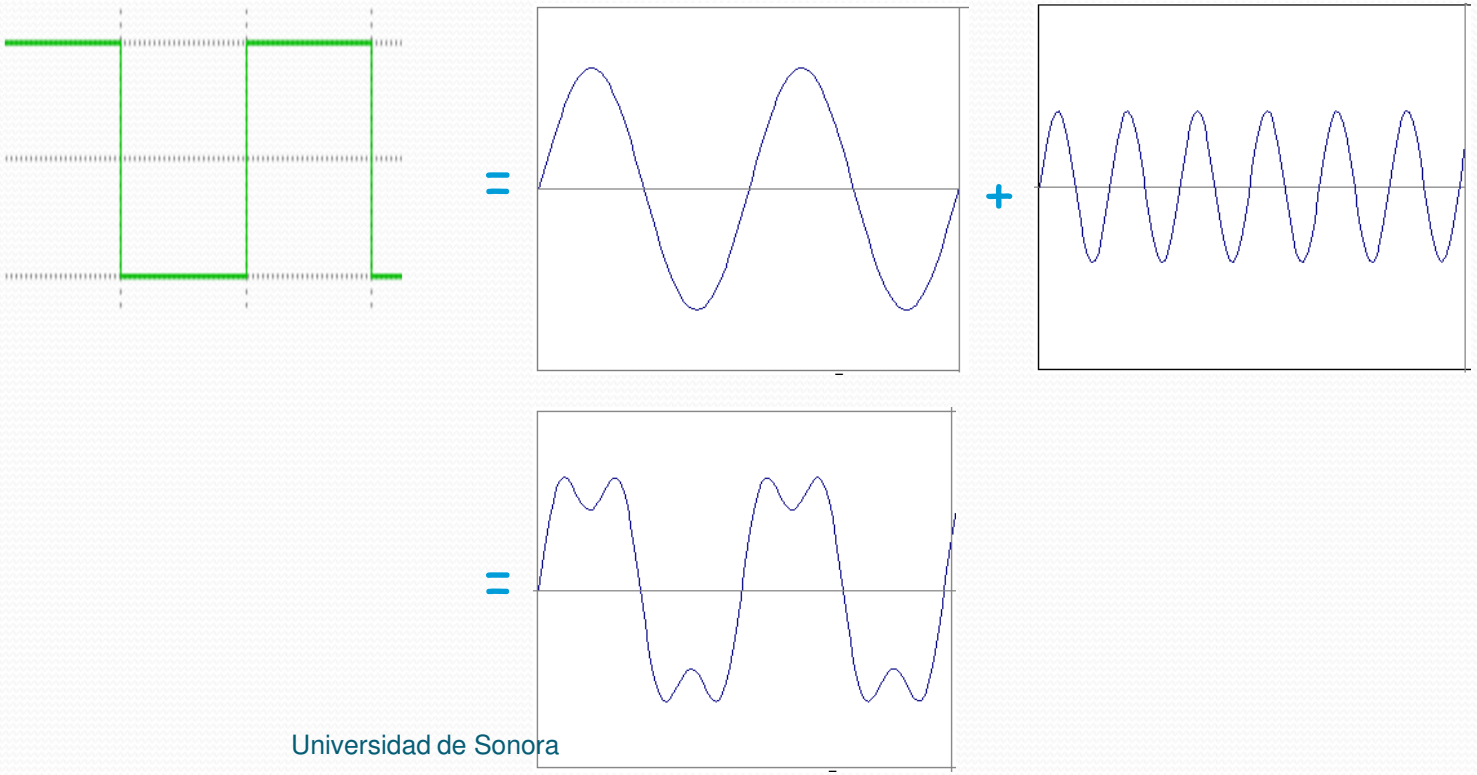


Espectro de frecuencias

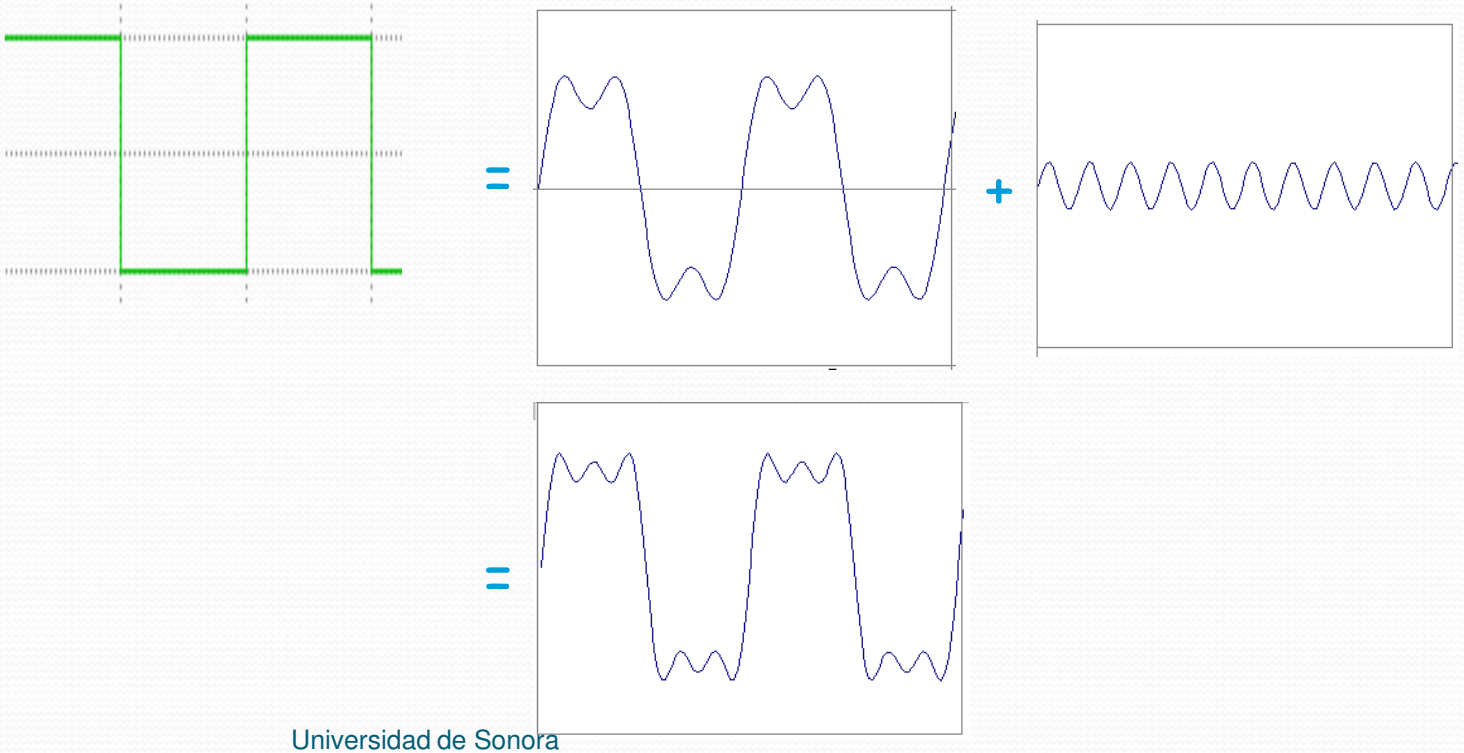


Fuente: Derek Hoiem (UIUC)

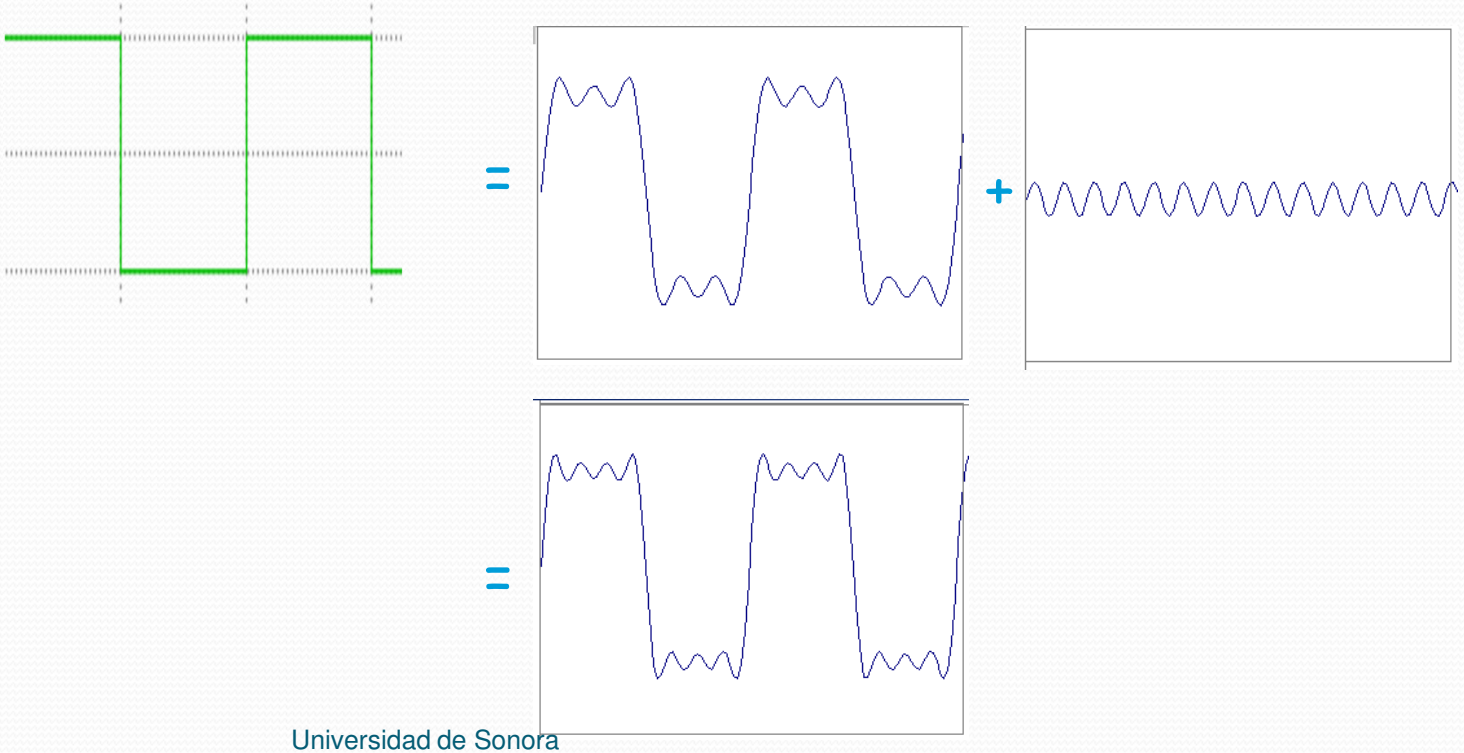
Espectro de frecuencias



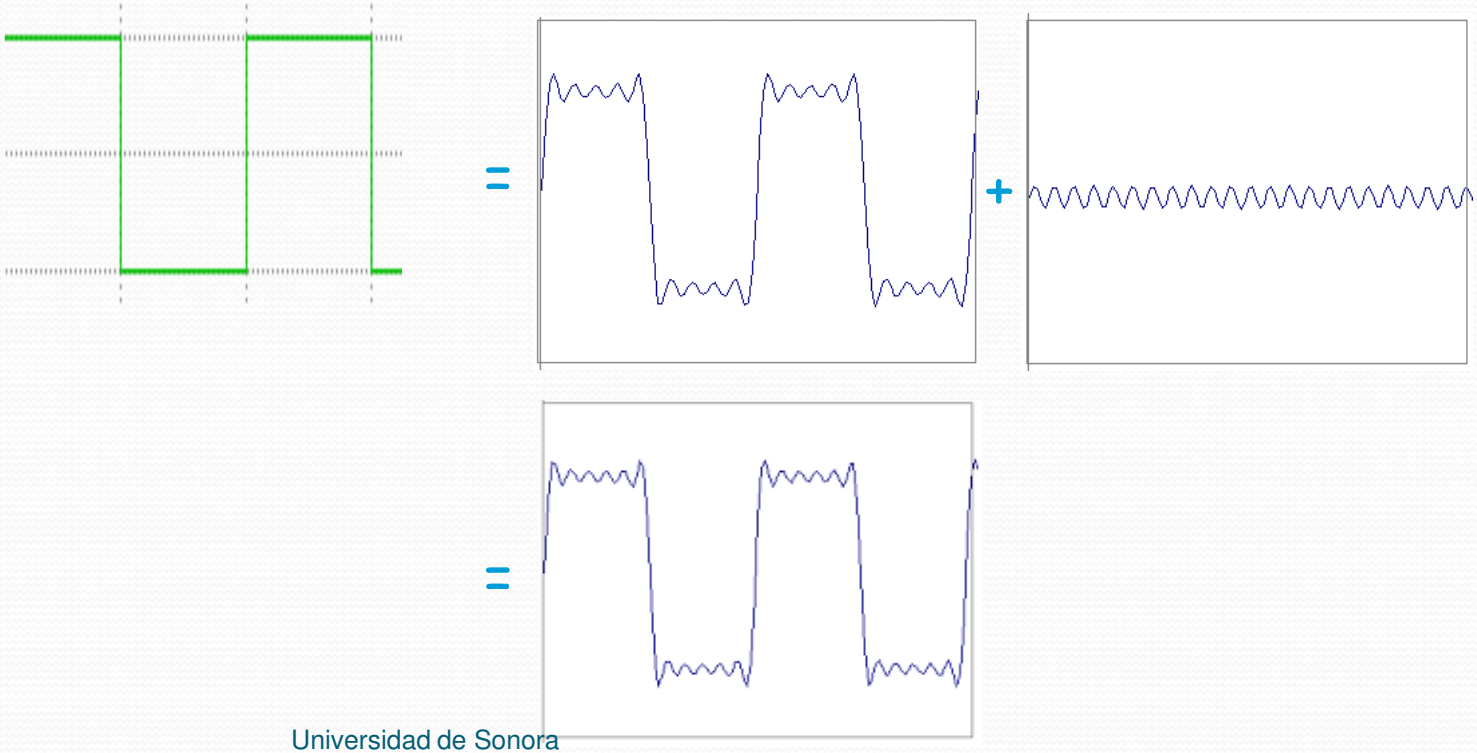
Espectro de frecuencias



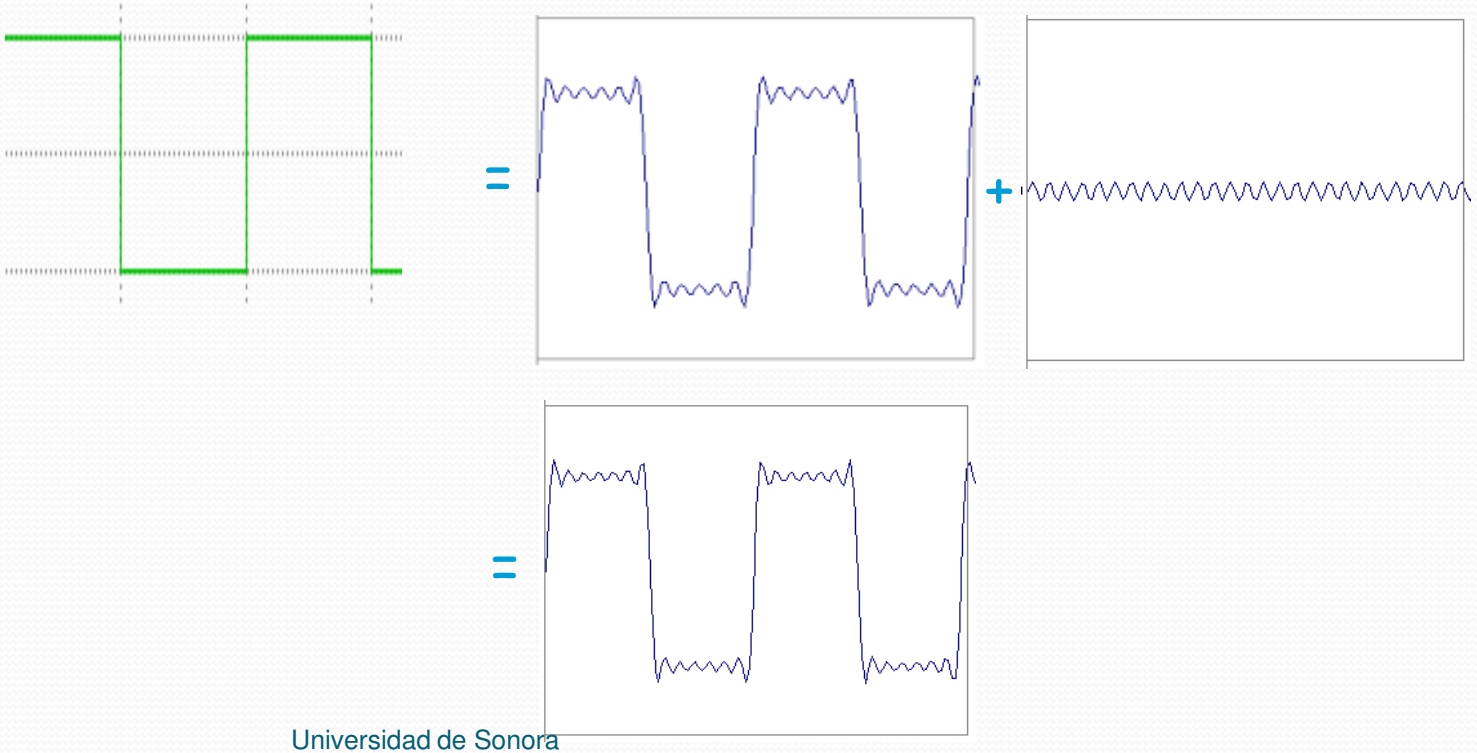
Espectro de frecuencias



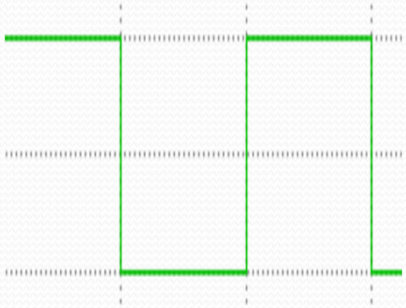
Espectro de frecuencias



Espectro de frecuencias

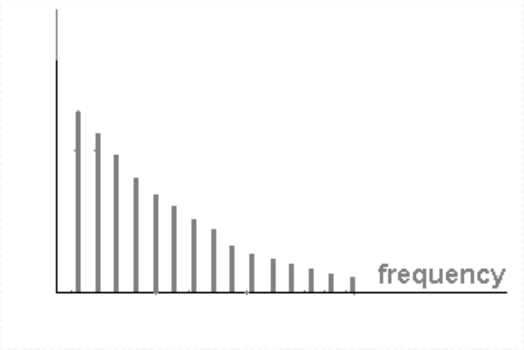


Espectro de frecuencias



=

$$A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi kt)$$



Otras señales

- El análisis de Fourier se puede aplicar a otro tipo de señales (p.e. música, imágenes).

Transformada de Fourier

- La transformada de Fourier almacena la magnitud y la fase en cada frecuencia.
- La magnitud es cuánto contribuye cada componente de frecuencia a esta señal.
- La fase codifica información espacial (indirectamente).
- Por conveniencia matemática, esto se expresa en números complejos.

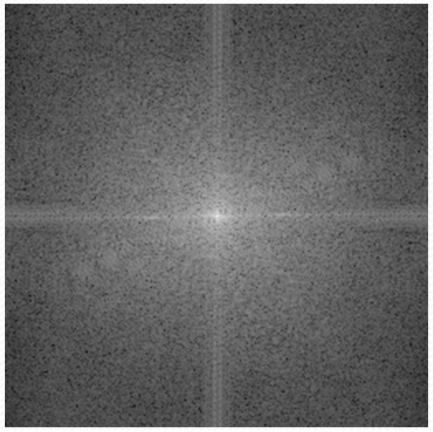
Transformada de Fourier

- Amplitud: $A = \pm\sqrt{R(\omega)^2 + I(\omega)^2}$
- Fase: $\varphi = \tan^{-1}\frac{I(\omega)}{R(\omega)}$
- Espectro de potencia: A^2
- Fórmula de Euler: $e^{inx} = \cos(nx) + i\sin(nx)$

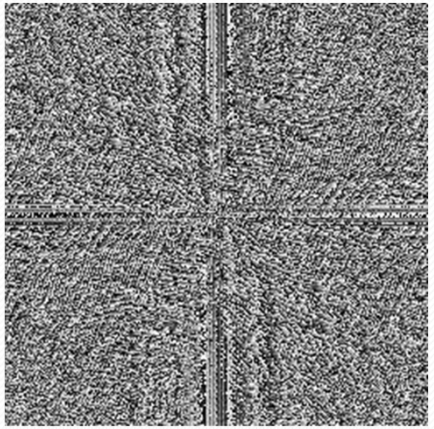
Análisis de Fourier en imágenes

Imagen

Amplitud

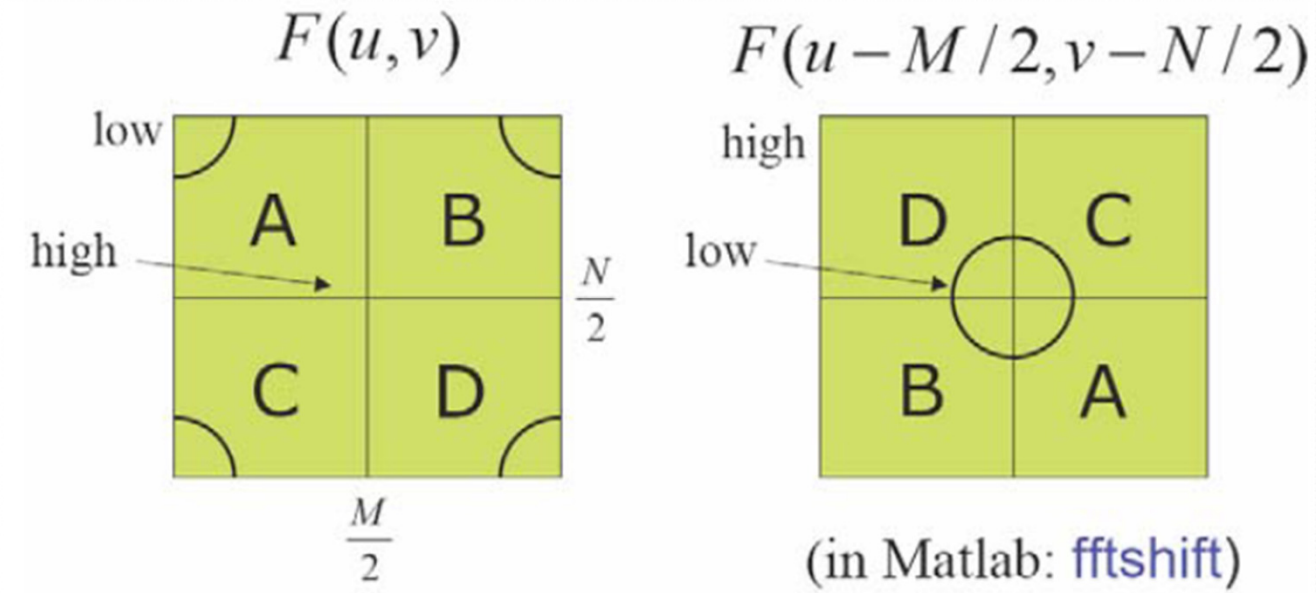


Fase



Fuente: https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm

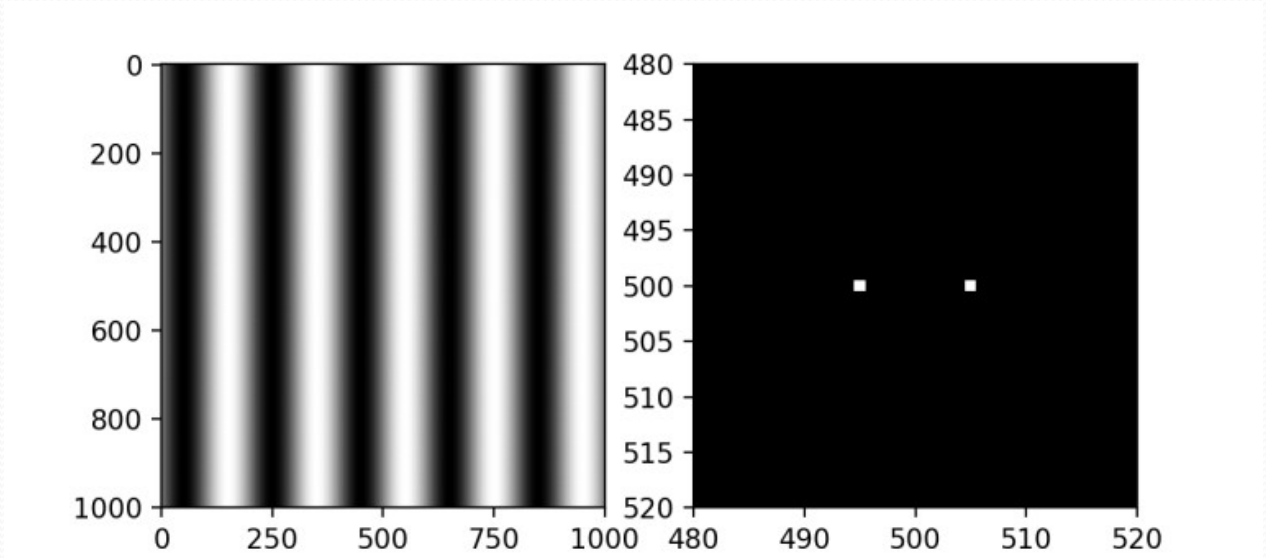
Desplegando la DFT 2-D



Fuente: https://staff-old.najah.edu/sites/default/files/Chapter4_Frequency_Enhancement.pdf

Análisis de Fourier en imágenes

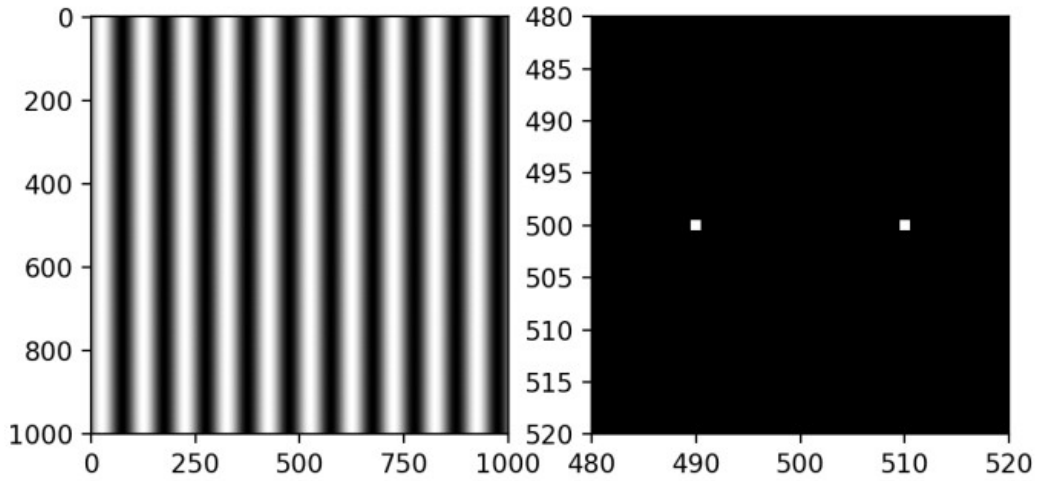
Wavelength = 200



Fuente: https://thepythoncodingbook.com/2021/08/30/2d-fourier-transform-in-python-and-fourier-synthesis-of-images

Análisis de Fourier en imágenes

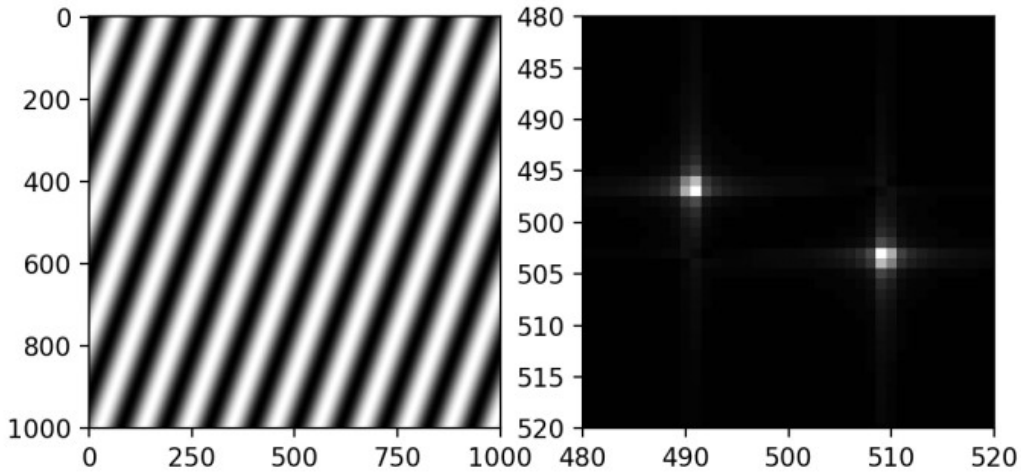
Wavelength = 100



Fuente: https://thepythoncodingbook.com/2021/08/30/2d-fourier-transform-in-python-and-fourier-synthesis-of-images

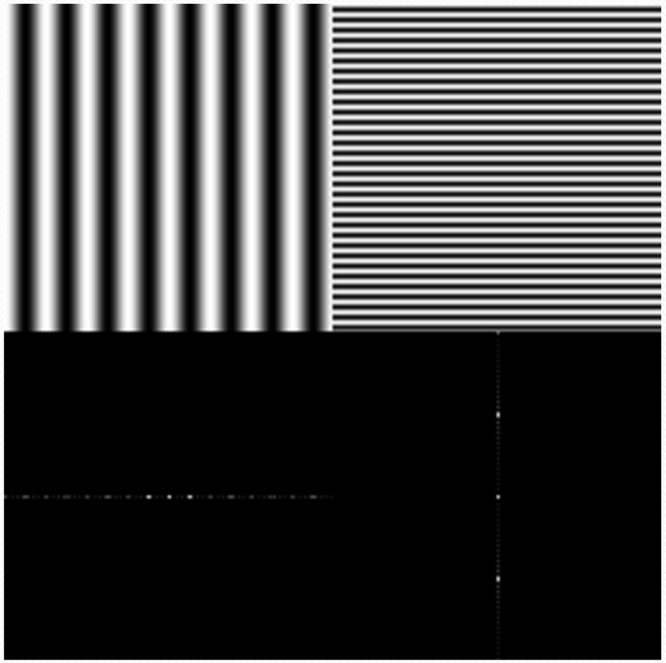
Análisis de Fourier en imágenes

Wavelength = 100



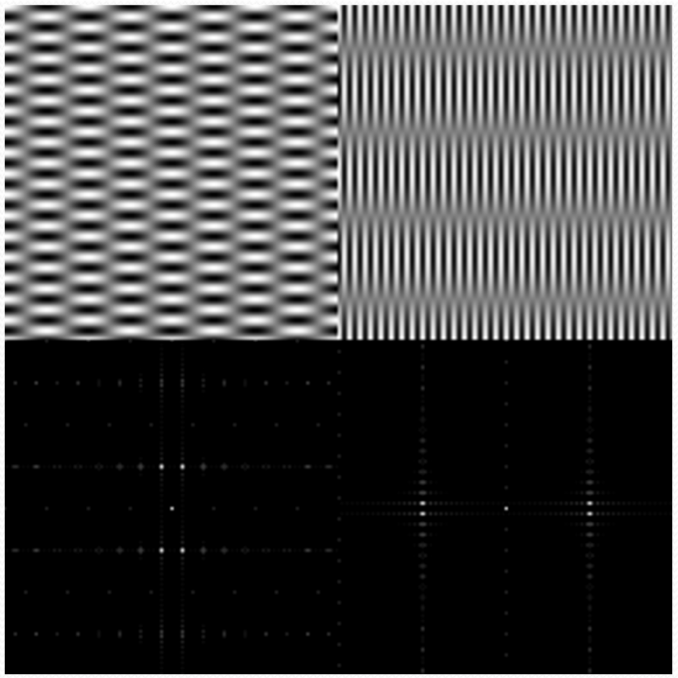
Fuente: https://thepythoncodingbook.com/2021/08/30/2d-fourier-transform-in-python-and-fourier-synthesis-of-images

Análisis de Fourier en imágenes



Fuente: https://www.cs.unm.edu/~brayer/vision/fourier.html

Análisis de Fourier en imágenes



Fuente: https://www.cs.unm.edu/~brayer/vision/fourier.html

Transformada de Fourier en 1D

- Transformada discreta en una dimensión:
- $H(k) = \frac{1}{N}\sum_{x=0}^{N-1} h(x) e^{-j\frac{2\pi kx}{N}}$
- $k = -\frac{N}{2} \ldots \frac{N}{2}$
- Aplicando la fórmula de Euler:
- $H(k) = \frac{1}{N}\sum_{x=0}^{N-1} h(x) \left[\cos\left(-\frac{2\pi kx}{N}\right) + i \sin\left(-\frac{2\pi kx}{N}\right)\right]$

Teorema de la convolución

- La transformada de Fourier de la convolución de dos funciones es el producto de sus transformadas de Fourier.
- $F[g \star h] = F[g]F[h]$
- La transformada inversa de Fourier del producto de dos transformadas de Fourier es la convolución de las dos transformadas inversas de Fourier.
- $F^{-1}[gh] = F^{-1}[g] \star F^{-1}[h]$
- La **convolución** en el dominio espacial es equivalente a la **multiplicación** en el dominio de la frecuencia.

Propiedades de la transformada de Fourier

- Linealidad $F[ax(t) + by(t)] = aF[x(t)] + bF[y(t)]$
- La transformada de Fourier de una señal real es simétrica con respecto al origen.
- La energía de la señal es la misma que la energía de su transformada de Fourier.

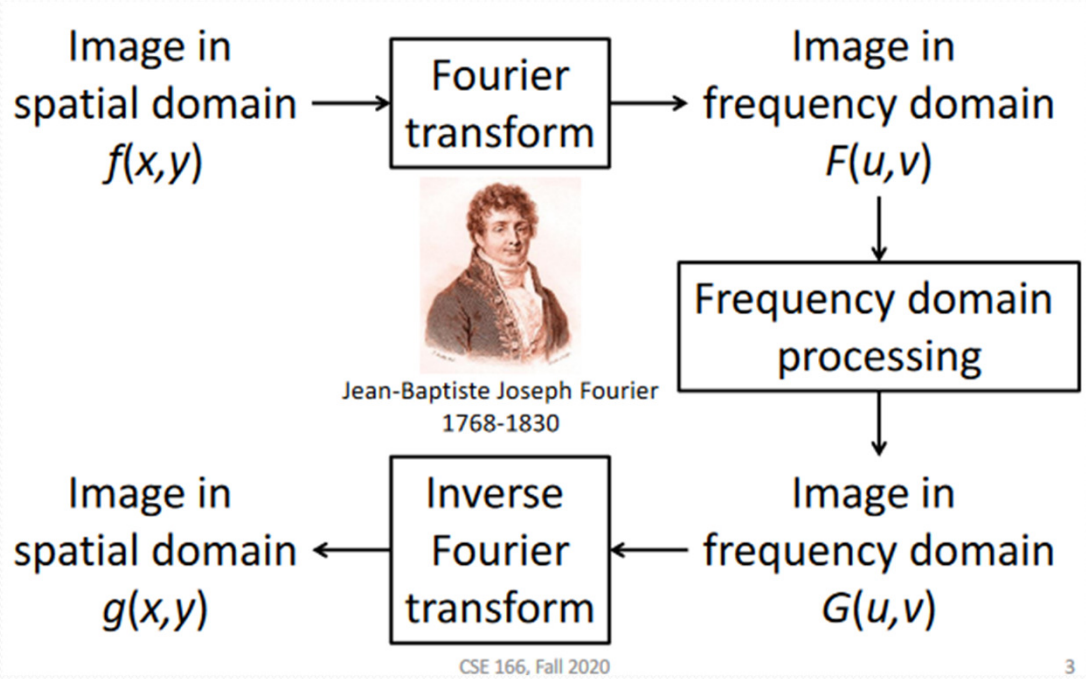
Transformada de Fourier en 2D

- Transformada discreta en dos dimensiones:
- $H(k_x, k_y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} h(x,y) e^{-j2\pi\left(\frac{k_x x}{M} + \frac{k_y y}{N}\right)}$
- Donde M y N son el ancho y el alto de la imagen.
- Todas las propiedades de la transformada de Fourier en 1D se aplican en 2D.

Comparación de filtrados

- Filtrado en el dominio espacial:
- $R = H \star I$
- R es la imagen resultado, H es el filtro, I es la imagen original.
- Filtrado en el dominio de la frecuencia:
- $G = D \times E$
- $R = \mathcal{F}^{-1}[G]$
- R es la imagen resultado, $D = \mathcal{F}[H]$, $E = \mathcal{F}[I]$.

Filtrado en frecuencia

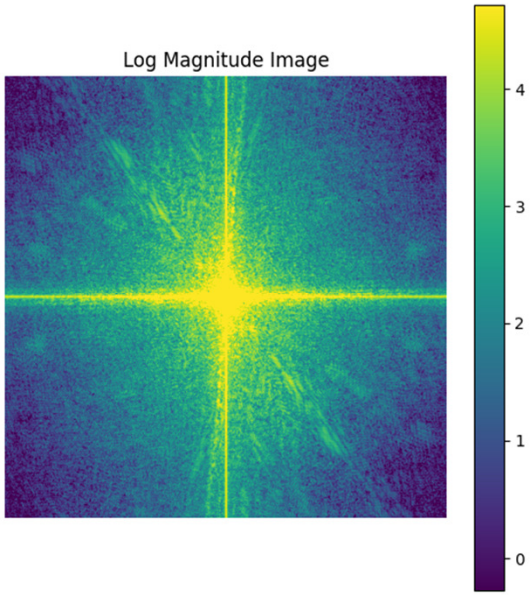


Fuente: https://cseweb.ucsd.edu/classes/fa20/cse166-a/lec7.pdf

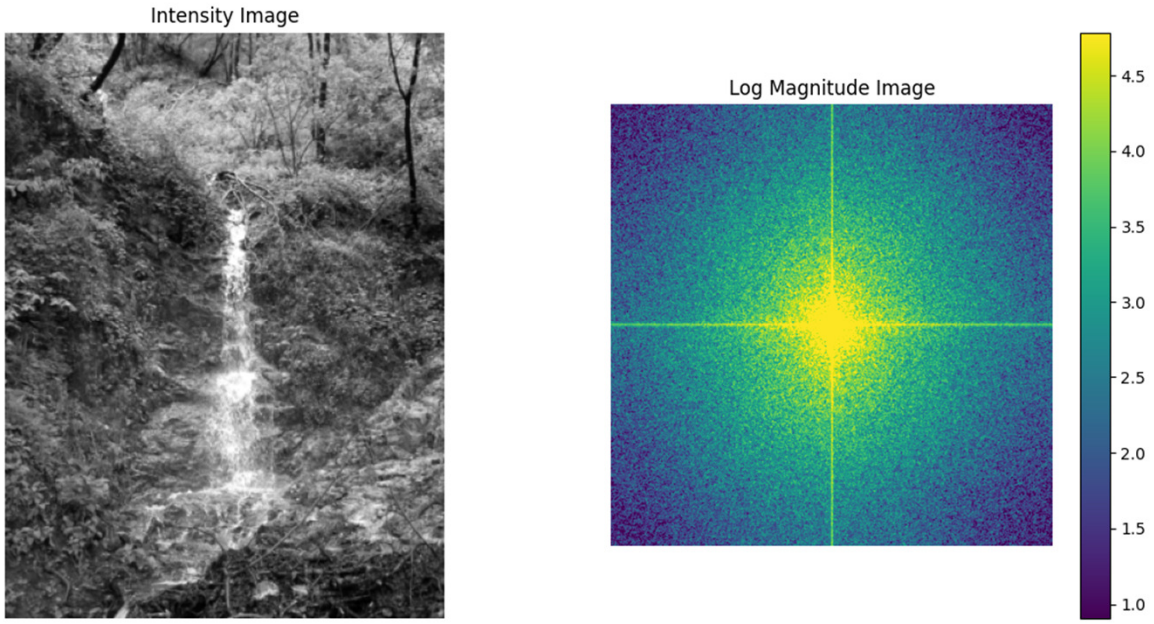
Comparación

- Todos los filtros de frecuencia se pueden implementar en el dominio espacial.
- Si existe un kernel para el efecto de filtro deseado, es computacionalmente menos costoso realizar el filtrado en el dominio espacial.
- El filtrado de frecuencia es apropiado si no se puede encontrar un kernel sencillo en el dominio espacial y también puede ser más eficiente.
- Ver https://homepages.inf.ed.ac.uk/rbf/HIPR2/freqfilt.htm

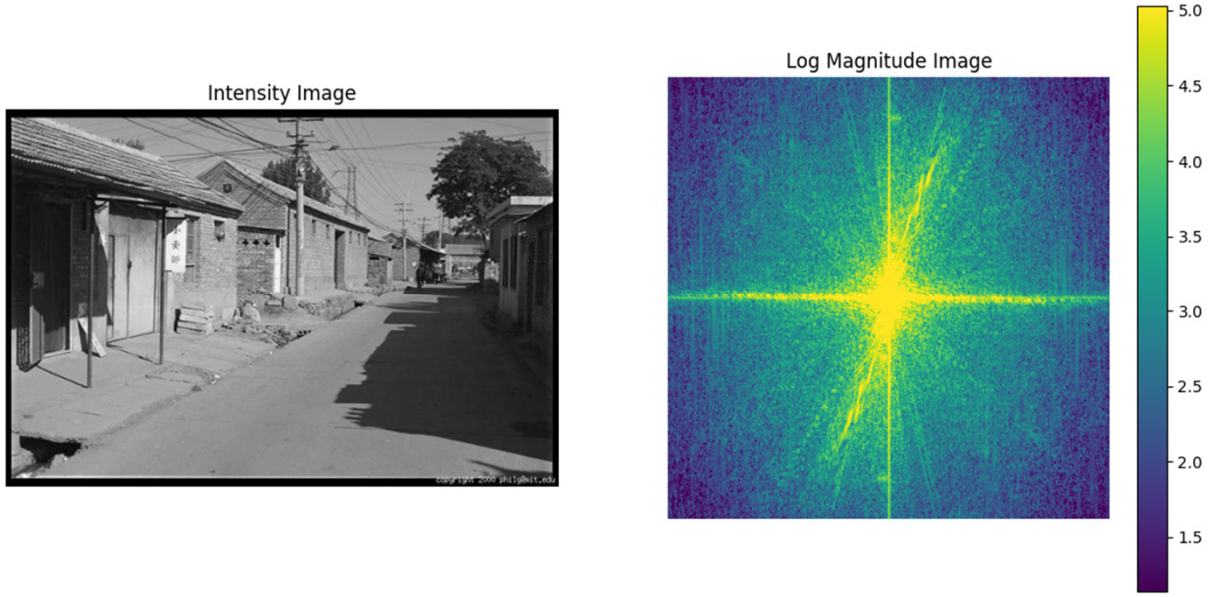
Ejemplo de imagen de Fourier



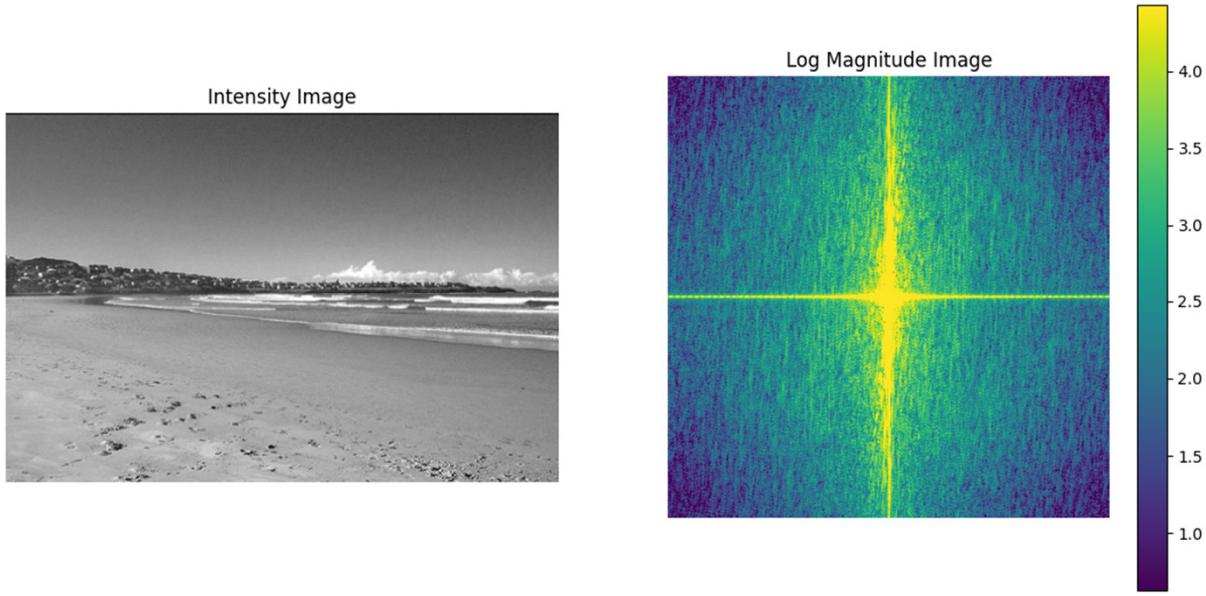
Ejemplo de imagen de Fourier



Ejemplo de imagen de Fourier



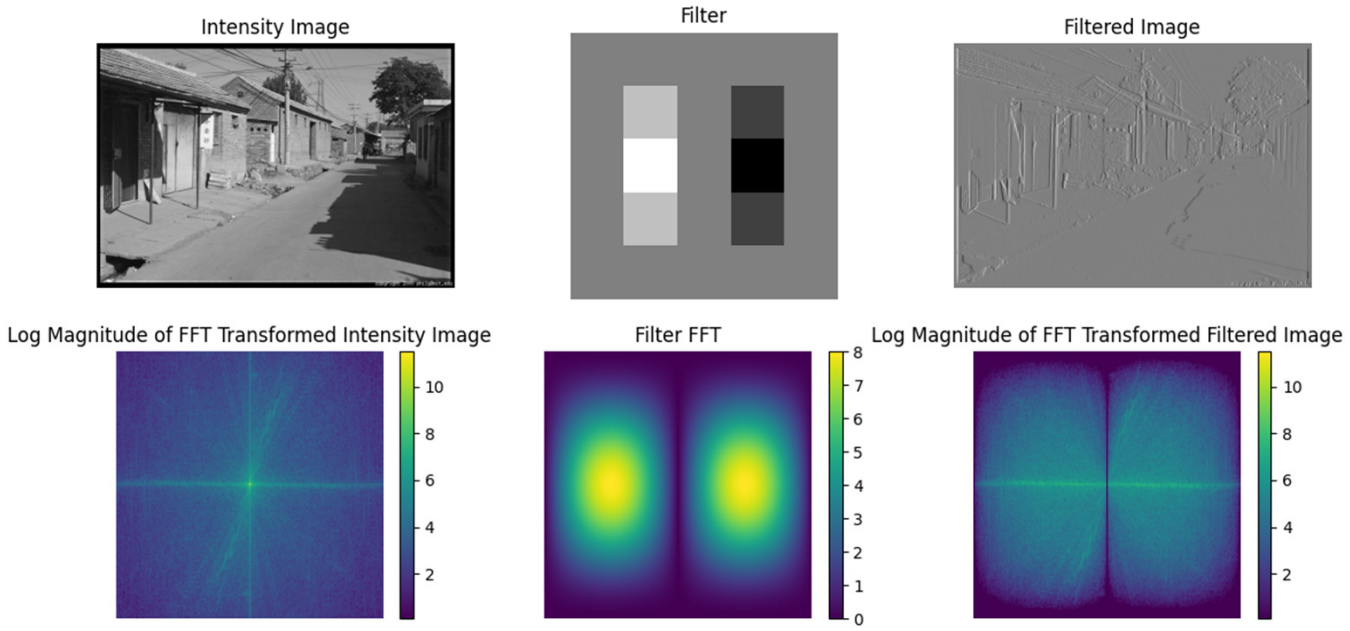
Ejemplo de imagen de Fourier



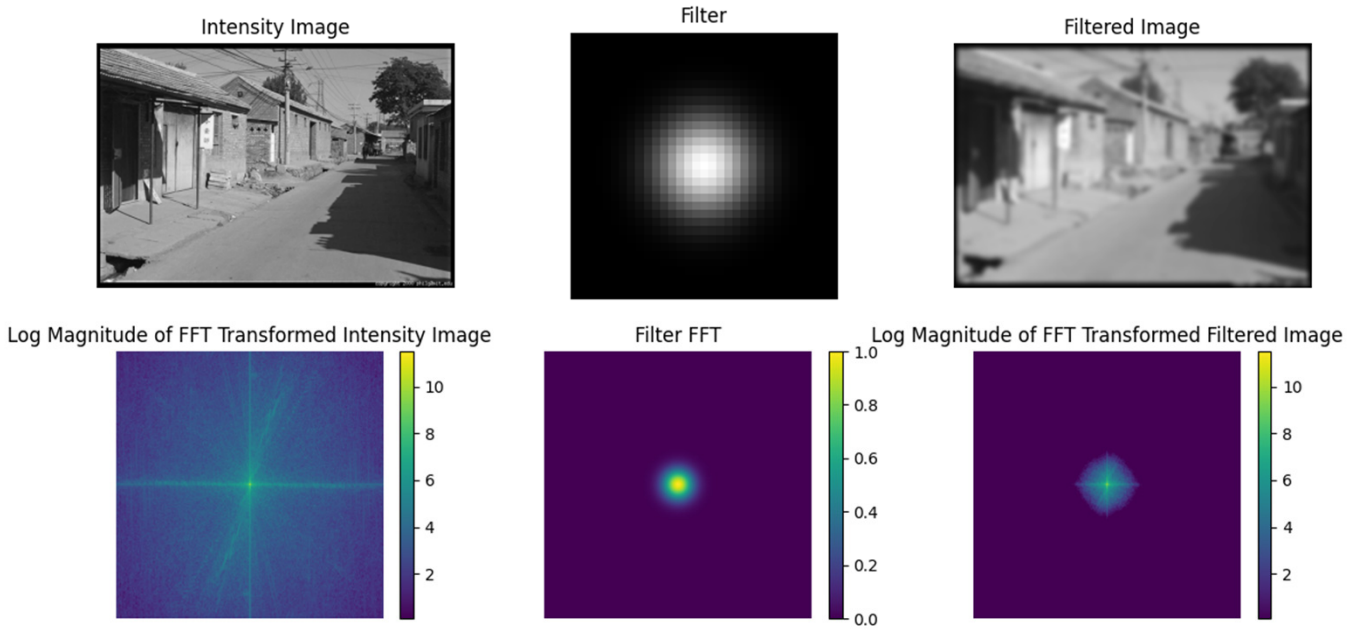
Ejemplos de filtrado

- Sobel
- Gaussiano
- Laplaciano del gaussiano (LoG)
- Caja

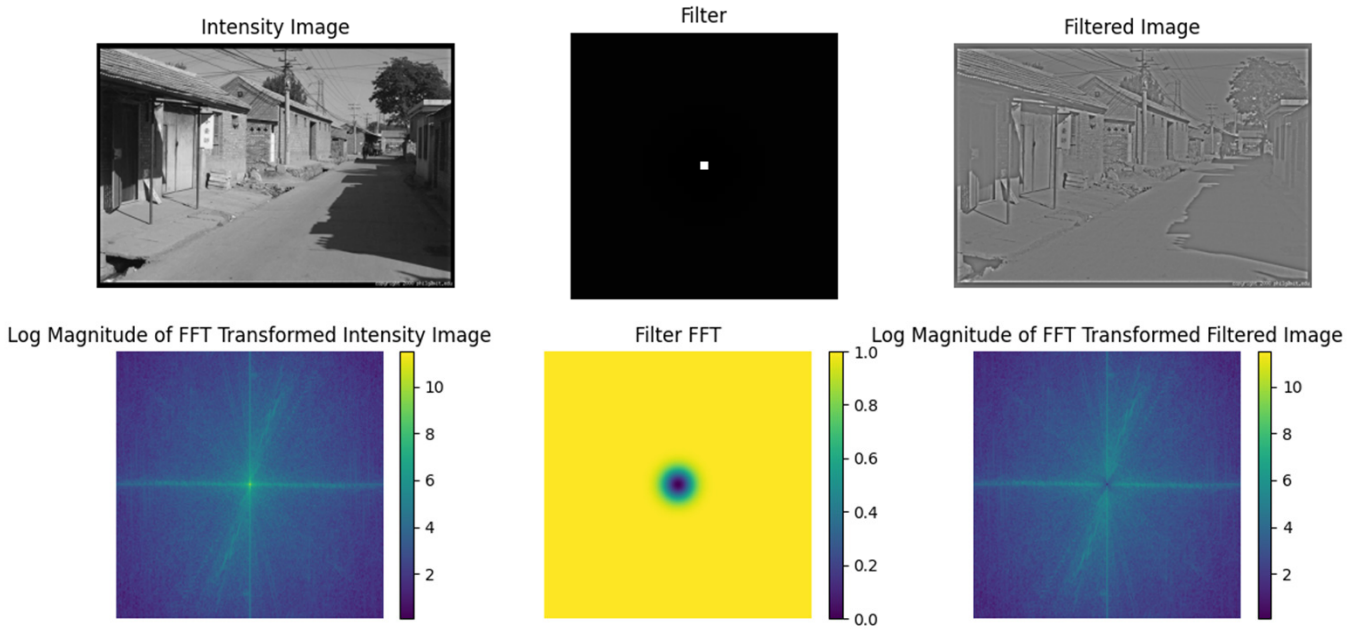
Filtro de Sobel



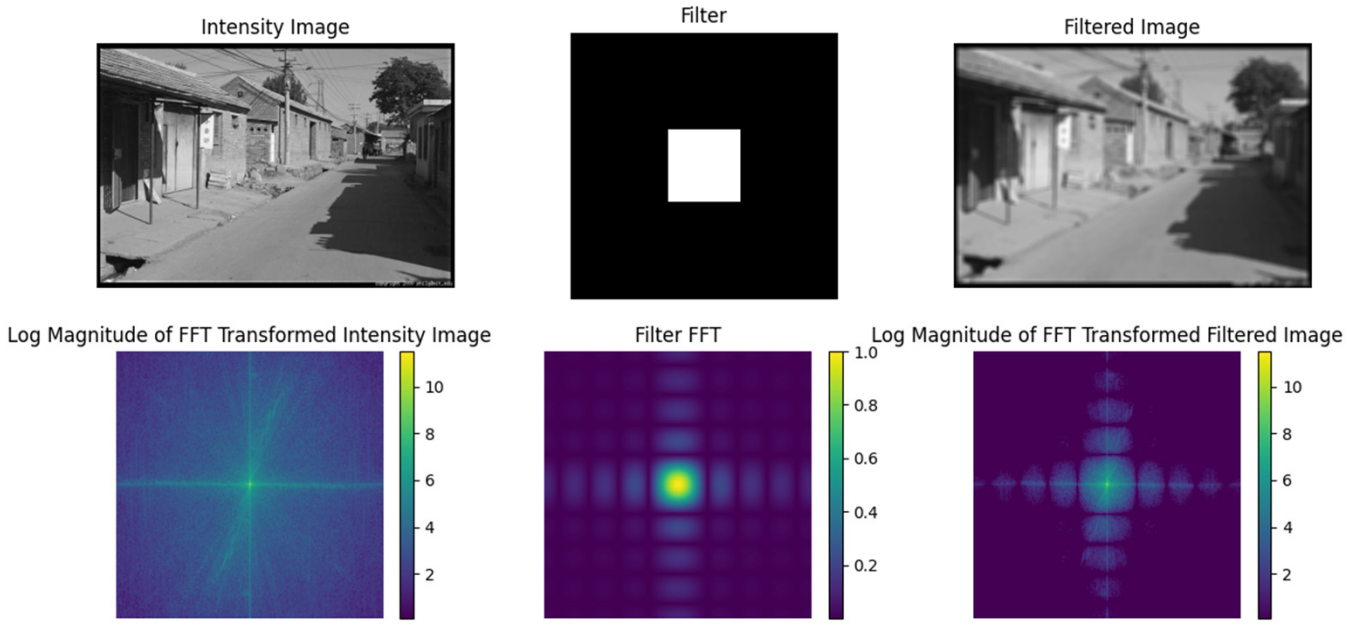
Filtro gaussiano



Filtro LoG


Intensity Image
Filter
Filtered Image
Log Magnitude of FFT Transformed Intensity Image
Filter FFT
Log Magnitude of FFT Transformed Filtered Image
10
8
6
4
2
1.0
0.8
0.6
0.4
0.2
0.0

Filtro de caja



FFT en Python

```
# https://courses.engr.illinois.edu/cs445/fa2023/
def fft_image(im):
    """
    im: H x W floating point numpy ndarray representing image in grayscale
    """
    fft_size = 1024  # should be order of 2 (for speed) and include padding
    return np.fft.fft2(im, (fft_size, fft_size))
```

Filtrado en Python 1/2

```
# https://courses.engr.illinois.edu/cs445/fa2023/
def filter_image(im, fil):
    """
    im: H x W floating point numpy ndarray representing image in grayscale
    fil: M x M floating point numpy ndarray representing 2D filter
    """
    h, w = im.shape
    hs = fil.shape[0] // 2                          # half of filter size
    fft_size = 1024                 # should be order of 2 (for speed) and include padding
    im_fft = np.fft.fft2(im, (fft_size, fft_size))   # 1) fft im with padding
```

Filtrado en Python 2 / 2

```
    fil_fft = np.fft.fft2(fil, (fft_size, fft_size))    # 2) fft fil, pad to same size as
image
    im_fil_fft = im_fft * fil_fft                       # 3) multiply fft images
    im_fil = np.fft.ifft2(im_fil_fft)                   # 4) inverse fft2
    im_fil = im_fil[hs:hs + h, hs:hs + w]               # 5) remove padding
    im_fil = np.real(im_fil)                            # 6) extract out real part
    return im_fil
```

Desplegar una imagen FFT 1 / 2

```
# https://courses.engr.illinois.edu/cs445/fa2023/
def display_frequency_image(frequency_image):
    """
    frequency_image: H x W floating point numpy ndarray representing
                     image after FFT in grayscale
    """
    log_amplitude_image = _to_log_amplitude(frequency_image)
    log_amplitude_image = filter_image(log_amplitude_image,
gaussian_kernel2d(3, 0.5))
```

Desplegar una imagen FFT 2 / 2

```
fig = plt.figure(figsize=(15, 10))
sv = np.sort(log_amplitude_image.flatten())
vmin = sv[int(round(len(sv) * 0.02))]
vmax = sv[int(round(len(sv) * 0.98))]
plt.imshow(log_amplitude_image, vmin=vmin, vmax=vmax)
plt.colorbar()
plt.axis('off')
plt.show()
```

2 preguntas 2

- ¿Cuál tiene más información, la fase o la magnitud?
- ¿Qué pasa si se toma la fase de una imagen y se combina con la magnitud de otra imagen?

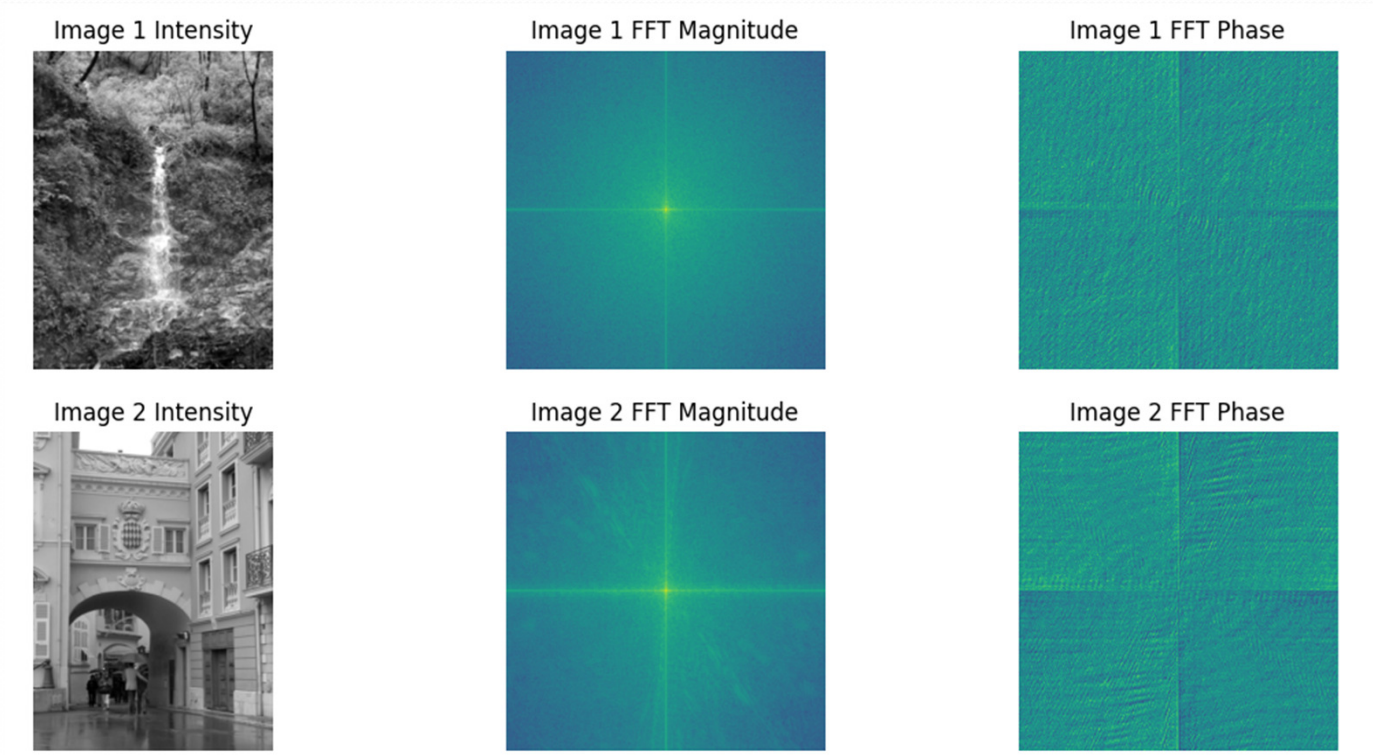
Paso 1

```
# Compute fft and decompose to magnitude and phase
im1_fft = fft_image(im1)
im2_fft = fft_image(im2)
im1_mag = np.abs(im1_fft)
im1_phase = np.angle(im1_fft)
im2_mag = np.abs(im2_fft)
im2_phase = np.angle(im2_fft)
```

Paso 2

```
# Combine mag and phase from different images and compute inverse FFT
mag1_phase2 = np.fft.ifft2(im1_mag * (np.cos(im2_phase) +
                          1j * np.sin(im2_phase)))
mag2_phase1 = np.fft.ifft2(im2_mag * (np.cos(im1_phase) +
                          1j * np.sin(im1_phase)))
```

Magnitud y fase



Magnitud y fase

Image 1

Image 2

Image 2 Mag + Image 1 Phase

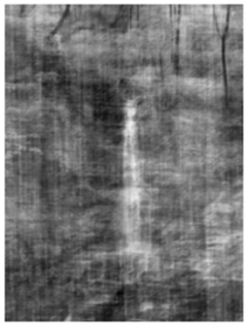


Image 1 Mag + Image 2 Phase

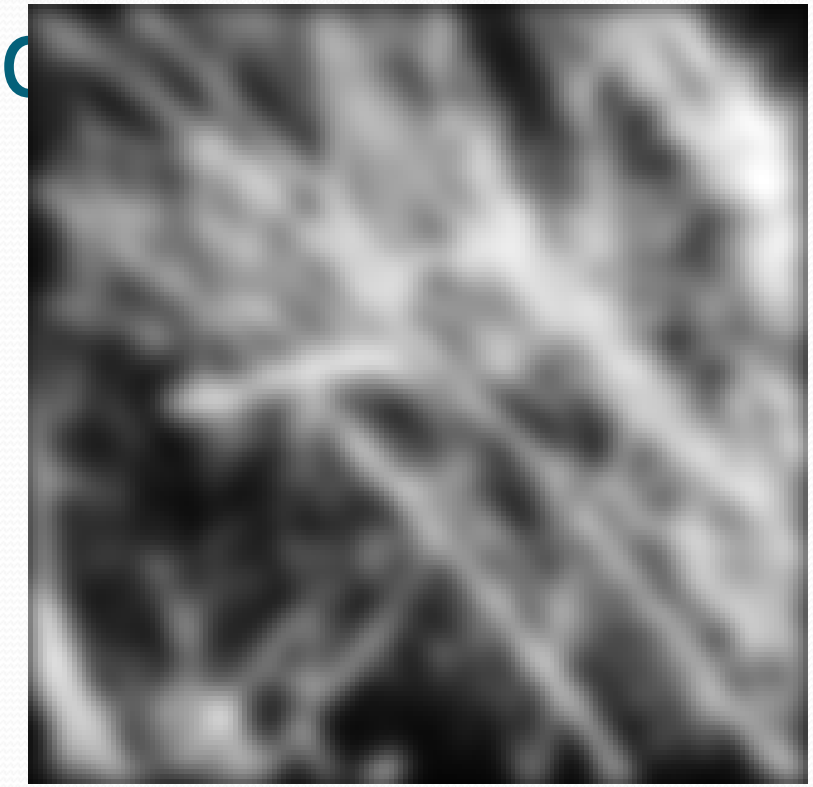
Filtrado

- ¿Por qué el gaussiano genera una imagen suave y el filtro de caja introduce artefactos?

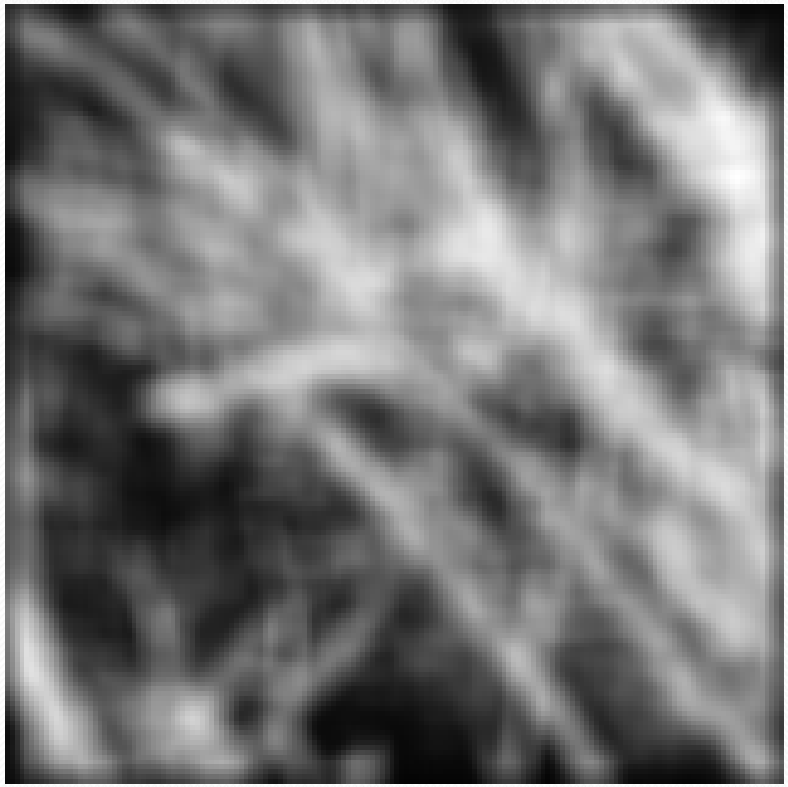
- Imagen original

Filtrac

Gaussian 

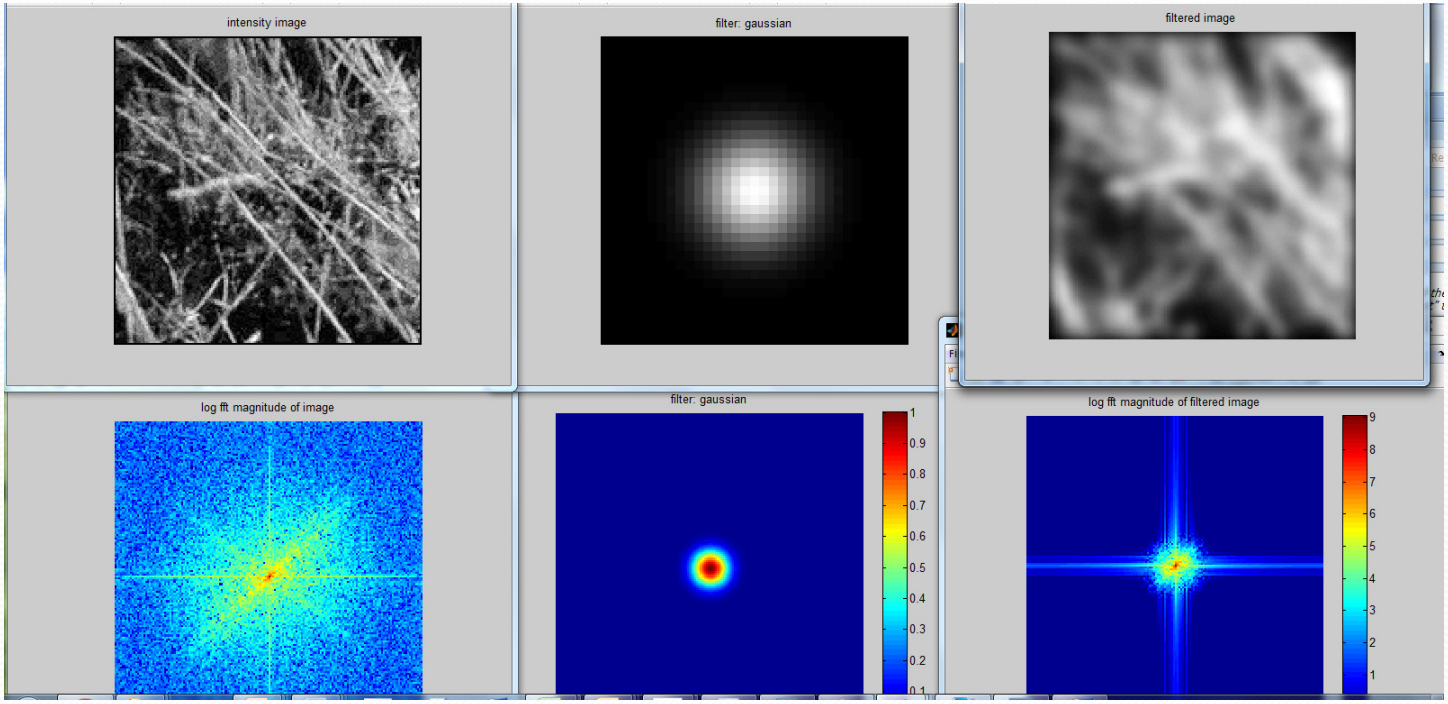


Box filter 



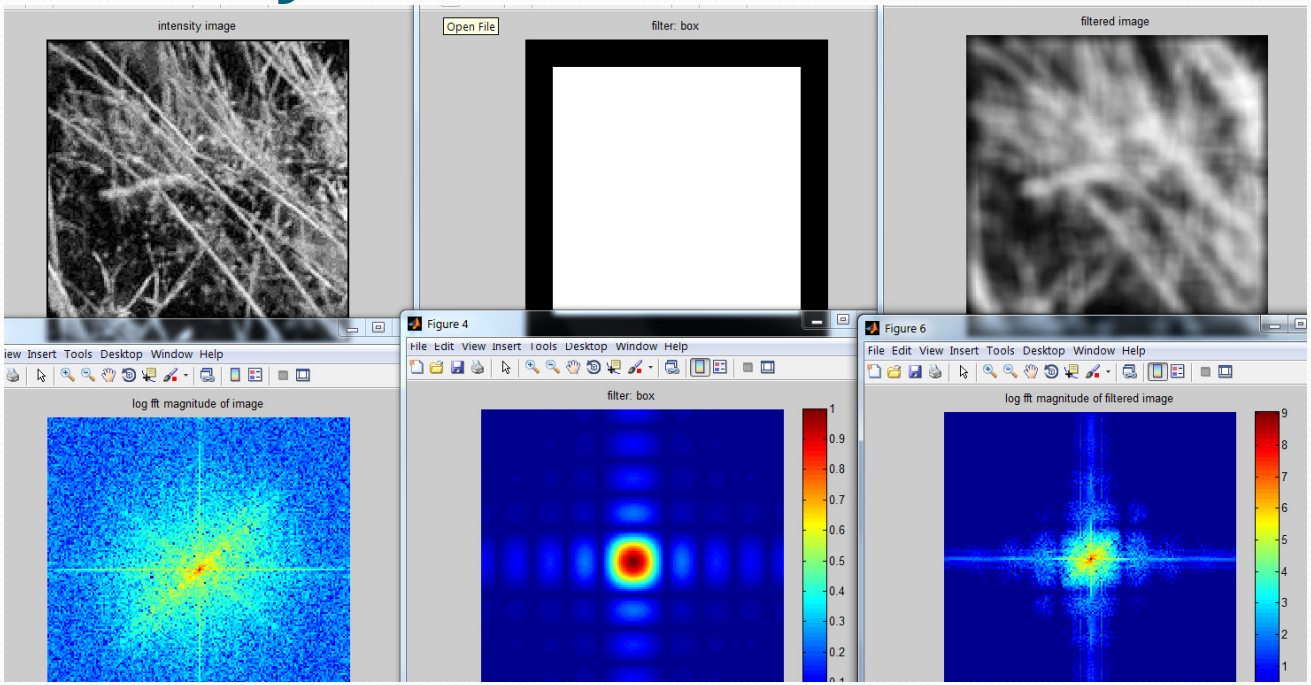
Fuente: Derek Hoiem (UIUC)

Gaussiano



Fuente: Derek Hoiem (UIUC)

Filtro de caja



Fuente: Derek Hoiem (UIUC)

Sampling

- Para procesamiento digital, una función de imagen $f(x, y)$ se debe digitalizar tanto espacialmente como en amplitud.
- Este proceso de digitalización implica dos procesos principales llamados:
- Sampling: es la digitalización del valor de las coordenadas.
- Cuantización: es la digitalización del valor de amplitud de la función.

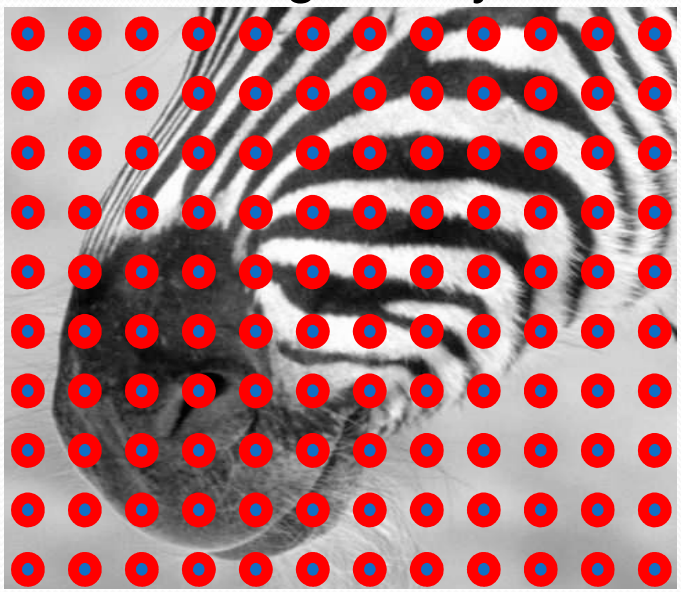
Subsampling

- Reducir la resolución de la imagen.

Fuente: http://www.flickr.com/photos/igorms/136916757/

Subsampling en un factor de 2

- Desechar cada dos renglones y columnas para crear una imagen de tamaño 1/2.



Fuente: Derek Hoiem (UIUC)

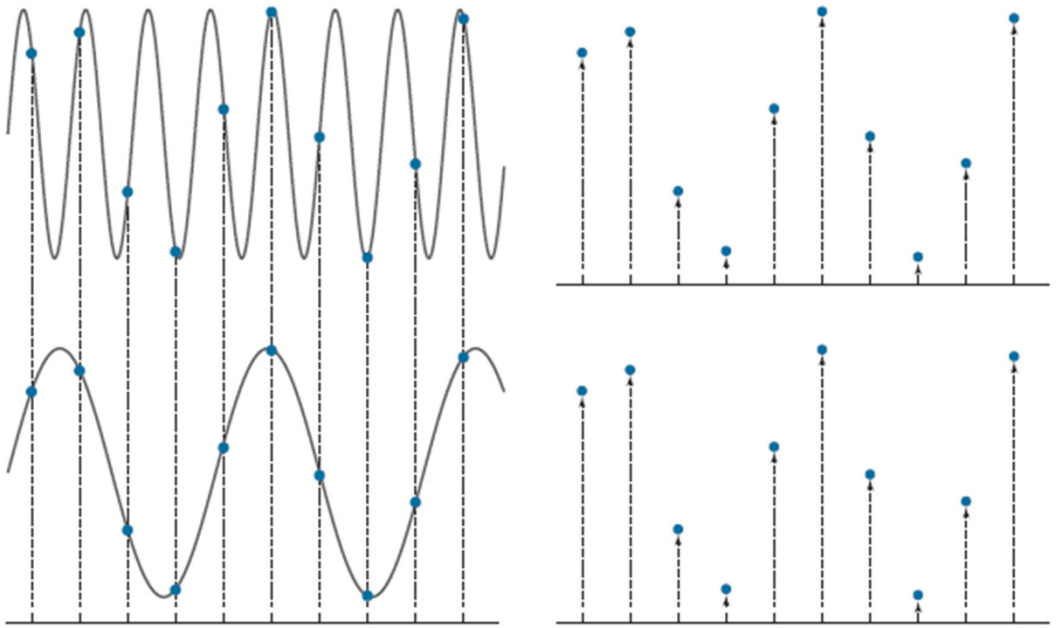
Alias

- La palabra alias significa “una identidad falsa”.
- En procesamiento de señales, aliasing se refiere a fenómenos de muestreo que hacen que diferentes señales se vuelvan indistinguibles entre sí después del muestreo, es decir, que una señal se “disfrace” de otra.

Alias

a b
c d

FIGURE 4.9
The functions in (a) and (c) are totally different, but their digitized versions in (b) and (d) are identical. Aliasing occurs when the samples of two or more functions coincide, but the functions are different elsewhere.



Fuente: DIP4 – GW, p. 221

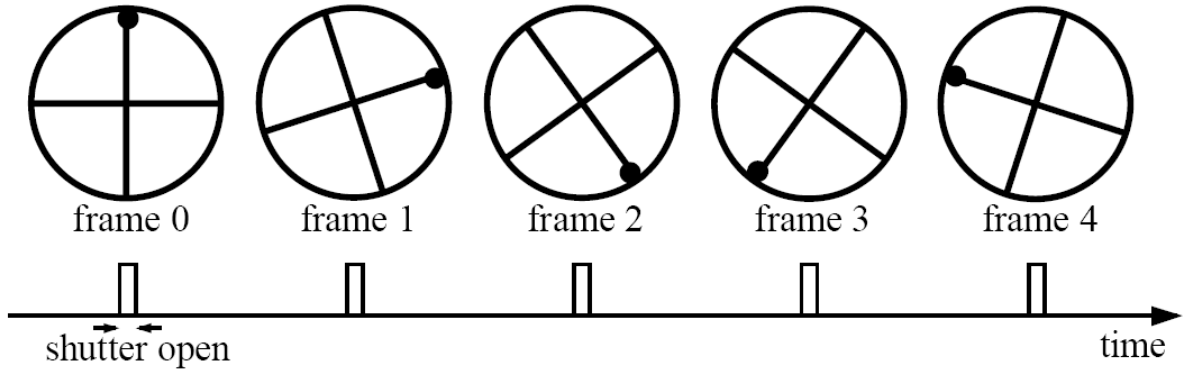
Alias

- El subsampling puede ser peligroso.
- Errores característicos:
- Las ruedas de los carros ruedan en sentido contrario en las películas.
- Los tableros de ajedrez se desintegran en el horizonte.
- Las camisas a rayas lucen raras en la televisión en color.

Alias en video

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what's happening.

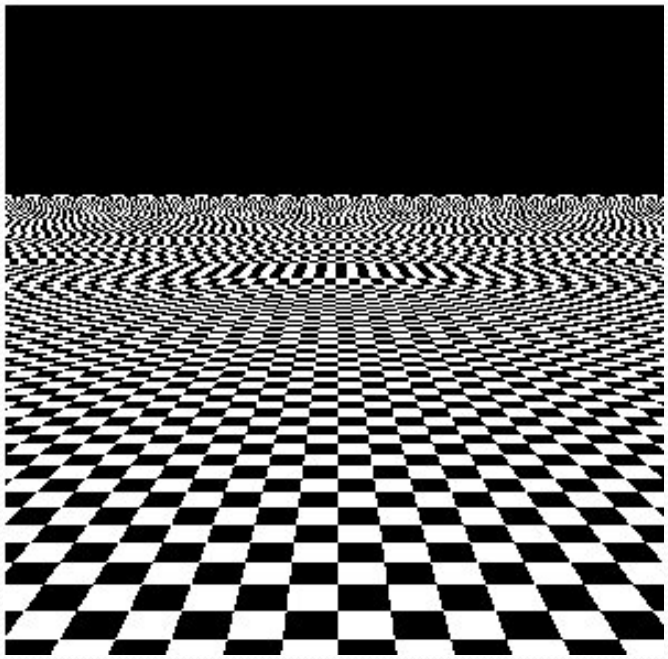
If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



Slide by Steve Seitz

Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

Alias en imagen



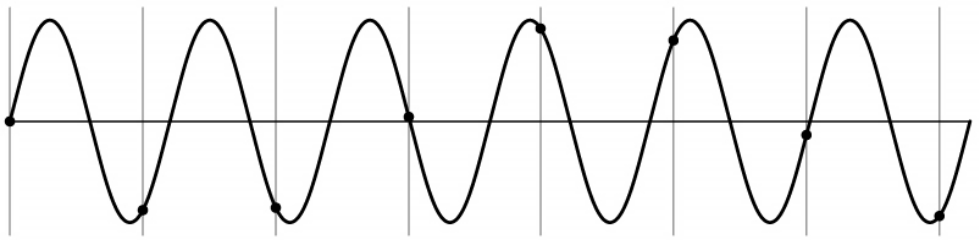
Fuente: https://textureingraphics.wordpress.com/what-is-texture-mapping/anti-aliasing-problem-and-mipmapping/

Patrones de Moiré por aliasing

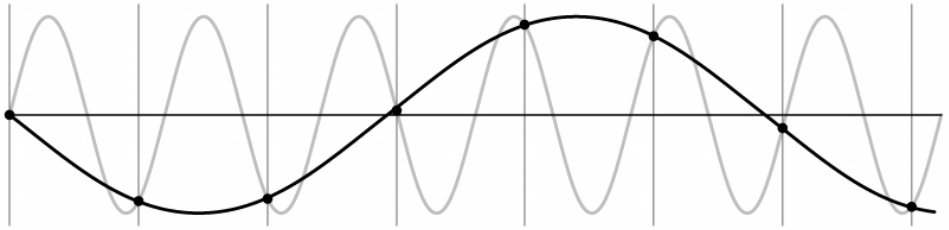
Fuente: https://commons.wikimedia.org/wiki/File:Shirt_with_moir%C3%A9_caused_by_aliasing.jpg

Teorema de muestreo de Nyquist

- Al muestrear una señal a intervalos discretos, la frecuencia de muestreo debe ser $\geq 2 \times f_{max}$.
- f_{max} = frecuencia máxima de la señal de entrada.
- Esto permite reconstruir el original a partir de la versión muestreada.
- bien



- mal



Anti-aliasing

- Soluciones:
- Muestrear más a menudo.
- Deshacerse de todas las frecuencias que sean mayores que la mitad de la nueva frecuencia de muestreo.
  - Se perderá información.
  - Pero es mejor que tener alias.
  - Aplicar un filtro suavizante.

Algoritmo para downsampling por 2

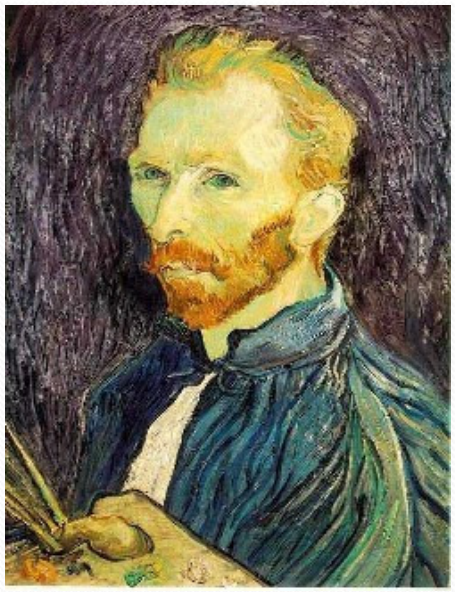
1. Comenzar con la imagen(h, w)
2. Aplicar un filtro pasa bajas

   ```
   image_blur = cv.GaussianBlur(image, ksize=(7, 7), sigmaX=0)
   ```

3. Muestrear cada segundo pixel

   ```
   image_small = image_blur[1:2:end, 1:2:end]
   ```

Subsampling sin filtrado previo



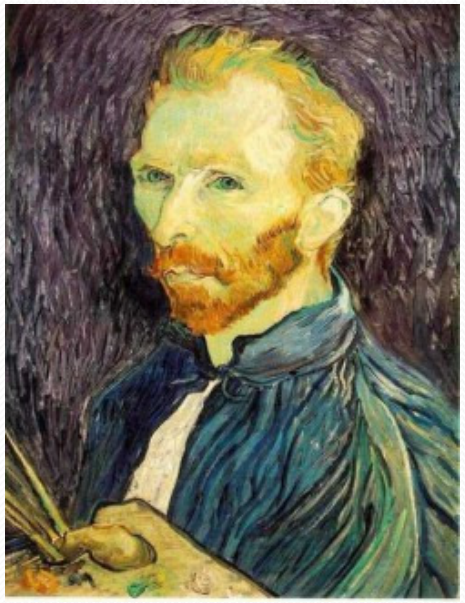
1/2

1/4 (2x zoom)

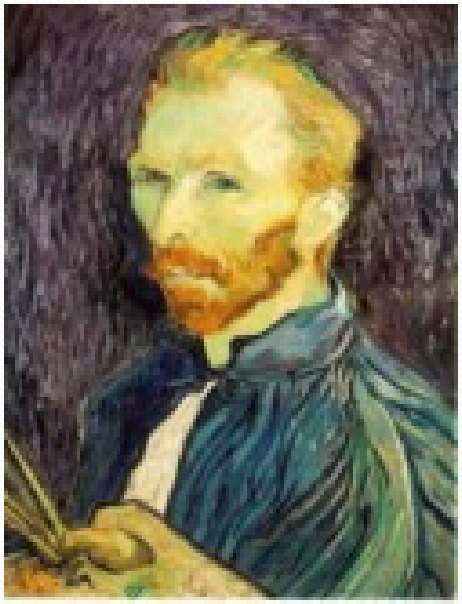
1/8 (4x zoom)

Fuente: Derek Hoiem (UIUC)

Subsampling con filtrado gaussiano previo



1/2



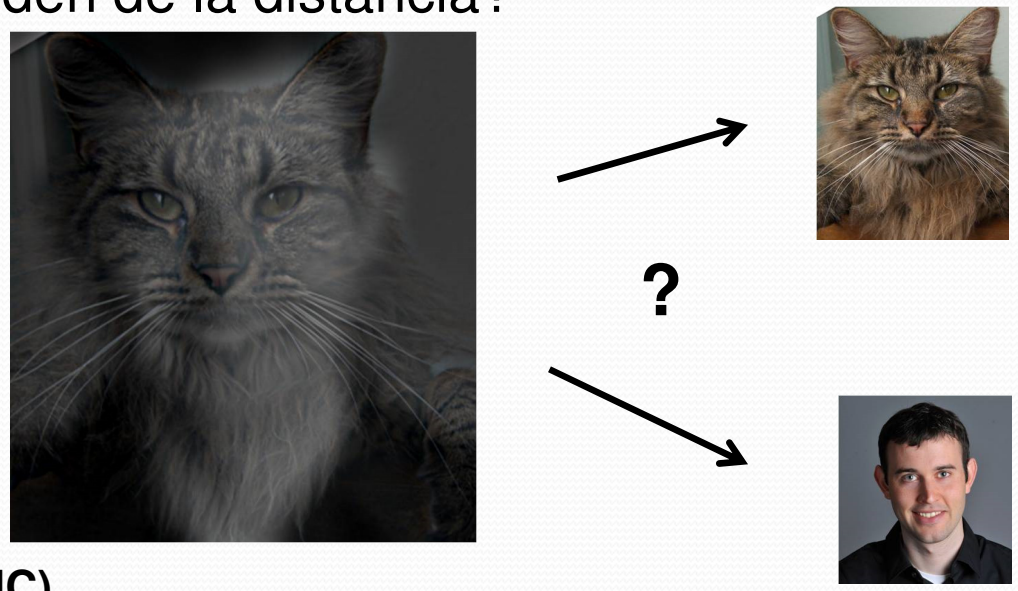
1/4 (2x zoom)

1/8 (4x zoom)

Fuente: Derek Hoiem (UIUC)

Imágenes híbridas

- ¿Por qué se obtienen interpretaciones diferentes de las imágenes híbridas que dependen de la distancia?



Fuente: Derek Hoiem (UIUC)

Lincoln in Dalivision

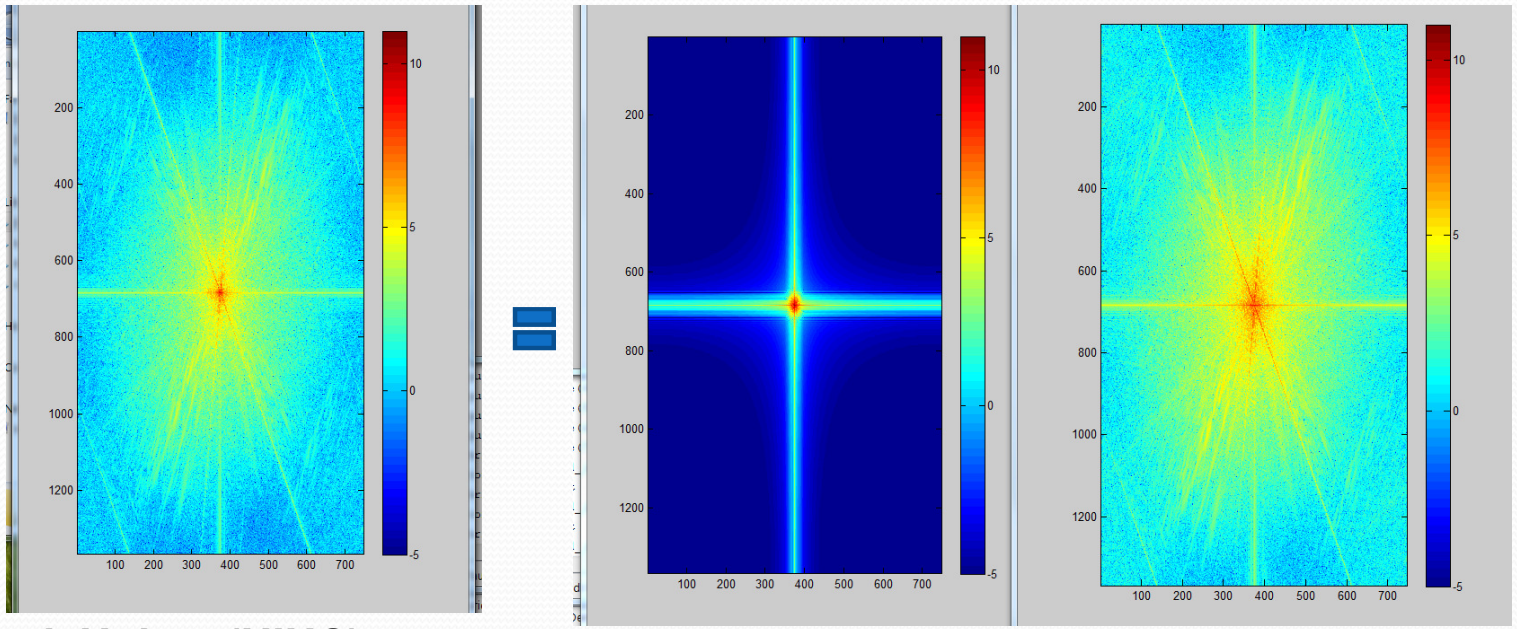
By Levine & Levine - http://www.doubletakeart.com/cgi-bin/dtg/dtg.sla?name=Salvador+Dali&lp=16323457802168094191&cc=dtg&ca=x&alc=dtg*64589&ai=00133*3676, Fair use, https://en.wikipedia.org/w/index.php?curid=36956616

Pistas de la percepción humana

- El procesamiento en humanos filtra varias orientaciones y escalas de frecuencia.
- Las señales perceptuales en las frecuencias medias dominan la percepción.
- Cuando se ve una imagen desde lejos, se le está submuestreando.

Imagen híbrida en FFT

Hybrid Image = Low-passed Image + High-passed Image



Fuente: Derek Hoiem (UIUC)

Conclusión

- A veces tiene sentido pensar en imágenes y filtrado en el dominio de la frecuencia.
  - Análisis de Fourier.
- Puede ser más rápido filtrar usando FFT para imágenes grandes ($N \log N$ vs N^2 para autocorrelación).
- Las imágenes son en su mayoría suaves.
  - Base para la compresión.
- Recordar realizar un suavizado antes de realizar el sampling.