

Implementación de algoritmos

Guillermo Blanco Núñez - 16/11/2025

Práctica 2 - Grupo 1.1 Principios de Visión por Computador

3º Grado en Inteligencia Artificial

Algoritmo 1: Implementar una función que calcule un kernel Gaussiano unidimensional con σ dado.

La función `gaussianFilterSpatial1D` genera un kernel gaussiano 1D para un valor dado de σ . Calcula el tamaño $N=2\lceil 3\sigma \rceil + 1$, centra la ventana, evalúa la gaussiana discretizada en cada posición y normaliza el vector para que actúe como un filtro de suavizado en convolución.

Algoritmo 2: Implementar una función que permita realizar un suavizado Gaussiano bidimensional usando un filtro $N \times N$ de parámetro σ , donde N se calcula igual que en la función anterior.

La función aplica un suavizado Gaussiano 2D usando la separabilidad del filtro. Primero obtiene el kernel 1D utilizando la función implementada anteriormente, luego realiza dos convoluciones: una horizontal y otra vertical. Así construye el filtro $N \times N$ sin calcularlo explícitamente y genera la imagen suavizada final.

Comparación de suavizado Gaussiano ($\sigma=2$)

Mi función `ApplyGaussianFilterSpatial`

Función de referencia SciPy `gaussian_filter`



Como se puede ver los resultados son casi idénticos que con la función nativa de python, y cuanto mayor sea el sigma más se suaviza la imagen entonces más borrosa es.

Algoritmo 3: Implementar una función que calcule un filtro gaussiano en frecuencia, de tamaño $N \times M$ y un σ dado.

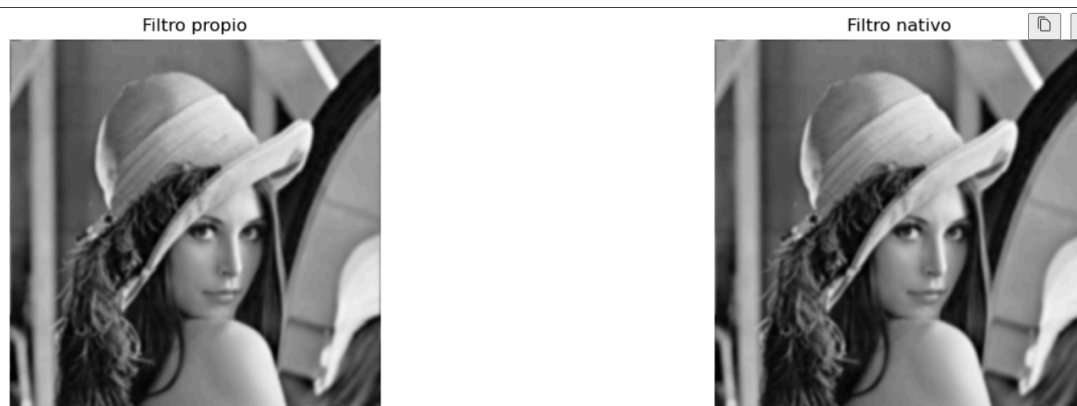
La función genera un filtro gaussiano en el dominio frecuencial del mismo tamaño que la imagen. Crea una malla de frecuencias centradas, normaliza las coordenadas y evalúa la Gaussiana en frecuencia:

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

El resultado es un filtro suavizante listo para multiplicarse con el espectro. Aunque la función esté utilizando un sigma con valor espacial el filtro sigue siendo un filtro en frecuencia porque la operación final se realiza multiplicando la transformada de Fourier de la imagen por $H(u,v)$, es decir, el filtrado ocurre completamente en el dominio de la frecuencia.

Algoritmo 4: Implementar una función que aplique el filtro gaussiano en frecuencia anterior sobre una imagen, con el filtro del tamaño de la imagen y un σ dado.

La función aplica el suavizado Gaussiano en el dominio frecuencial: calcula la FFT de la imagen, la centra, multiplica su espectro por el filtro Gaussiano y realiza la transformada inversa. El resultado es la imagen suavizada obtenida mediante filtrado en frecuencia.



Al utilizar un valor de sigma espacial en la función `GaussianFilterFrec()`, la imagen se hace más borrosa según aumenta el valor de sigma, al igual que en la función nativa. Esto sucede ya que un sigma mayor ensancha la gaussiana, suaviza más la imagen y elimina un rango mayor de altas frecuencias.