

Filtrado de Imágenes

Diana Patricia Tobón Vallejo, PhD

Tratamiento de Señales III
Facultad de Ingeniería
Universidad de Antioquia



2024-2

Material elaborado por: Hernán Felipe García Arias

Contenido

- 1 [Definiendo el filtrado](#)
- 2 [Filtros pasa-bajas y *denoising*](#)
- 3 [Filtros no-lineales para *denoising*](#)
- 4 [Filtros pasa-altos y detección de bordes](#)
- 5 [Filtros en el dominio de la frecuencia](#)

Contenido

Definiendo el filtrado

2

Filtros

Definición

Un filtro modifica la información de la imagen dada una vecindad.

Propósito: mejorar o suprimir información.

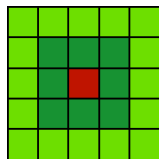
Se cubrirán principalmente dos clases de filtros:

- Filtros lineales espacialmente invariantes. Calculado con convolución
- Filtros no lineales

Vecindades

Definición

Una vecindad de un píxel/vóxel son los píxeles adyacentes. Los vecindarios se definen de manera diferente en diferentes cuadrículas.



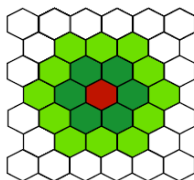
Center pixel



3x3-neighborhood



5x5-neighborhood



Center pixel



6-neighborhood



18-neighborhood

Filtros lineales espacialmente invariantes

Definiciones

- Continuos:

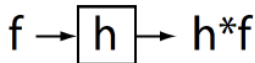
$$g(x) = h * f(x) = \int_{\Omega_h} f(x - \tau) h(\tau) d\tau \quad (1)$$

- Discretos:

$$g[x] = h * f[x] = \sum_{p \in \Omega_n} f[x - p] h[p] \quad (2)$$

donde $f(x)$ es la imagen a ser filtrada y h es el kernel del filtro para la convolución.

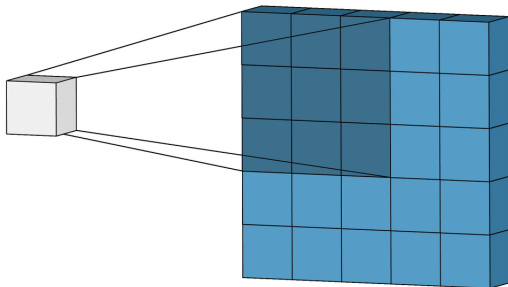
Diagrama de bloques



Convolución: Principio

La convolución discreta se calcula mediante:

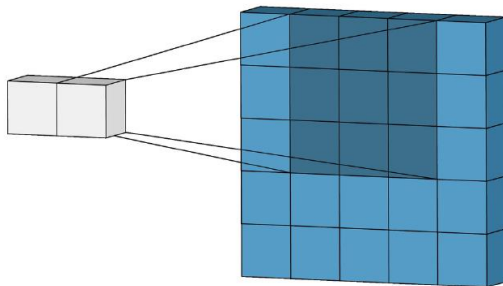
$$g[x] = f[x] * h = \sum_{\tau \in \Omega} f[x - \tau]h[\tau] \quad (3)$$



Convolución: Principio

La convolución discreta se calcula mediante:

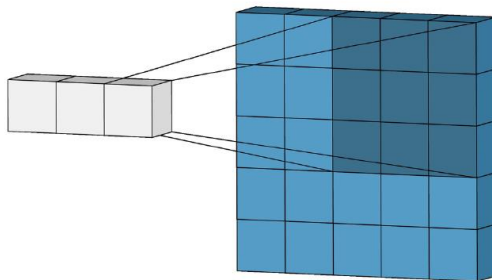
$$g[x] = f[x] * h = \sum_{\tau \in \Omega} f[x - \tau]h[\tau] \quad (3)$$



Convolución: Principio

La convolución discreta se calcula mediante:

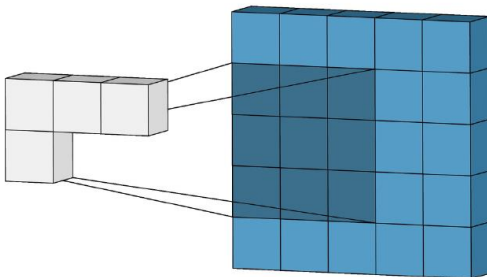
$$g[x] = f[x] * h = \sum_{\tau \in \Omega} f[x - \tau]h[\tau] \quad (3)$$



Convolución: Principio

La convolución discreta se calcula mediante:

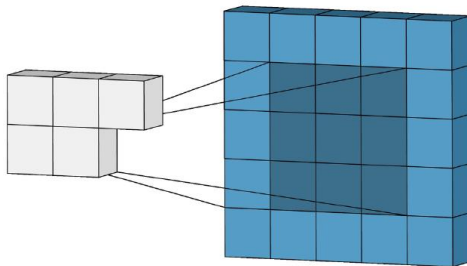
$$g[x] = f[x] * h = \sum_{\tau \in \Omega} f[x - \tau]h[\tau] \quad (3)$$



Convolución: Principio

La convolución discreta se calcula mediante:

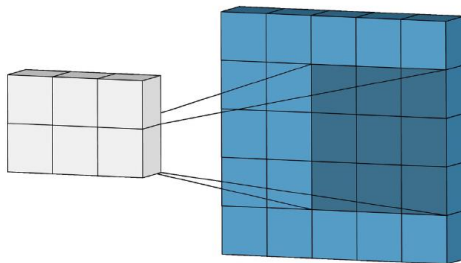
$$g[x] = f[x] * h = \sum_{\tau \in \Omega} f[x - \tau]h[\tau] \quad (3)$$



Convolución: Principio

La convolución discreta se calcula mediante:

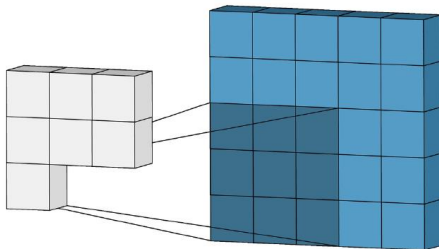
$$g[x] = f[x] * h = \sum_{\tau \in \Omega} f[x - \tau]h[\tau] \quad (3)$$



Convolución: Principio

La convolución discreta se calcula mediante:

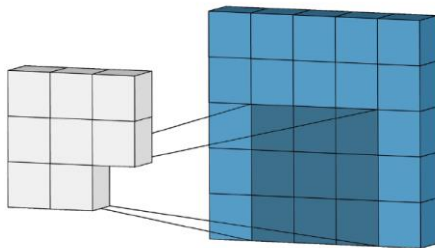
$$g[x] = f[x] * h = \sum_{\tau \in \Omega} f[x - \tau]h[\tau] \quad (3)$$



Convolución: Principio

La convolución discreta se calcula mediante:

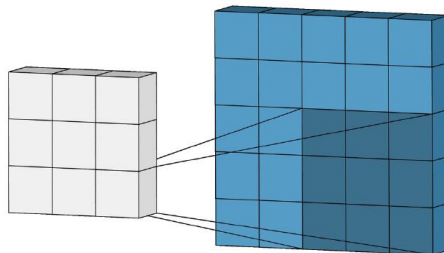
$$g[x] = f[x] * h = \sum_{\tau \in \Omega} f[x - \tau]h[\tau] \quad (3)$$



Convolución: Principio

La convolución discreta se calcula mediante:

$$g[x] = f[x] * h = \sum_{\tau \in \Omega} f[x - \tau]h[\tau] \quad (3)$$

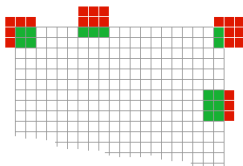


Procesamiento en los bordes

La convolución se define para imágenes infinitamente grandes.

Problema

Las imágenes discretas son finitas. El kernel de convolución no tiene soporte de imagen en los bordes



Estrategias para el procesamiento de bordes

- ☐ Ignorar los bordes
- ☐ Utilizar sólo la parte soportada del kernel
- ☐ Envolver alrededor
- ☐ Espejo

Propiedades de la convolución

- ❑ Conmutativa: $(a * b) * c = a * (b * c)$

Esta propiedad es útil si el kernel es separable.

- ❑ Asociativa: $a * (b + c) = a * b + a * c$

La parte izquierda de la ecuaciones es menos costosa computacionalmente que la derecha.

- ❑ Invariante espacialmente:

- Las operaciones de filtros sólo dependen de las intensidades en la imagen.
- Independiente de la posición.

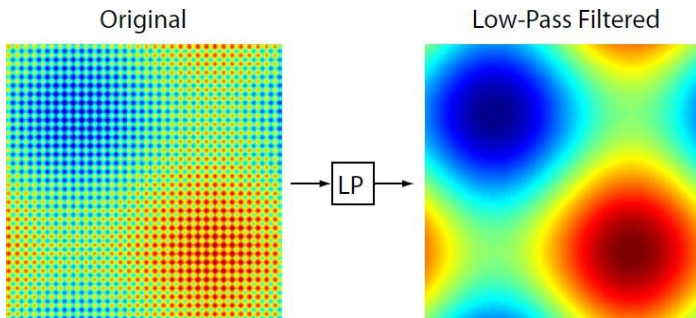
Contenido

- 1 [Definiendo el filtrado](#)
- 2 [Filtros pasa-bajas y *denoising*](#)
- 3 [Filtros no-lineales para *denoising*](#)
- 4 [Filtros pasa-altos y detección de bordes](#)
- 5 [Filtros en el dominio de la frecuencia](#)

Filtros pasa-bajo

Los efectos de un filtro pasa-bajos son en general:

- ❑ Cambios lentos son resaltados
- ❑ Cambios rápidos son suprimidos



Filtros pasa-bajo: Filtro de caja

Calcula el valor promedio local del vecindario de píxeles. Un filtro de caja 3×3 se define como:

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = h_x * h_y = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} * \frac{1}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (4)$$

Este filtro también es llamado filtro promedio (*mean filter*)

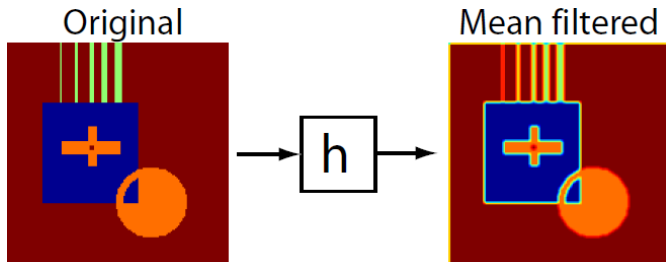
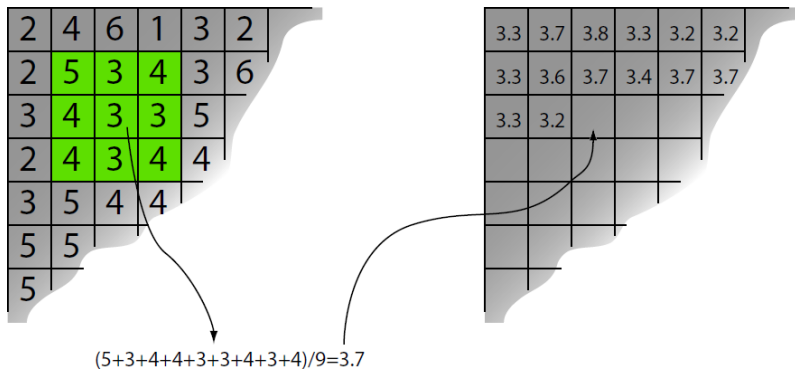


Figura: Observe: Los bordes son menos nítidos después del filtro de paso bajo.

Ejemplo detallado del filtro promedio



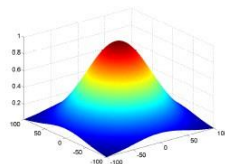
Filtro Gaussiano

El kernel Gaussiano

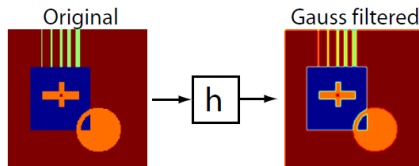
$$G_{\sigma} = \frac{1}{(2\pi\sigma)^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5)$$

kernel gaussiano de 3×3 , con $\sigma = 1$

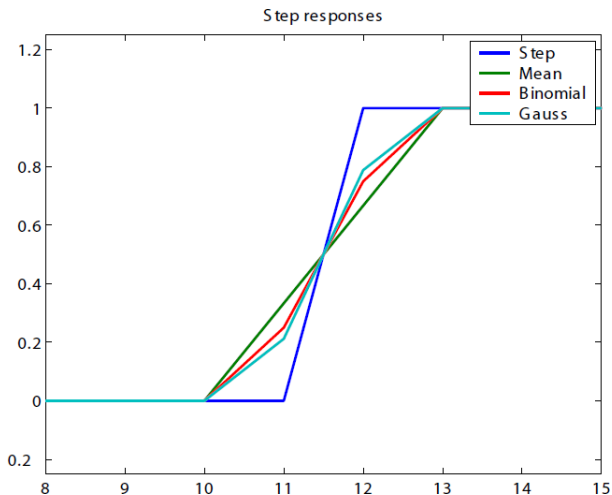
$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



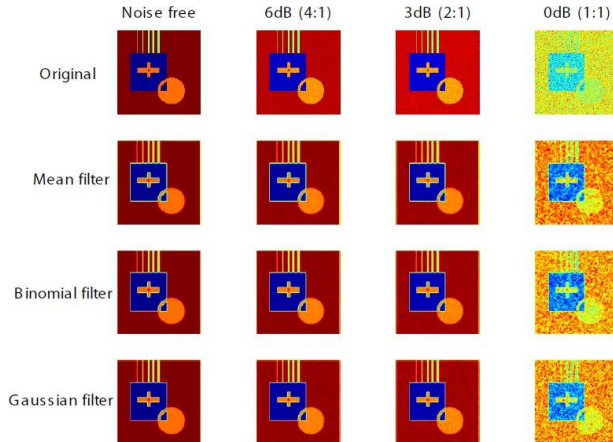
Ejemplo



Respuesta de diferentes filtros pasa-bajas



Filtros pasa-bajas con ruido



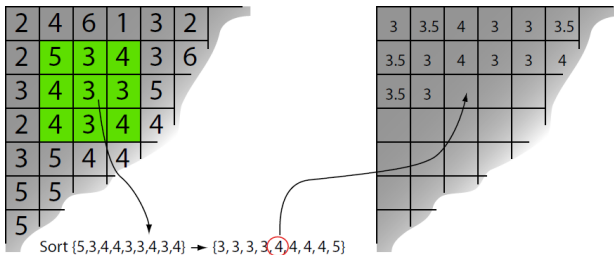
Contenido

- 1 [Definiendo el filtrado](#)
- 2 [Filtros pasa-bajas y *denoising*](#)
- 3 [Filtros no-lineales para *denoising*](#)
- 4 [Filtros pasa-altos y detección de bordes](#)
- 5 [Filtros en el dominio de la frecuencia](#)

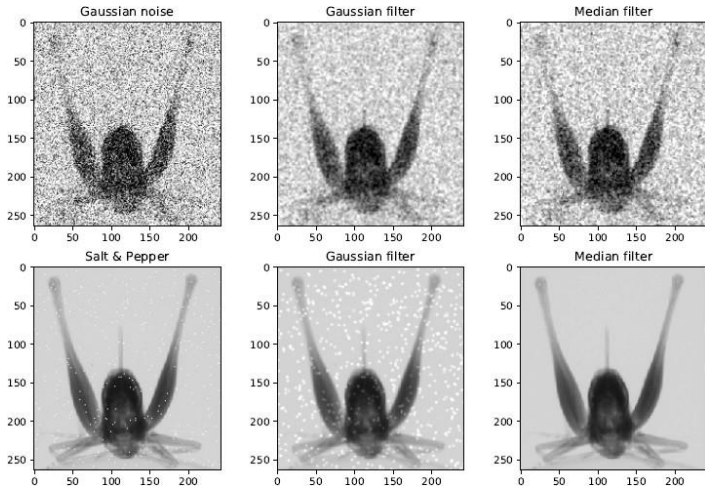
Filtro de mediana: Principio

El filtro de mediana es un tipo de paso bajo no lineal.

- ❑ Eliminación de ruidos atípicos como el ruido de sal y pimienta
- ❑ Es más suave para los bordes que la convolución con el mismo suavizado



Comparación de filtros de mediana y filtrado Gaussiano



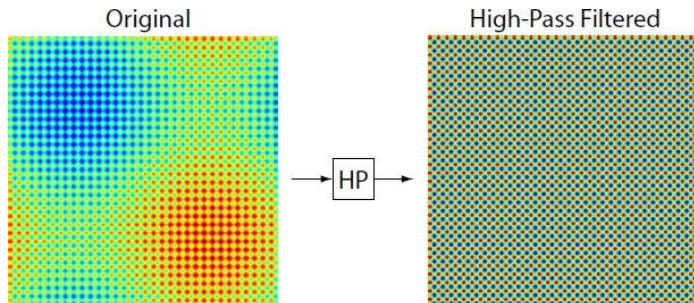
Contenido

- 1 [Definiendo el filtrado](#)
- 2 [Filtros pasa-bajas y *denoising*](#)
- 3 [Filtros no-lineales para *denoising*](#)
- 4 [Filtros pasa-altos y detección de bordes](#)
- 5 [Filtros en el dominio de la frecuencia](#)

Filtros pasa-alto

Los efectos de un filtro pasa-alto son en general:

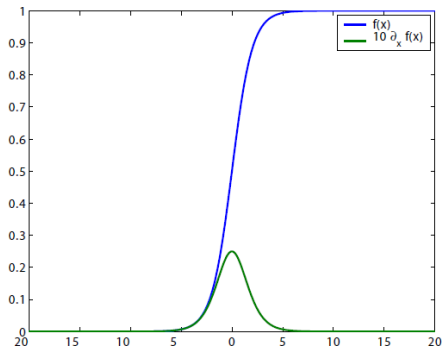
- ❑ Resaltar los cambios rápidos de intensidad.
- ❑ Suprimir los cambios lentos de intensidad.



Filtros paso-alto: Gradientes

Las propiedades del gradiente son:

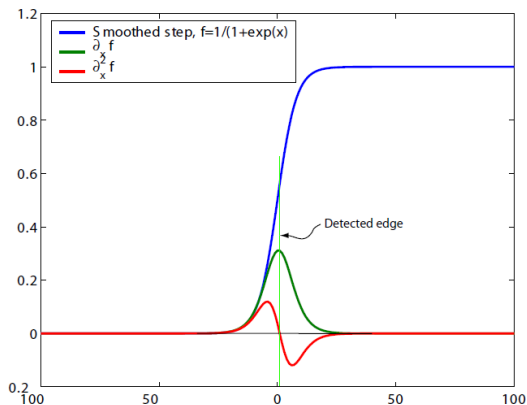
- ❑ Respuestas largas a cambios tipo paso (step).
- ❑ No presenta respuesta para niveles de intensidad constantes.



Detección de bordes - Introducción

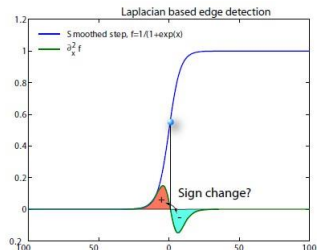
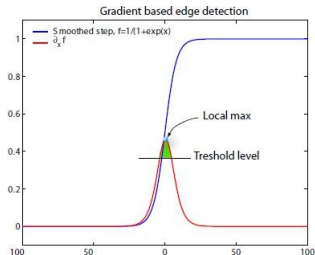
Las características relevantes de un borde son:

- Los bordes presentan una alta componente frecuencial.
- El gradiente tiene una amplitud grande en el borde.



Detección de bordes - Principios

Los bordes pueden ser detectados mediante las primeras y segundas derivadas de la imagen.



Detección de bordes y ruido

Problema

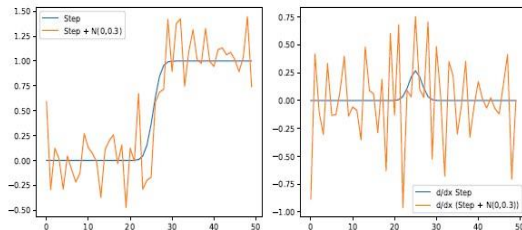
El mayor problema para un detector de bordes es el ruido, ya que puede:

- ❑ Ocultar bordes relevantes
- ❑ Resultar en bordes no deseados

Soluciones

- ❑ Realizar un suavizado antes de la detección de bordes
- ❑ *Track edges*

Ejemplo



Detección de bordes: Implementaciones

La robustez en la detección de bordes es mejorada por un suavizado perpendicular.

Algunos detectores de bordes famosos son:

Sobel edge detector masks

-1	-2	-1
1	2	1

-1		1
-2		2
-1		1

Prewitt edge detector masks

-1	-1	-1
1	1	1

-1		1
-1		1
-1		1

See e.g. Haralik and Shapiro (1992)

Contenido

- 1 [Definiendo el filtrado](#)
- 2 [Filtros pasa-bajas y *denoising*](#)
- 3 [Filtros no-lineales para *denoising*](#)
- 4 [Filtros pasa-altos y detección de bordes](#)
- 5 [Filtros en el dominio de la frecuencia](#)

La transformada de Fourier

Se procede a cambiar la representación de la imagen por:

- ❑ Se mapea la información de la imagen en un nuevo dominio
- ❑ Se exploran las componentes de frecuencia espacial

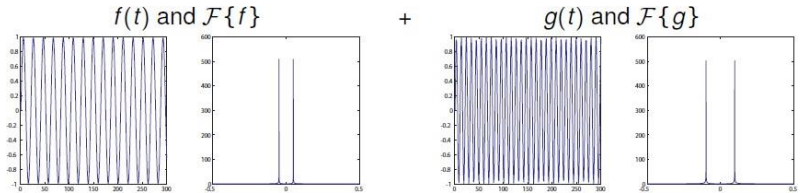
$$G(\zeta_1, \zeta_2) = \mathcal{F}\{g\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \exp\{-i(\zeta_1 x + \zeta_2 y)\} dx dy$$

Algunas propiedades útiles de la FT^a

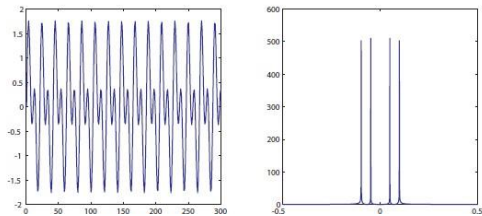
^aGeneral Jähne Jähne (2002) or Detailed Granlund Granlund and Knutsson (1995),[1]

- ❑ Adición: $af(x) + bg(x) \leftrightarrow aF(w) + bG(w)$
- ❑ Convolución: $f * g(t) \leftrightarrow F(w) \cdot G(w)$
- ❑ Simetría: $x \in \mathbb{R} \rightarrow F(w) = F(-w)$

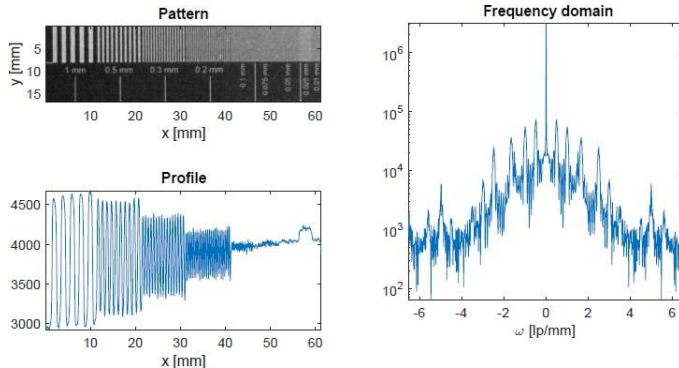
Adición de señales 1D



$f(t) + g(t)$ and $\mathcal{F}\{f\} + \mathcal{F}\{g\}$



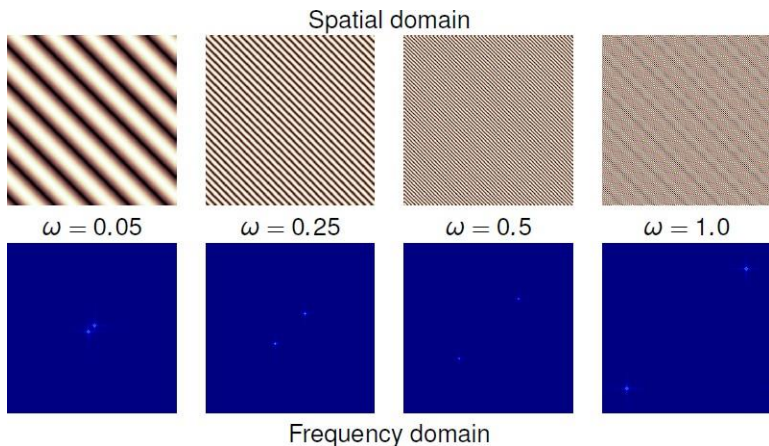
Transformada de Fourier de una línea de patrón real



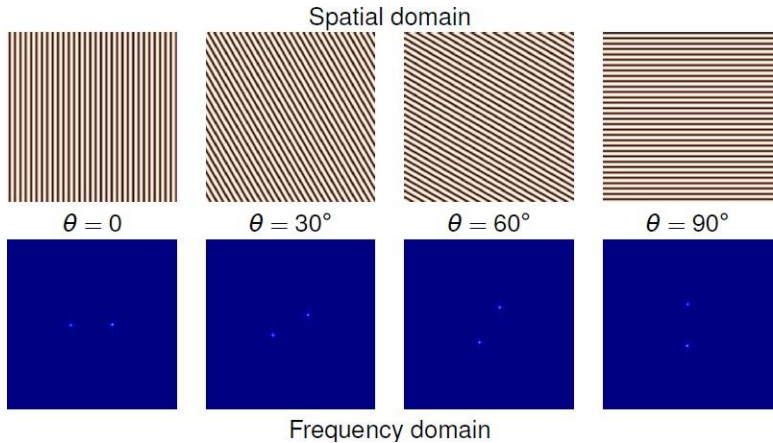
¿Qué podemos ver?

- ☐ Picos para cada línea de frecuencia
- ☐ Nivel de ruido

Transformada de Fourier en 2D - frecuencias espaciales



2D FFT - Orientaciones



Bibliography



Rafael C. Gonzalez and Richard E. Woods.
Digital image processing.
Prentice Hall, Upper Saddle River, N.J., 2008.