

Procesamiento de Imágenes

Ingeniería Biomédica

Unidad 3: Filtrado en el dominio espacial

- ☐ Vecindad
- ☐ Convolución
- ☐ Ruido aditivo y multiplicativo
 - ❖ Ruido Gaussiano
 - ❖ Ruido impulsivo (sal y pimienta)
 - ❖ Ruido speckle
- ☐ Filtros lineales: pasa bajos, pasa altos, detecta bordes (Roberts, Sobel, Prewitt, Laplaciano, LOG, ...)
- ☐ Filtro no lineal: mediana

Tipos de vecindad

(horizontal, vertical y diagonal)

	(x,y-1)	
(x-1,y)	(x,y)	(x+1,y)
	(x,y+1)	

N4: 4 píxeles (H+V)

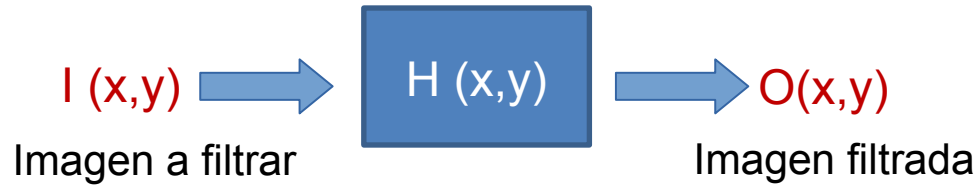
(x-1,y-1)		(x+1,y-1)
	(x,y)	
(x-1,y+1)		(x+1,y+1)

ND: 4 píxeles diagonales

(x-1,y-1)	(x,y-1)	(x+1,y-1)
(x-1,y)	(x,y)	(x+1,y)
(x-1,y+1)	(x,y+1)	(x+1,y+1)

N8: 8 píxeles vecinos

Convolución 2D



$$O(x,y) = I(x,y) \otimes H(x,y)$$

Es la base del filtrado espacial de imágenes. Difiere de la correlación en que primero se rota 180° la **máscara o kernel**.

Imagen

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

Máscara

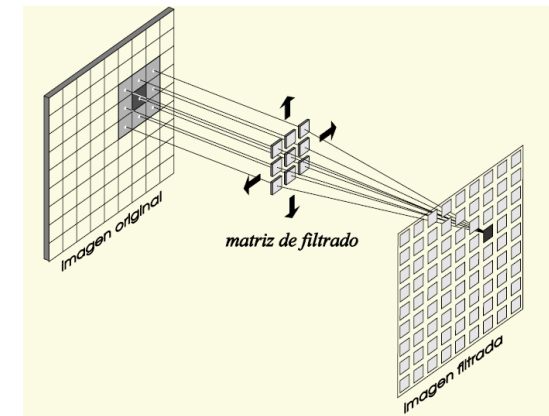
8	1	6
3	5	7
4	9	2

1) **Rotar la máscara 180°**

La máscara rotada será:

2	9	4
7	5	3
6	1	8

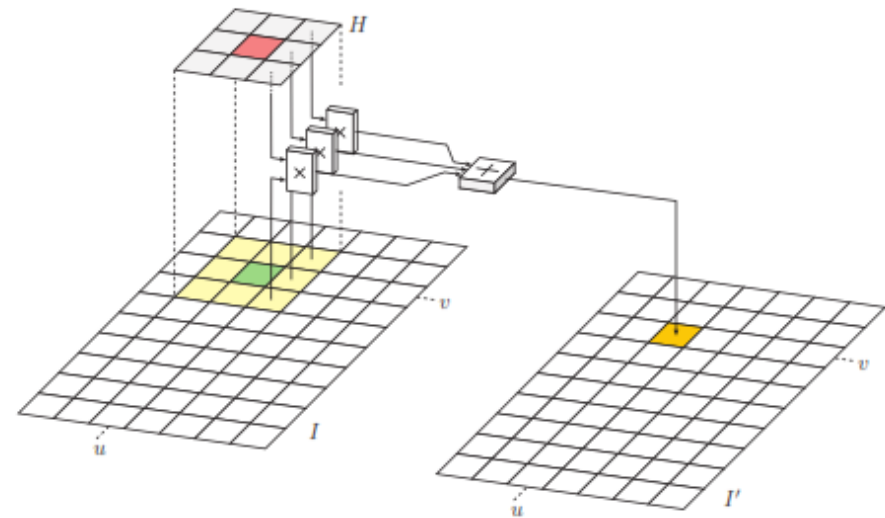
2) **Superponer** el elemento central de la máscara de tal forma que quede sobre el elemento de interés de la imagen.



3) **Multiplicar** cada peso (coeficiente de convolución) de la máscara rotada por el píxel de la imagen que se encuentra “debajo” de la máscara.

4) **Sumar los productos** individuales obtenidos en el paso anterior. El resultado reemplaza el valor del píxel debajo del centro de la máscara.

Obs: en los bordes de la imagen se acostumbra a insertar ceros (**zero padding**).



Comando imfilter

$O = \text{imfilter}(I, H, \text{modo}, \text{borde}, \text{tamaño})$

Modo: 'corr' para realizar la correlación (default)

'conv' para efectuar la convolución

Borde:

1) **P** los bordes de la imagen se extienden con píxeles de valor P (default P=0)

2) 'replicate' copia el valor de los píxeles de los bordes de la imagen.

3) 'symetric' extiende la imagen espejando los bordes

4) 'circular' interpreta la imagen como una función periódica 2D.

Tamaño: 'same' (default) o 'full' (no existe la opción 'valid')

- ❑ **Full**: la operación se realiza usando zero-padding o replicando los bordes. Inicio: el pixel de la esquina inferior derecha de H, se alinea con el primer pixel de la imagen
 - ❑ **Same**: el resultado de la operación es del mismo tamaño que la imagen. Hay zero-padding también. Inicio: el centro de la máscara se alinea con el primer pixel de la imagen.
 - ❑ La operación se realiza usando dobles precisión, aritmética de punto flotante.
 - ❑ La **imagen de salida** tendrá la misma clase de datos que la **imagen de entrada**.
- ❑ Si $I(x,y)$ es un array de números enteros, los elementos de la salida que excedan el rango se recortan, y las fracciones se redondean.
 - ❑ Se debería convertir $I(x,y)$ a clase doble (im2double) antes de filtrar.

Comando conv2

$$O = \text{conv2}(I, H, \text{modo})$$

- Modo: **'full'** (default), **'same'** devuelve la parte central de la convolución, **'valid'** retorna solo aquella parte de la convolución que se computa sin zero-padding.

Ruido

- Se considera ruido a cualquier perturbación que afecta a una señal en el proceso de **adquisición - transmisión - almacenamiento**. Se modela usualmente como un proceso *estocástico*.
- Los sensores tienden a degradar la calidad de las imágenes al introducir *ruido, deformaciones geométricas, movimiento, desenfoque*, etc.

Fuentes de ruido en imágenes

- ❑ Ruido generado durante la adquisición/digitalización: defectos en los sensores, bajos niveles de luz, CCD (Charge-Coupled Device) sensibles a la temperatura, etc.
- ❑ Ruido generado durante la transmisión: interferencias en el canal de transmisión

Algunos tipos de ruido:

Ruido multiplicativo: la imagen obtenida es el resultado de la multiplicación de dos señales.

Ruido aditivo: se suma a la imagen una cierta cantidad de ruido que se origina durante el registro o en la transmisión de la misma

Ruido Gaussiano: ruido generado por digitalizadores, perturbaciones en la transmisión, etc. El ruido presenta una distribución del tipo «campana de Gauss».

Ruido impulsional (sal y pimienta): algunos píxeles toman valores extremos (puntos blancos o negros). En cámaras IR, las imágenes de objetos muy calientes pueden saturar algunos píxeles y generar ruido.

Comando imnoise

Este comando permite corromper una imagen con ruido, su sintaxis es:

O = imnoise (I, type, parameters)

Obs: imnoise convierte la imagen I a clase double (rango [0, 1]) antes de agregarle ruido

❖ $O = \text{imnoise}(I, \text{'salt \& pepper'}, d)$

Corrompe la imagen con ruido sal y pimienta, donde d es la densidad del ruido (% de píxeles de la imagen conteniendo ruido). Son afectados aproximadamente $d \cdot \text{numel}(I)$ píxeles. Default $d=0.05$

❖ $O = \text{imnoise}(I, \text{'gaussian'}, m, \text{var})$

Adiciona ruido Gaussiano de media m y varianza var . Default $m=0$ y $\text{var}=0.01$

❖ $O = \text{imnoise}(I, \text{'speckle'}, \text{var})$

Agrega ruido multiplicativo a la imagen I , usando la ecuación $O = I + n \cdot I$, donde n es ruido aleatorio distribuido uniformemente con media $m=0$ y varianza var (default $\text{var}=0.04$).

Patrón granular presente en las imágenes de ultrasonido (ecografías) y de radar.

Ejercicio N°1: Abra una imagen y agréguele ruido con el comando `imnoise`.

Imagen original

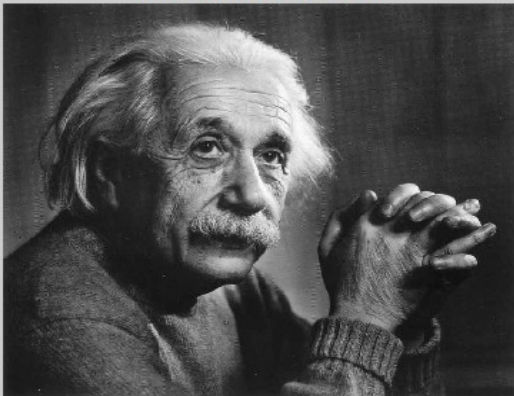
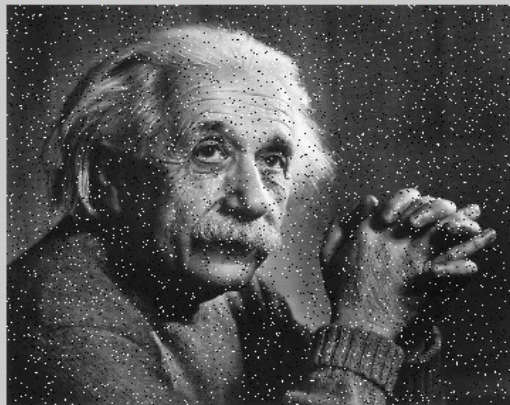
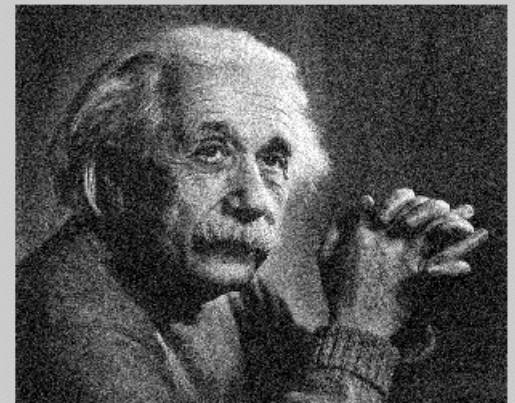


Imagen con ruido sal y pimienta



Ruido Gaussiano, $\mu=0$ $\sigma^2=0.01$



Filtros pasa-bajos

$$\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

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

- ☐ $\sum_{\text{coef}} = 1 \rightarrow$ no cambia el brillo total de la imagen (redistribuimos los valores de intensidad).
- ☐ Todos los coeficientes del kernel son positivos.
- ☐ Se realiza un promedio ponderado.
- ☐ Se reducen las altas frecuencias, *suavizan* los bordes (*difuminado*), las imágenes se perciben más *borrosas*.

Ejercicio N°2: Abra una imagen y aplíquela diferentes filtros pasa bajos. Visualice los resultados y observe qué ocurre cuando se modifican los coeficientes de los kernel 3x3.



Imagen original

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Imagen filtrada

Filtros pasa-altos

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

- ❑ $\sum_{\text{coef}} = 1 \rightarrow$ no cambia el brillo total de la imagen, el coeficiente central es positivo y grande, y los vecinos son positivos pequeños o negativos.
- ❑ Realzan las componentes espaciales de altas frecuencias (detalles finos)

Ejercicio N°3: Abra una imagen y aplíquela diferentes tipos de filtros pasa altos. Visualice el resultado y observe qué ocurre cuando se modifican los coeficientes (kernel 3x3).



Imagen original

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



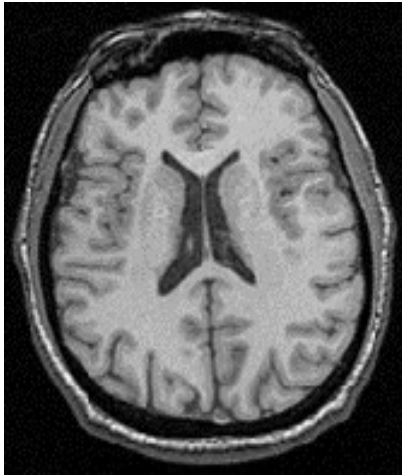
Imagen filtrada

Diferentes tamaños de kernel

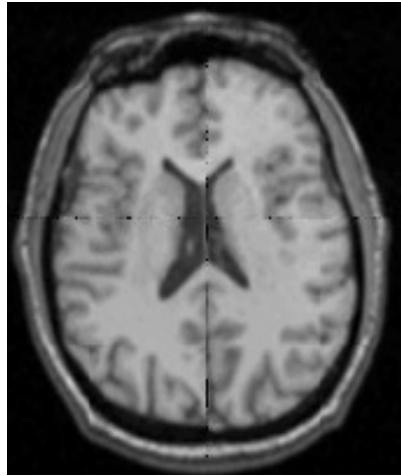
Ejemplos de filtros pasa-bajos y pasa-altos

Ejercicio N°4: Modifique el tamaño del kernel y vea que sucede. Pruebe con kernels impares.

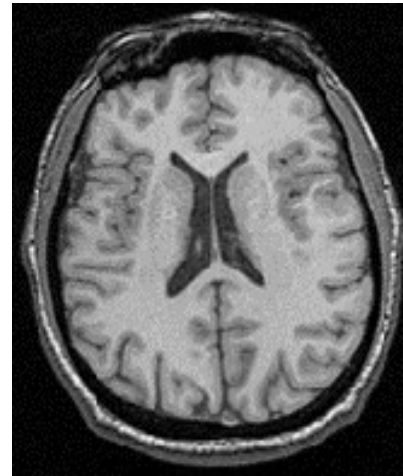
original



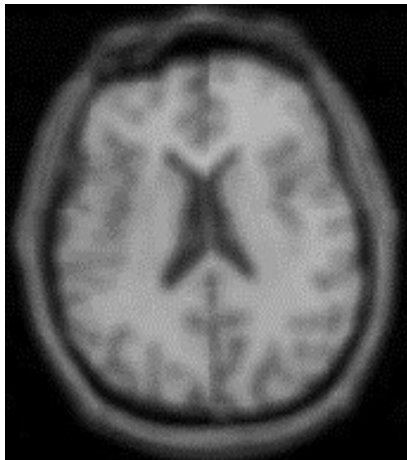
3 x 3



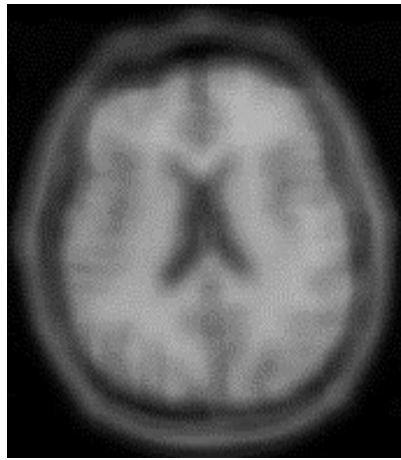
original



3 x 3



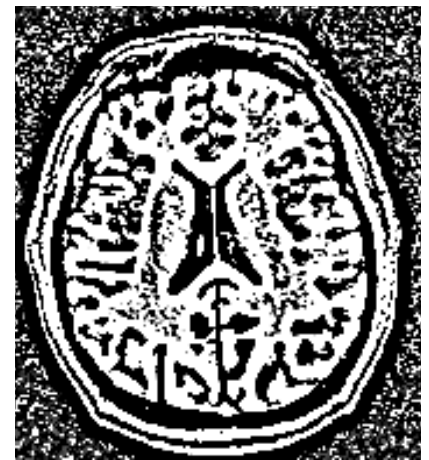
7 x 7



11 x 11



7 x 7



11 x 11

Filtros espaciales lineales

$H = \text{fspecial}(\text{'tipo'}, \text{parámetros})$

❖ **Filtro promediador:** $H = \text{fspecial}(\text{'average'}, [r \ c])$
[r c] es el tamaño del filtro (default 3x3).

$H1 = \text{fspecial}(\text{'average'})$

$H2 = \text{fspecial}(\text{'average'}, [5 \ 5])$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

Imagen original



Filtrado average 3x3



Filtrado average 5x5



❖ **Filtro promediador circular:** $H = \text{fspecial}(\text{'disk'}, r)$

Circulo de radio r , dentro de un cuadrado de tamaño $2*r+1$ (default $r=5$)

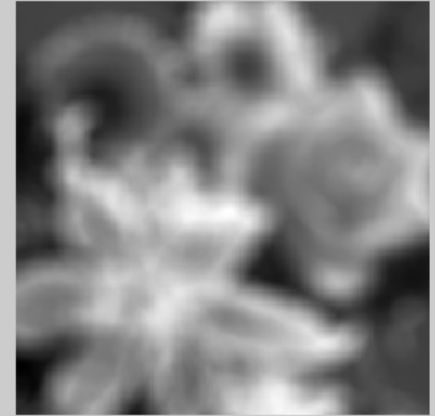
Imagen original



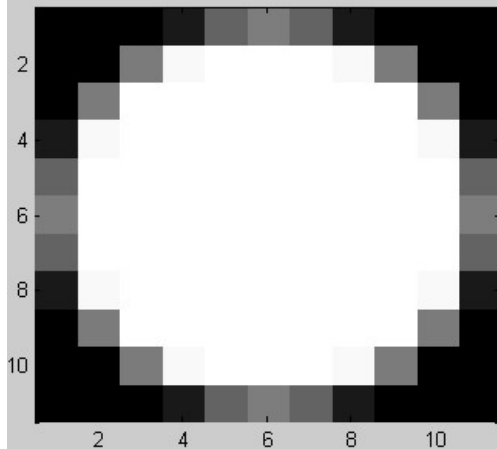
Filtro disk de radio $r=5$



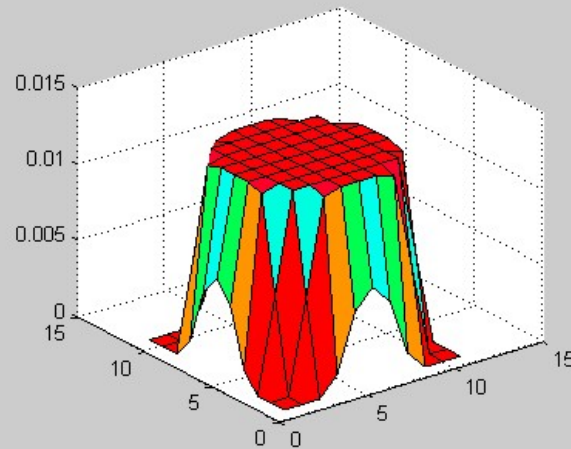
Filtro disk de radio $r=10$



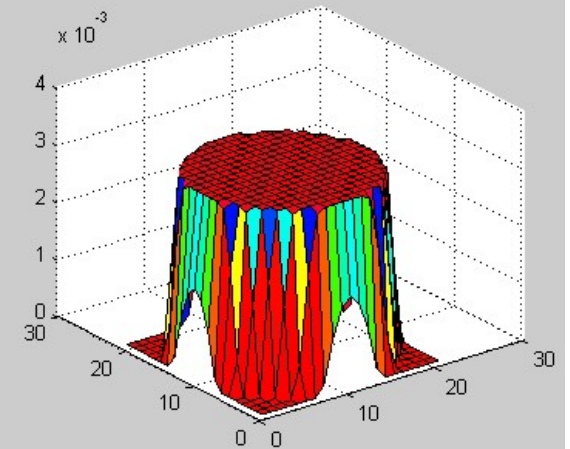
Filtro disk de radio $r=5$



Disco $r=5$



Disco $r=10$



❖ **Filtro Guassiano:** `H = fspecial ('gaussian', [r c], sigma)`

sigma: desvío estándar

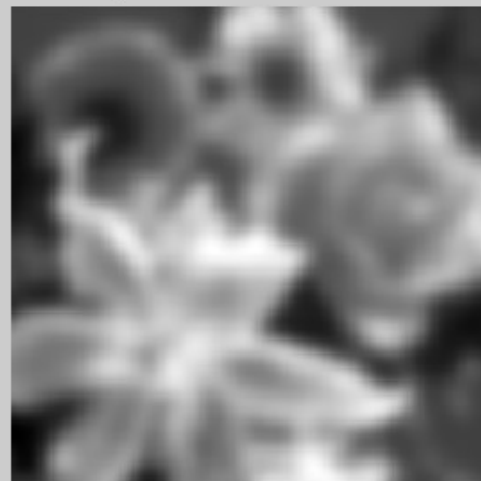
Imagen original



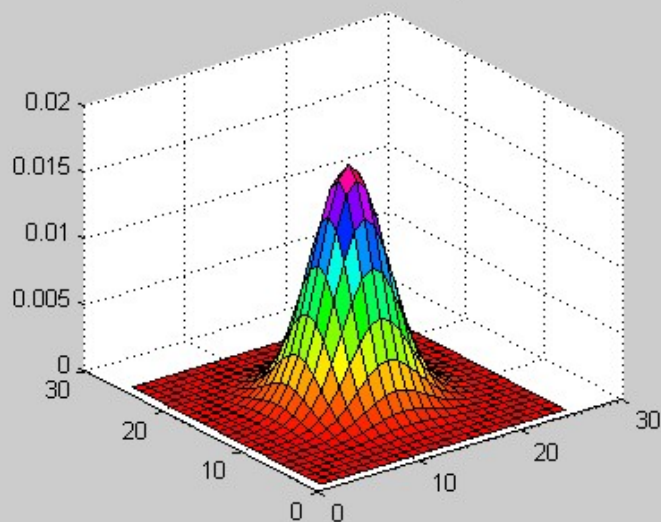
Filtro 'gaussian' 25x25 $\sigma = 3$



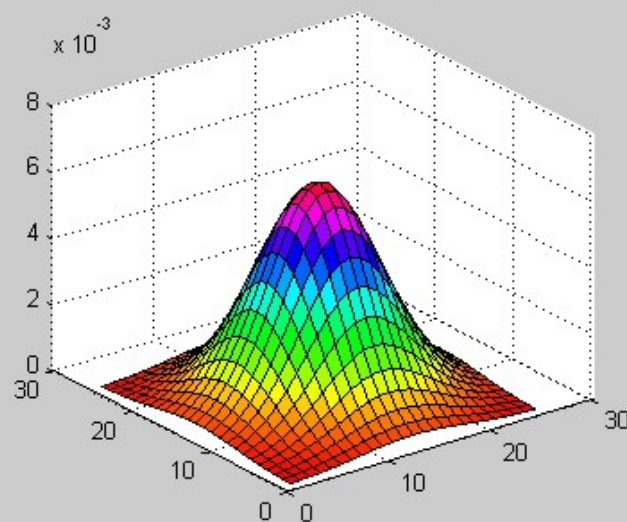
Filtro 'gaussian' 25x25 $\sigma = 5$



Gaussiana 25x25, $\sigma = 3$



Gaussiana 25x25, $\sigma = 5$



Filtros basados en la primera y segunda derivada

La derivada primera (**gradiente**) de una función $f(x,y)$ en la posición (x,y) se define:

$$\nabla f(x,y) = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}$$

Las derivadas pueden definirse en términos de **diferencias**:
ver **aproximación numérica del gradiente**: Teorema de Taylor-LaGrange

$$\frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y) \quad \frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y)$$

Observación:

- ❖ $\nabla f(x,y) = 0 \rightarrow$ en las zonas que tienen un nivel de gris constante
- ❖ $\nabla f(x,y) \neq 0 \rightarrow$ ante un cambio del tipo escalón o rampa en el nivel de gris

- ❑ El requisito básico de un operador de derivación es que la **suma de sus coeficientes** sea **cero**.
- ❑ El **operador Sobel** es uno de los filtros que permite determinar derivadas primeras
- ❑ Obs: la derivación **acentúa el ruido**

Operador de Roberts

$$\frac{\partial f}{\partial x} \cong f \otimes \begin{bmatrix} 1 & -1 \end{bmatrix} \quad \frac{\partial f}{\partial y} \cong f \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \frac{\partial f}{\partial v} \cong f \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{pour } \alpha = \frac{\pi}{4}$$

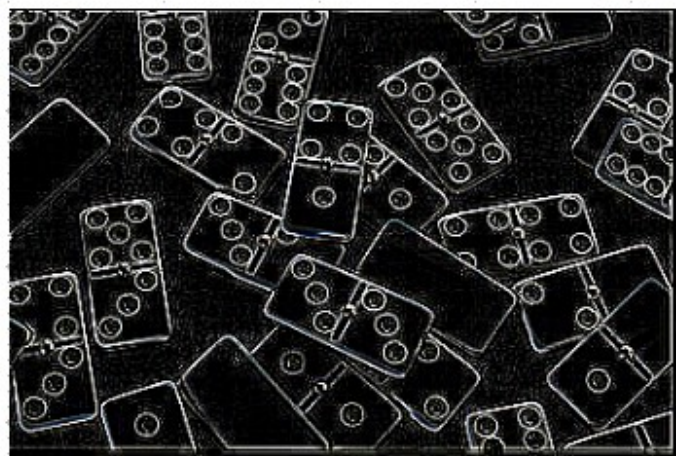
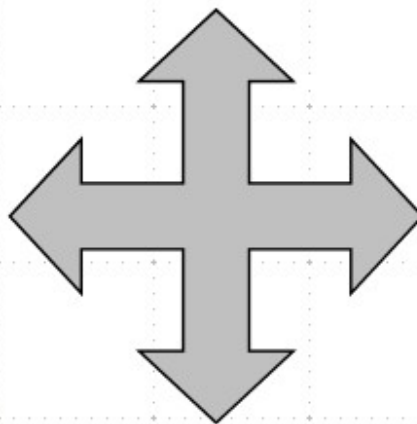
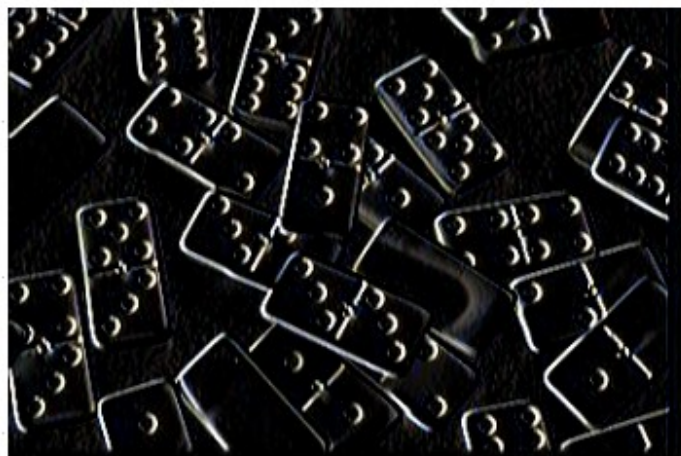


- Buena respuesta ante bordes verticales, horizontales y diagonales (según los diferentes kernels).
- Extremada sensibilidad al ruido.

$$\frac{\partial f}{\partial x} \cong f \otimes \begin{bmatrix} 1 & -1 \end{bmatrix}$$



$$\frac{\partial f}{\partial y} \cong f \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$



$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

❖ Filtro Sobel: $H = \text{fspecial}('sobel')$

Detecta gradientes verticales. Para gradientes horizontales hace falta rotar la máscara. El filtro H tiene dimensión 3x3.

$H1 = \text{fspecial}('sobel') \%S$

$H2 = \text{rot90}(H1) \%E$

$H3 = \text{rot90}(H1, 2) \%N$

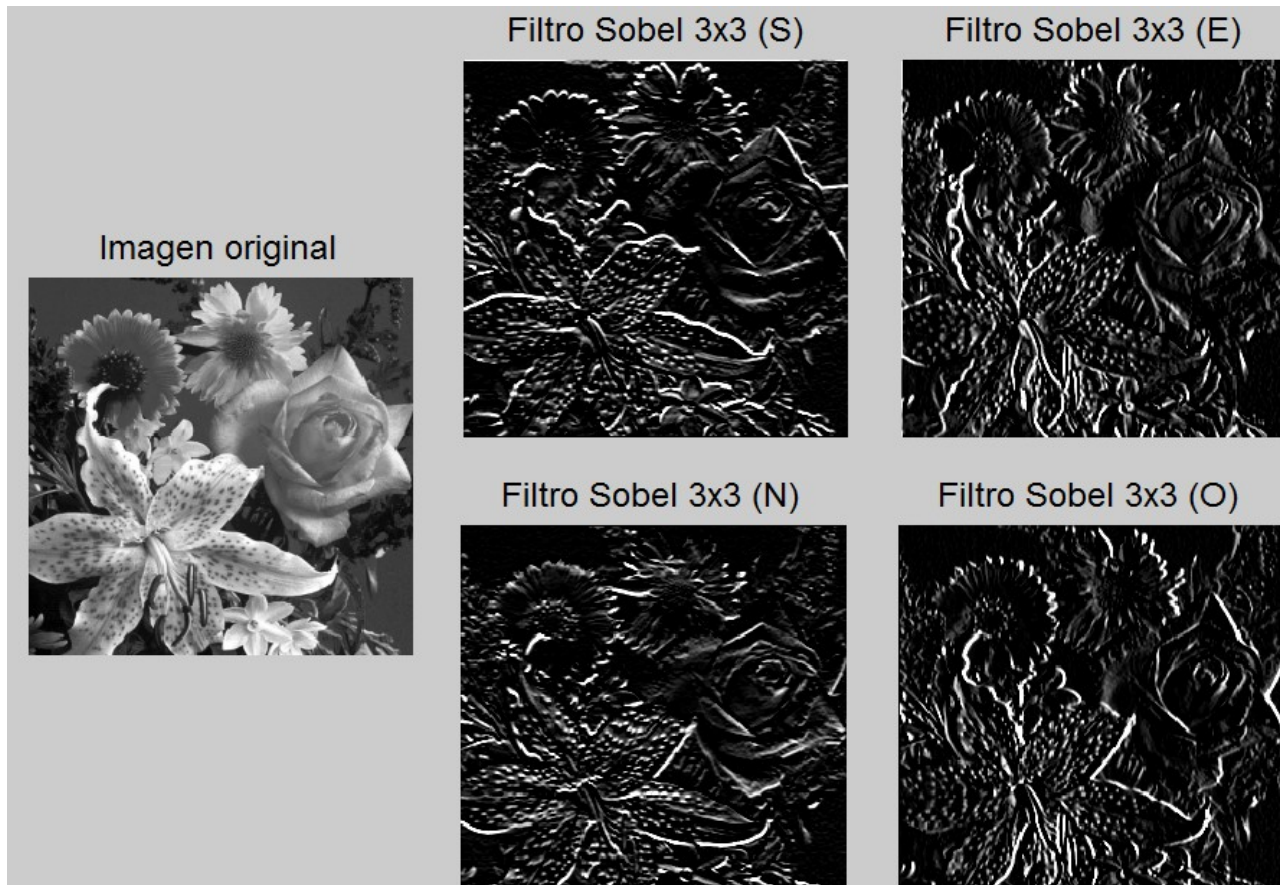
$H4 = \text{rot90}(H1, 3) \%O$

$$H1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$H2 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$H3 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$H4 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



❖ Filtro de Prewitt: $H = \text{fspecial}(\text{'prewitt'})$

Por defecto H detecta gradientes verticales. Rotando la matriz se consigue detectar gradientes en otras direcciones. Máscara de dimensión 3x3

Prewitt 3x3 (E)

$$H = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Prewitt 3x3 (O)

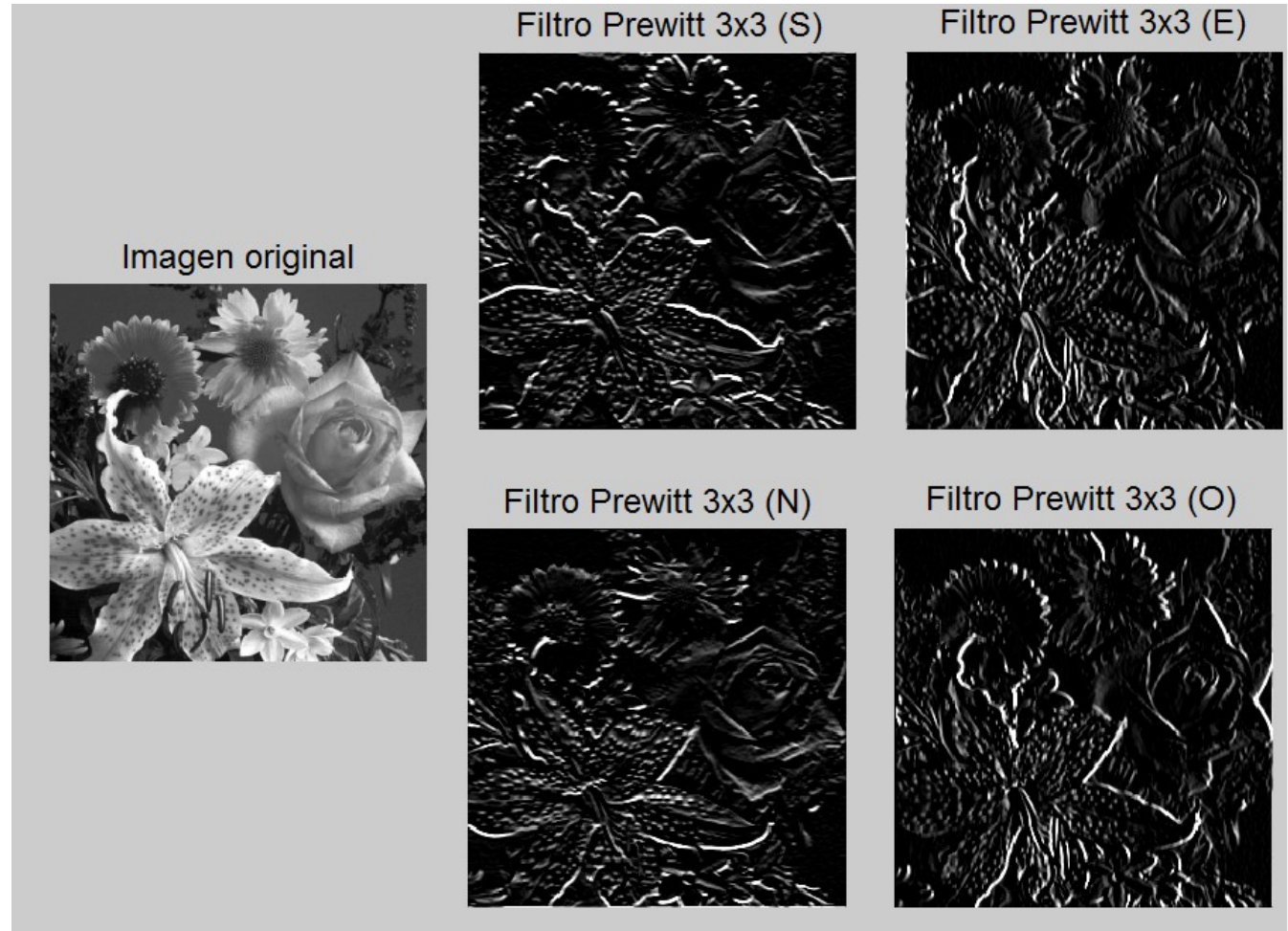
$$H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Prewitt 3x3 (N)

$$H = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Prewitt 3x3 (S)

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

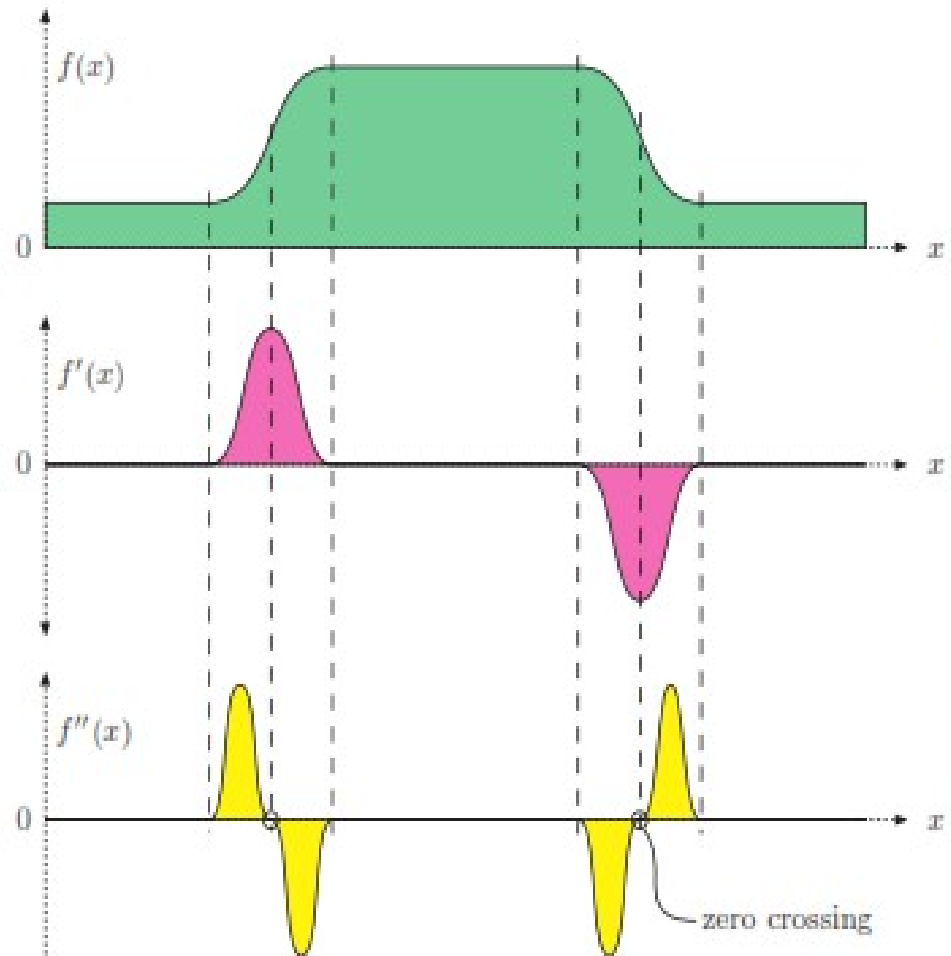


Derivada segunda:

- ❑ $\nabla^2 f(x,y) = 0 \rightarrow$ en zonas con nivel de gris constante
- ❑ $\nabla^2 f(x,y) \neq 0 \rightarrow$ al principio o al final de un cambio del tipo escalón o rampa
- ❑ $\nabla^2 f(x,y) = 0 \rightarrow$ zona que tiene un cambio del tipo rampa con pendiente cte.

Detección de bordes:

- ❑ Los puntos del borde están localizados donde la **segunda derivada cruza por cero** y la primera derivada tiene una magnitud máxima.
- ❑ Filtro Laplaciano: operador de derivación (*suma coeficientes = cero*).
- ❑ Desventaja: es muy sensible a la presencia de ruido.



Filtro Laplaciano

- ❑ Filtro **isotrópico** (invariante frente a la rotación).
- ❑ El coeficiente asociado con el píxel central debe ser positivo y todos los demás negativos o cero, o viceversa. La suma de los coeficientes debe ser cero.
- ❑ Detecta bordes en todas las direcciones

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$

Aproximación de la segunda derivada

$$\frac{\partial^2 f(x, y)}{\partial x^2} = f(x + 1, y) + f(x - 1, y) - 2f(x, y)$$

$$\frac{\partial^2 f(x, y)}{\partial y^2} = f(x, y + 1) + f(x, y - 1) - 2f(x, y)$$

$$\nabla^2 f(x, y) = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y)$$

$$\text{Laplaciano V4} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\text{Laplaciano V8} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{Laplaciano V4 bis} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\text{Laplaciano V8 bis} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

❖ **Filtro Laplaciano:** detecta bordes en todas direcciones

Imagen original



Filtro Laplaciano V4 3x3



Filtro Laplaciano V8 3x3



H1 =

0 1 0

1 -4 1

0 1 0

H2 =

-1 -1 -1

-1 8 -1

-1 -1 -1

Realce de bordes usando la segunda derivada

(edge sharpening or enhancement)

$$O(x, y) = F(x, y) + c[\nabla^2 F(x, y)]$$

$c=1$ si el coef. central del filtro es positivo, sino, $c=-1$

1) Convierto la imagen a double
 $F = \text{im2double}(F);$

2) Defino los filtros

$H4 = [0 \ 1 \ 0; 1 \ -4 \ 1; 0 \ 1 \ 0];$

$H8 = [-1 \ -1 \ -1; -1 \ 8 \ -1; -1 \ -1 \ -1];$

3) Aplicar el filtro

$G1 = \text{conv2}(F, H4, 'same');$

$G2 = \text{conv2}(F, H8, 'same');$

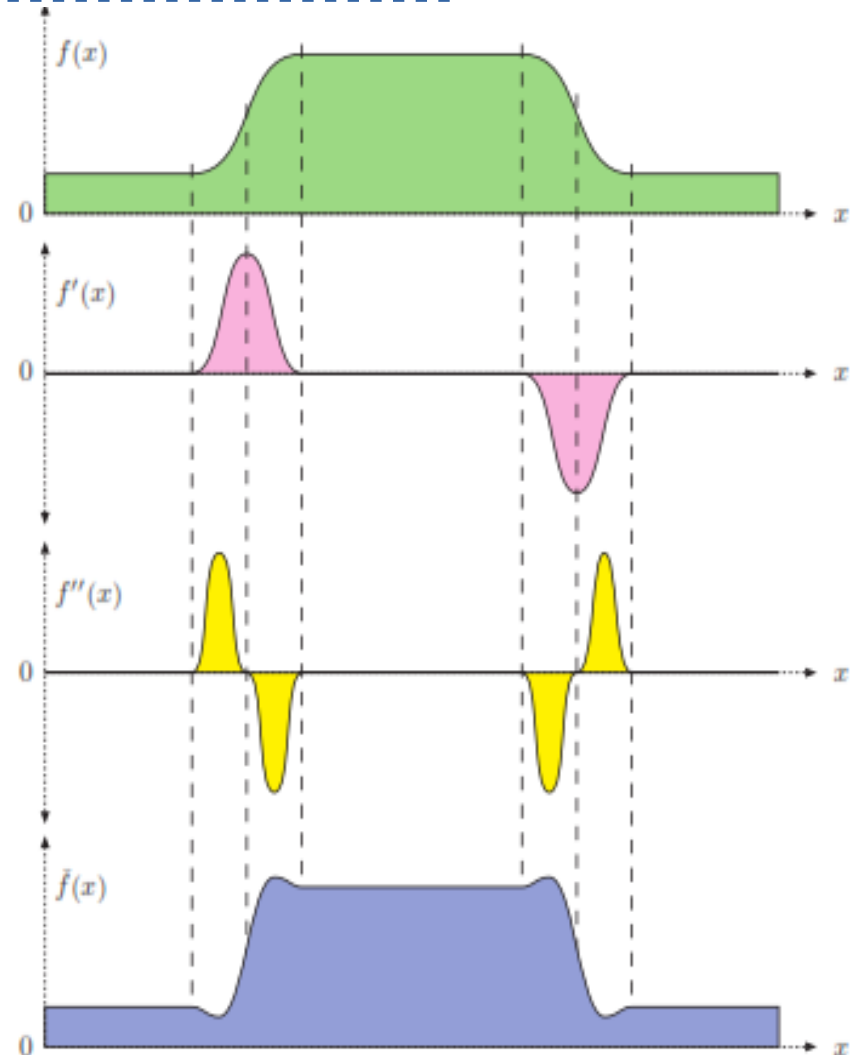
4) Sumo la imagen original y la filtrada para realzar

$O1 = F - G1;$

$O2 = F + G2;$

%se restauran los niveles de gris que se perdieron
al aplicar el filtro

Ejercicio N°5: Realce los bordes de una imagen
usando la segunda derivada.



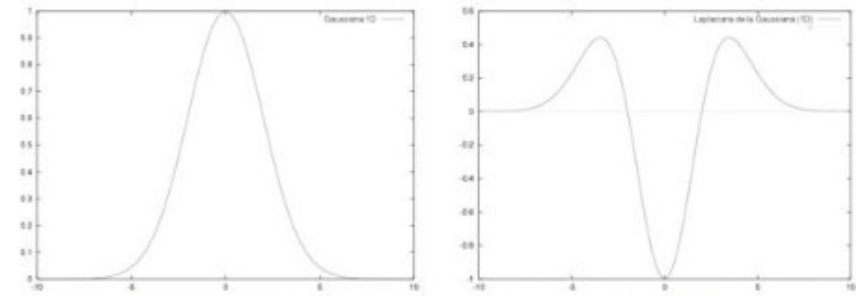
❖ Filtro Laplaciano de una Gaussiana (LoG):

$H = \text{fspecial}('log', [r \ c], \text{sigma})$

Donde $[r \ c]$ es el tamaño del filtro (default 5x5) y sigma el desvío estándar (default 0.5). Es un detector de bordes.

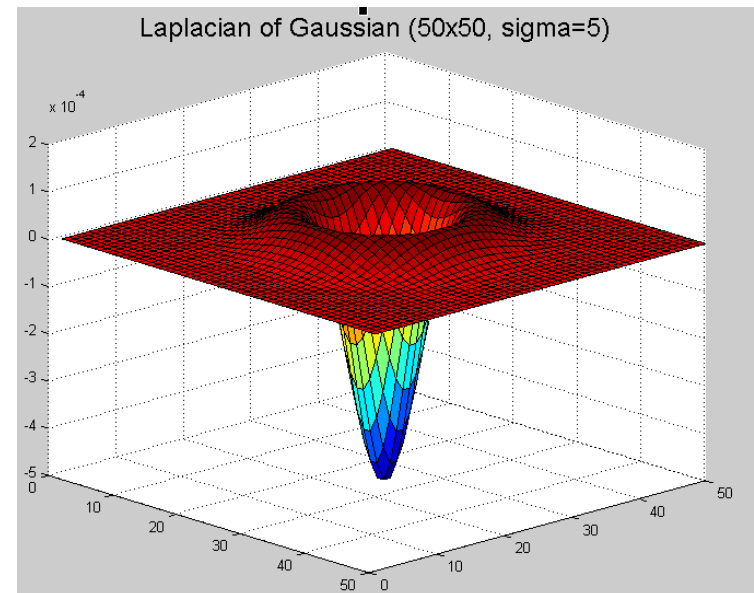
- 2ª derivada de la Gaussiana (Marr-Hildreth, 1980)
- = Laplaciana de una gaussiana (LoG)
- = Operador de Marr-Hildreth
- = Sombrero Mejicano

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2} \right) e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

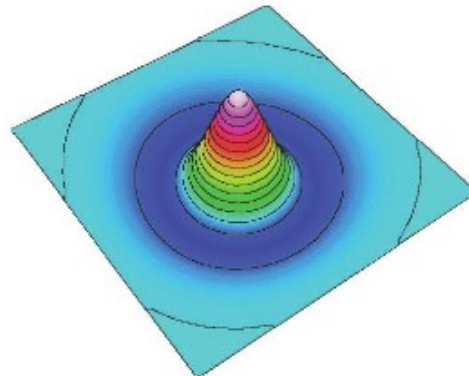
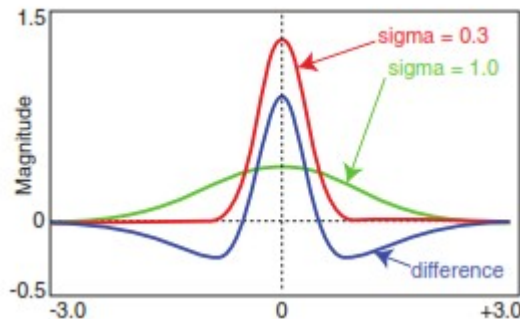


Gaussiana 1D

LoG 1D



❖ Filtro Diferencia de Gaussianas (DoG):



DoG es una aproximación del filtro LoG. Se restan dos Gaussianas de distinto σ

❖ Filtro movimiento: $H = f_{\text{special}}('motion', len, theta)$

Crea sensación de movimiento en una imagen de *len* píxeles. La dirección del movimiento viene dada por *theta*.



Ejercicio 6: Aplíquelo a una imagen los diferentes filtros vistos (siempre que sea posible, use *fspecial*).

Filtros espaciales no-lineales

- ☐ Se basan en operaciones no lineales involucrando píxeles de una vecindad.
- ☐ Ejemplo: buscar el valor máximo dentro de la vecindad y asignar ese valor al pixel central de dicha vecindad.
- ☐ Cuando veamos degradación/restauración de imágenes entraremos en más detalles.

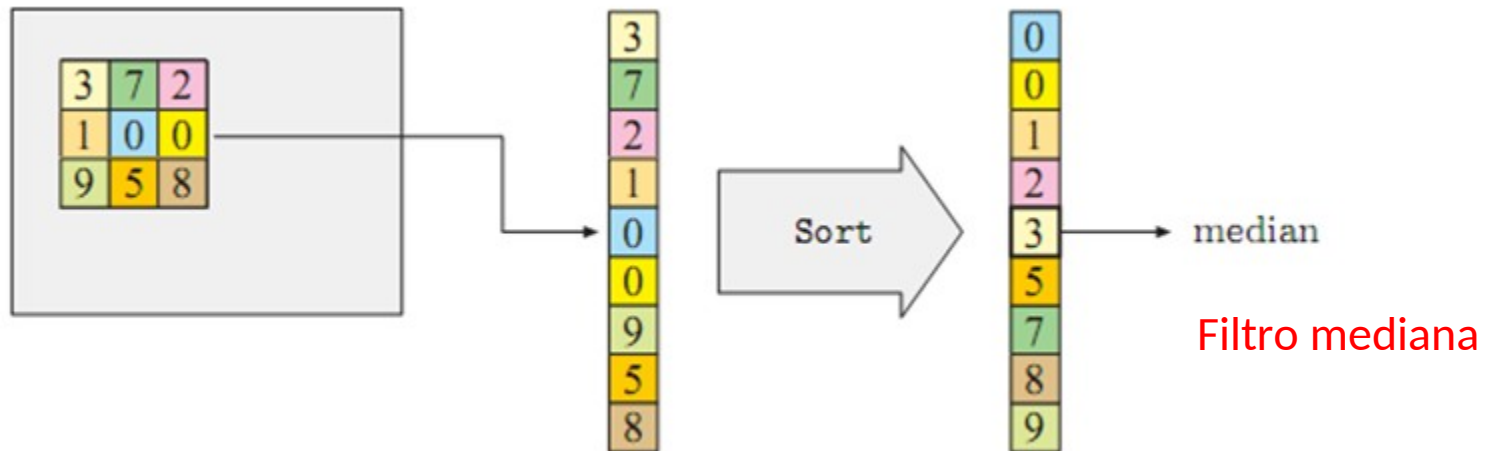
Filtro de mediana

$O = \text{medfilt2}(I, [m \ n])$

$[m \ n]$ define el tamaño de la máscara (default 3x3).

La mediana se calcula ordenando los valores de los pixeles dentro de la ventana de observación (de menor a mayor) y seleccionando el valor del medio.

Ese valor reemplazará al pixel de la imagen correspondiente al centro de la ventana.



Ruido sal y pimienta: pasa bajos vs mediana

Ejercicio 7: Abra una imagen y agréguele ruido sal y pimienta. Aplíquese dos filtros, uno pasa bajos y otro no lineal (mediana). Observe como trabaja cada filtro. Modifique la dimensión del filtro mediana y la densidad de puntos con ruido.



filtrado promediador 3x 3



*Imagen original con
ruido sal y pimienta*



fltro de mediana 3x3

- ☐ El filtro mediana elimina el ruido tipo "sal y pimienta" (salt & pepper) y los bordes se conservan relativamente bien.
- ☐ **Resultados no satisfactorios** cuando el nº píxeles con ruido en la ventana de análisis es mayor o igual a la mitad del nº píxeles en la ventana.