

Filtrado paso alto y paso bajo

Visión por Computador, curso 2023-2024

Silvia Martin Suazo, silvia.martin@u-tad.com

3 de octubre de 2023

U-tad | Centro Universitario de Tecnología y Arte Digital



Motivación

¿Para qué procesar imágenes?

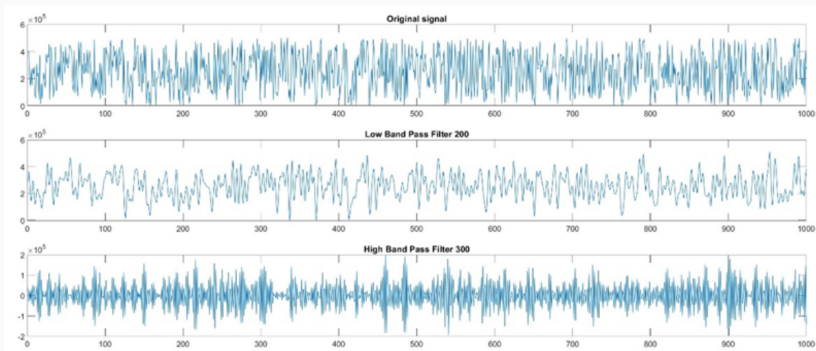
Como se ha visto anteriormente la **convolución** permite realizar cambios como detección de características o suavizado a una imagen.

Las operaciones de filtrado se pueden dividir en **dos grandes grupos**:

- Filtrado de paso alto
- Filtrado de paso bajo

¿Para qué procesar imágenes?

Ambos tipos de filtros son una extensión de los filtros utilizados para procesamiento de señales.

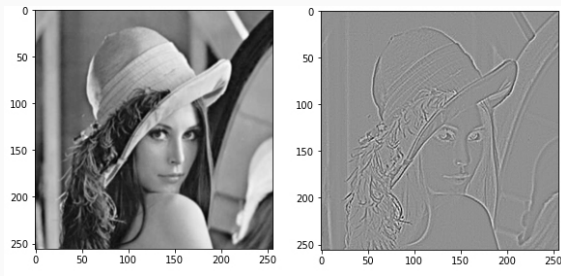


¿Para qué procesar imágenes?

Como se ha visto anteriormente la **convolución** permite realizar cambios como detección de características o suavizado a una imagen.

Las operaciones de filtrado se pueden dividir en **dos grandes grupos**:

- **Filtrado de paso alto**
- Filtrado de paso bajo



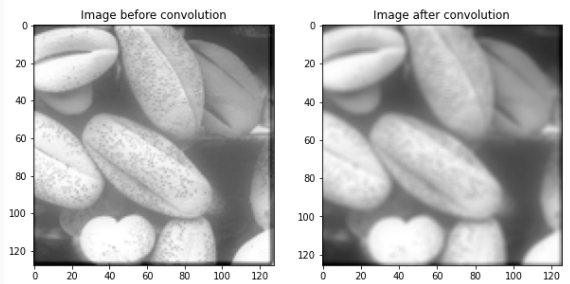
Se centra en **resaltar estructuras** de la imagen.

¿Para qué procesar imágenes?

Como se ha visto anteriormente la **convolución** permite realizar cambios como detección de características o suavizado a una imagen.

Las operaciones de filtrado se pueden dividir en **dos grandes grupos**:

- Filtrado de paso alto
- **Filtrado de paso bajo**



Tiene como objetivo **eliminar ruido** de la imagen manteniendo sus

Notebook de ejemplos de filtros

Los filtros que se explicarán a continuación pueden ser observados de manera práctica en el siguiente notebook.



· 04-Filtrado.ipynb

Transformada de Fourier

Transformada de Fourier

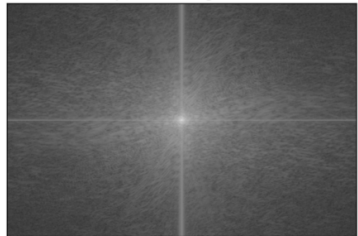
La transformada de fourier permite el análisis de la **distribución de las frecuencias**, para decidir que frecuencias mantener o eliminar.

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \quad (1)$$

Imagen Original



Espectro de Magnitudes



Filtrado de paso bajo

Filtrado de caja o media

El filtrado de **caja** es una técnica de suavizado que realiza una media del vecindario o caja de cada píxel, para así atenuar el ruido de la imagen.

Es una operación de **convolución** con el siguiente kernel:

$$box_{m,n} = \frac{1}{mn} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}_{m \times n}$$

[1]

Filtrado de caja o media

$$\text{ker}^{5 \times 5} = \frac{1}{25}$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

GIF example

Filtrado de caja o media

Cuanto **mayor** sea el tamaño del filtro, **mayor** será el **promediado**.

El principal **inconveniente** de este filtro es que es demasiado **duro** en la eliminación de ruido.

Existen filtros más **sofisticados** que son capaces de eliminar ruido de manera **selectiva**.

Imagen Original



Filtro de caja



Filtro de 20x20

Filtrado gaussiano

Realiza un promediado de los píxeles del vecindario siguiendo una **distribución gaussiana** bidimensional.

El valor del filtro de la convolución viene marcado por una distribución gaussiana centrada en el **píxel central** con desviación típica σ

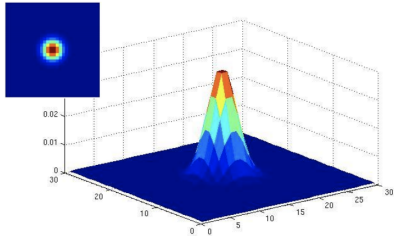
$$K(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

$$\text{ker}^{3 \times 3} = \frac{1}{16} \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

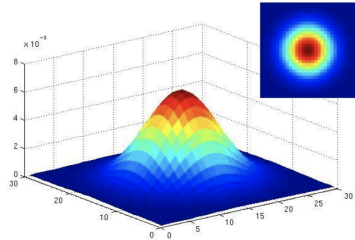
Matriz del kernel 3x3 para un σ de 1

Filtrado gaussiano

A mayor **desviación**, mayor será la **atenuación** de la imagen.



$\sigma = 2$ with 30×30
kernel

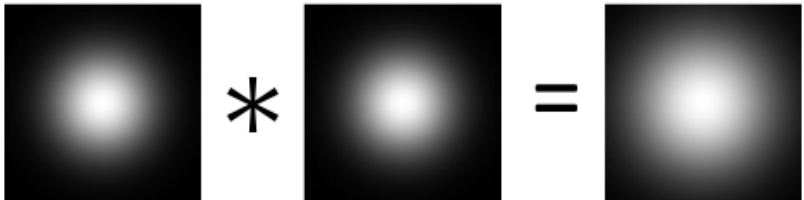


$\sigma = 5$ with 30×30
kernel

[2]

Filtrado gaussiano

Al aplicar dos veces un filtrado gaussiano de desviación σ el resultado es el mismo que aplicar un **único filtrado** de desviación $\sigma\sqrt{2}$.



[3]

$$\text{Tamaño del kernel} = 2\lceil 3\sigma \rceil + 1 \quad (3)$$

Imagen Original



Filtro gaussiano



Filtro de 31×31 y $\sigma = 5$

Otros filtros de paso bajo

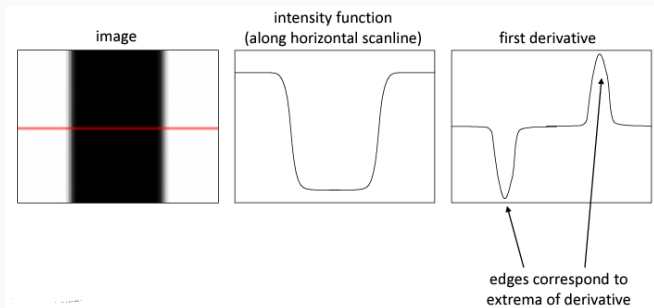
- Filtro de suavizado exponencial: asigna pesos exponenciales a los píxeles en lugar de pesos uniformes, y dando más importancia a píxeles cercanos.
- Filtro de promedio ponderado: asigna pesos predefinidos a los píxeles según su posición en la ventana.
- Filtro de Butterworth

Filtrado de paso alto

Detección de bordes

A la hora de hallar los bordes de una imagen, estos pueden ser observados con los **cambios de intensidad** de los píxeles que la forman.

En este sentido, a través de los máximos y mínimos de la **derivada** de la función de intensidad es posible identificar los bordes de una imagen.

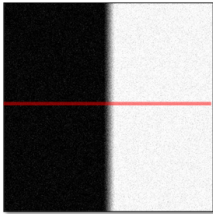


El **gradiente** de una imagen calcula el **cambio en la intensidad** de los píxeles en un área. A través del gradiente se puede ver los límites de ciertas **estructuras** en una imagen.

El gradiente en una imagen **bidimensional** es calculado a través de **derivadas parciales** en el eje X e Y.

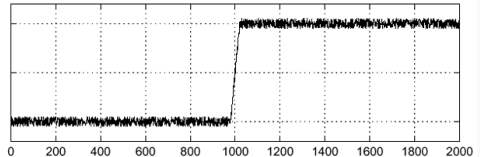
Derivadas y ruido

Si se aplica la derivada a una imagen que contiene **ruido**, el resultado **estará afectado** por la presencia de dicho ruido.

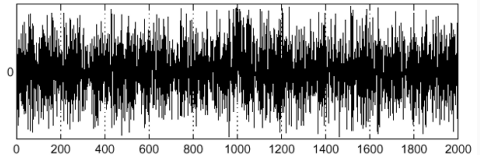


Noisy input image

$$f(x)$$



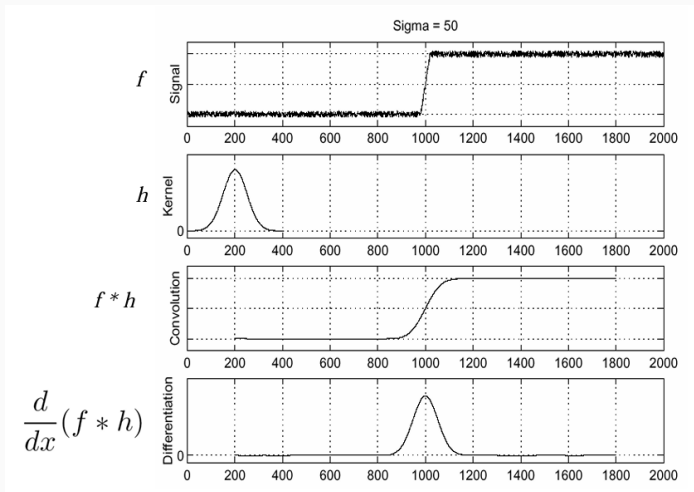
$$\frac{d}{dx}f(x)$$



[5]

Derivadas y ruido

La solución para evitar este problema es aplicar primero un **filtrado de paso bajo**, que **elimine** la presencia de ruido en la imagen.



Debido a la **propiedad asociativa** de la convolución se tiene la siguiente igualdad:

$$\frac{d}{dx}(f * h) = f * \frac{d}{dx}h \quad (4)$$

Lo cual permite calcular la derivada de una convolución simplemente **derivando el filtro** de esta.

El **filtro de Prewitt** [7] sería el equivalente al filtro de tipo caja para un filtrado de paso alto.

$$\text{ker}_y^{3 \times 3} = \frac{1}{6} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

$$\frac{1}{6} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

(a) Kernel para bordes horizontales

$$\frac{1}{6} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

(b) Kernel para bordes verticales

Filtro de Prewitt

Imagen Original



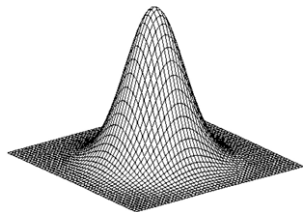
Filtro Prewitt



Con filtro de 3x3

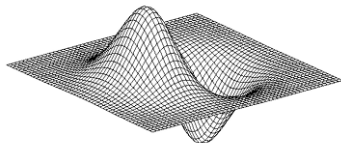
Filtro derivada del Gaussiano

El **filtro derivada del Gaussiano** consiste en derivar parcialmente la función **gaussiana** en alguna de sus dos dimensiones.



Gaussian

$$h_{\sigma}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian (x)

$$\frac{\partial}{\partial x} h_{\sigma}(u, v)$$

[8]

El **filtro de Sobel** es una aproximación a la derivada del Gaussiano.

$$\text{ker}_y^{3 \times 3} = \frac{1}{8} \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array}$$

$$\frac{1}{8} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

(a) Kernel para bordes horizontales

$$\frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

(b) Kernel para bordes verticales

Filtro de Sobel

Imagen Original



Filtro Sobel



Con filtro de 31x31

Filtro del Laplaciano

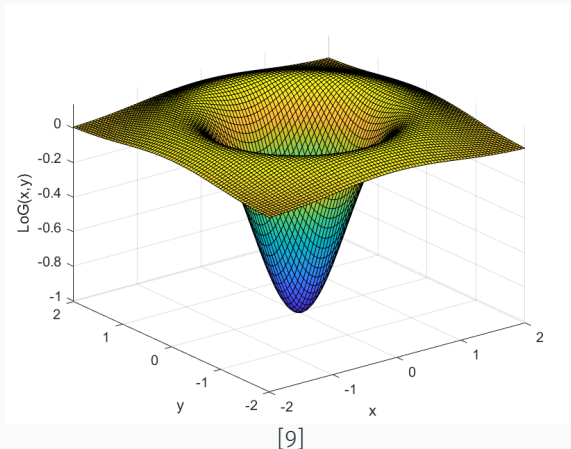
El **filtro Laplaciano** viene dado por una convolución que tiene valores negativos para sus píxeles centrales y positivos para una región más amplia.

$\text{ker}^{3 \times 3} =$

0	1	0
1	-4	1
0	1	0

Filtro del Laplaciano

La función que define la convolución se llama **operador laplaciano**, que aplicado a una **Gaussiana** obtiene como resultado el **filtro del Laplaciano**. Esta operación se conoce como la **Laplaciana de la Gaussiana**.

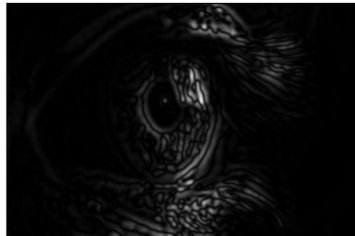


Filtro de Laplaciano

Imagen Original



Filtro Laplaciano



Con filtro de 31x31

- [1] NVidia.
Box filter.
[Online; accessed September, 2023].
- [2] jun94 (Medium).
Gaussian kernel comparison image.
[Online; accessed August, 2022].
- [3] Noah Snavely.
Two gaussian convolutions image.
[Online; accessed August, 2022].
- [4] Noah Snavely.
Characterizing edges image.
[Online; accessed August, 2022].

- [5] Noah Snavely.
Derivative noise image.
[Online; accessed August, 2022].
- [6] Noah Snavely.
Noise attenuation image.
[Online; accessed August, 2022].
- [7] Judith MS Prewitt et al.
Object enhancement and extraction.
Picture processing and Psychopictorics, 10(1):15–19, 1970.
- [8] Noah Snavely.
Gaussian derivative image.
[Online; accessed August, 2022].

- [9] Manuel Henriques, Duarte Valério, Paulo Gordo, and Rui Melicio.
Fractional-order colour image processing.
Mathematics, 9(5):457, 2021.