

PROCESAMIENTO DE IMÁGENES DIGITALES

DPTO. MATEMÁTICA APLICADA I

# Tema 3: FILTRADO EN EL DOMINIO ESPACIAL



## ÍNDICE:

- Nociones básicas
- Concepto de ruido
- Dominio espacial:  
Técnica de filtrado

## Tema 3: Filtrado en el dominio espacial

- **Suavizar la imagen:**

Reducir la cantidad de variaciones de intensidad entre píxeles vecinos.

- **Eliminar ruido:**

Restaurar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos.

- **Realzar los detalles:**

Aumentar las variaciones de intensidad existentes.

- **Detectar bordes:**

Detectar píxeles donde se produce un cambio brusco en la función de intensidad.

## Tema 3: Filtrado en el dominio espacial

- **RUIDO:**

- Información no deseada que contamina la imagen.
- El origen del ruido puede estar:
  - en el proceso de adquisición,
  - en el proceso de transmisión,
  - en el procesamiento de la imagen.



## Tema 3: Filtrado en el dominio espacial

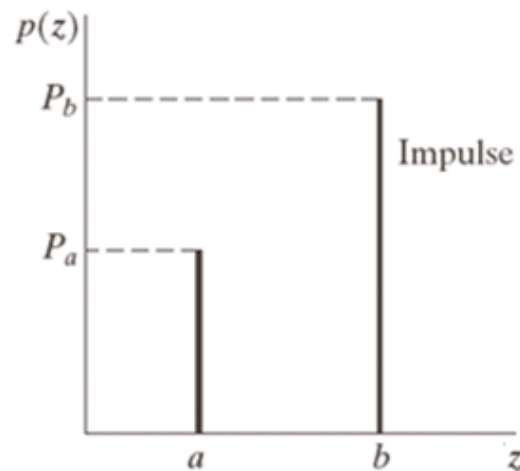
- **TIPOS DE RUIDO:**

- Impulsivo / sal y pimienta.
- Ruido aditivo.

## Tema 3: Filtrado en el dominio espacial

- **Ruido impulsivo o sal y pimienta:**

- Ciertos píxeles de la imagen poseen unos niveles de intensidad que difieren bastante de los niveles de los píxeles de su entorno.
- Suele aparecer en forma de impulsos blancos y negros (sal y pimienta).



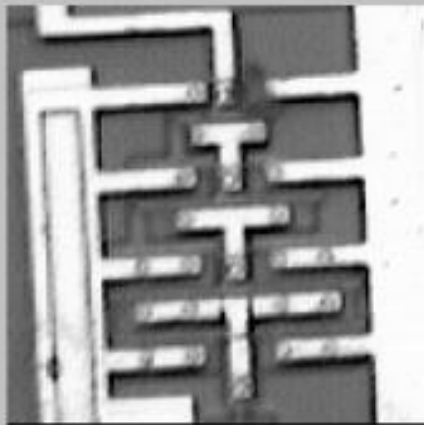
## Tema 3: Filtrado en el dominio espacial

- **Ruido impulsivo / sal y pimienta:**

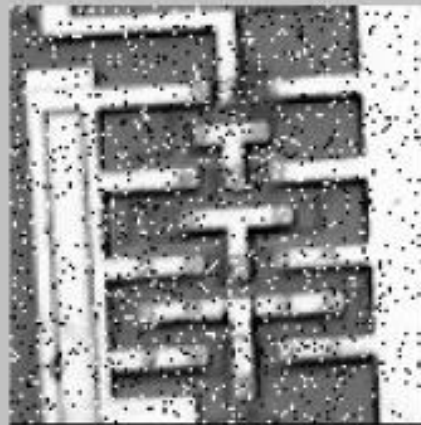


## Tema 3: Filtrado en el dominio espacial

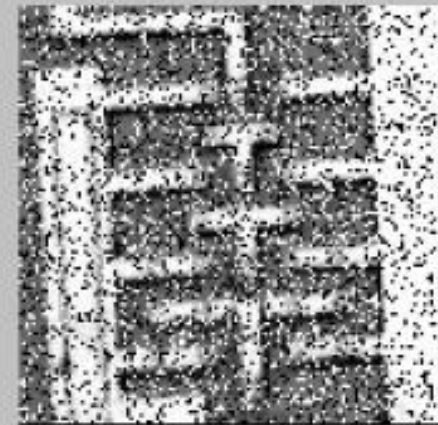
- **Ruido impulsivo / sal y pimienta:**



Original



R. Impulsional  
10%



R. Impulsional  
30%



## Tema 3: Filtrado en el dominio espacial

- **Ruido aditivo:  $r(x,y)$**

- Es independiente de la señal de la imagen  $f(x,y)$

$$g(x,y)=f(x,y)+r(x,y)$$

( $f$  y  $r$  son funciones independientes).

- Se habla de distintos tipos de ruido aditivo según su distribución de probabilidad (gaussiano, exponencial, uniforme, etc.)

## Tema 3: Filtrado en el dominio espacial

- **Ruido gaussiano:**

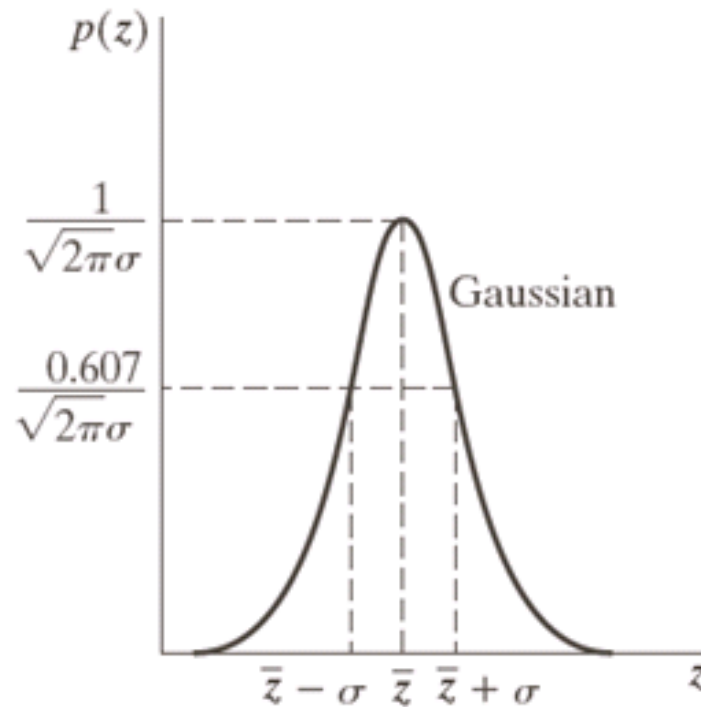
- Modela el ruido producido por los circuitos electrónicos o ruido de los sensores por falta de iluminación y/o altas temperaturas.
- Afecta a la imagen completa, es decir, la intensidad de todos los píxeles se ve afectada, añadiéndole un valor que sigue una distribución normal o gaussiana:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$$

donde  $z$  representa los niveles de intensidad,  $\bar{z}$  el valor medio de  $z$  (normalmente 0) y  $\sigma$  su desviación estándar.  $\sigma^2$  es la varianza.

## Tema 3: Filtrado en el dominio espacial

- **Ruido gaussiano:**



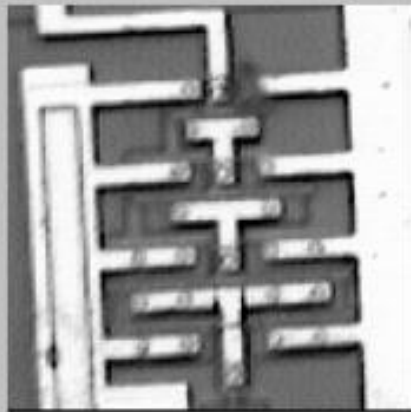
## Tema 3: Filtrado en el dominio espacial

- **Ruido gaussiano:**

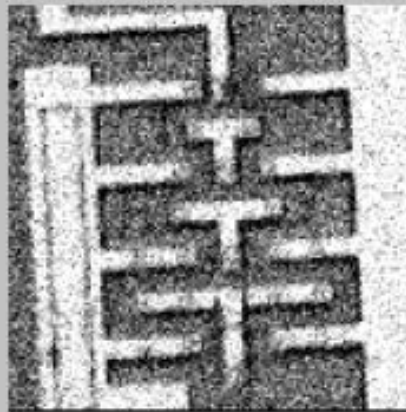


## Tema 3: Filtrado en el dominio espacial

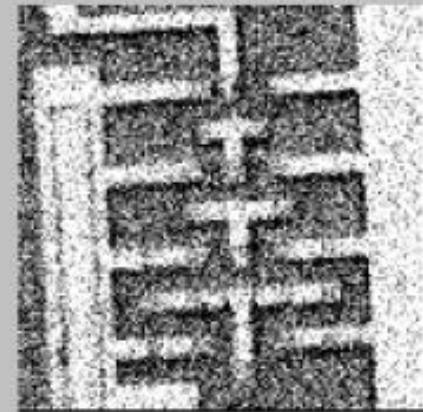
- **Ruido gaussiano:**



Original



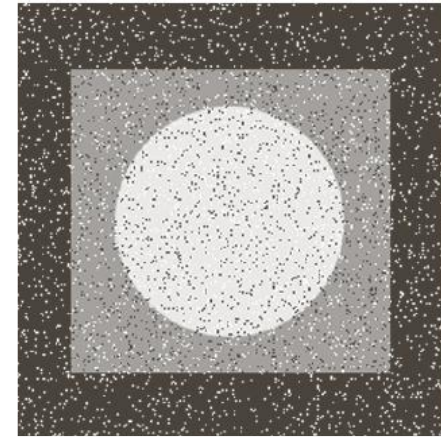
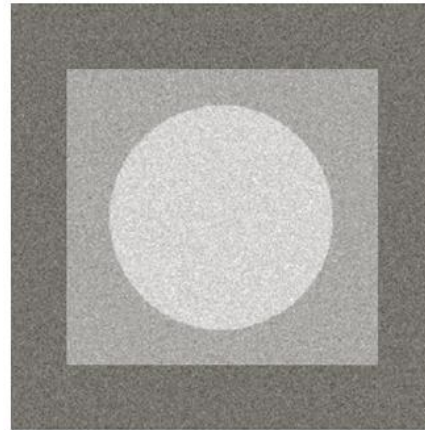
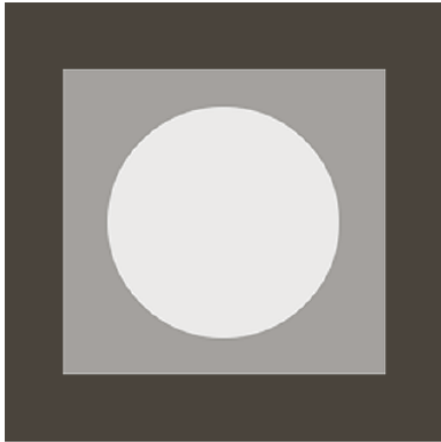
R. Gaussiano  
 $s=0,025$



R. Gaussiano  
 $s=0,05$

## Tema 3: Filtrado en el dominio espacial

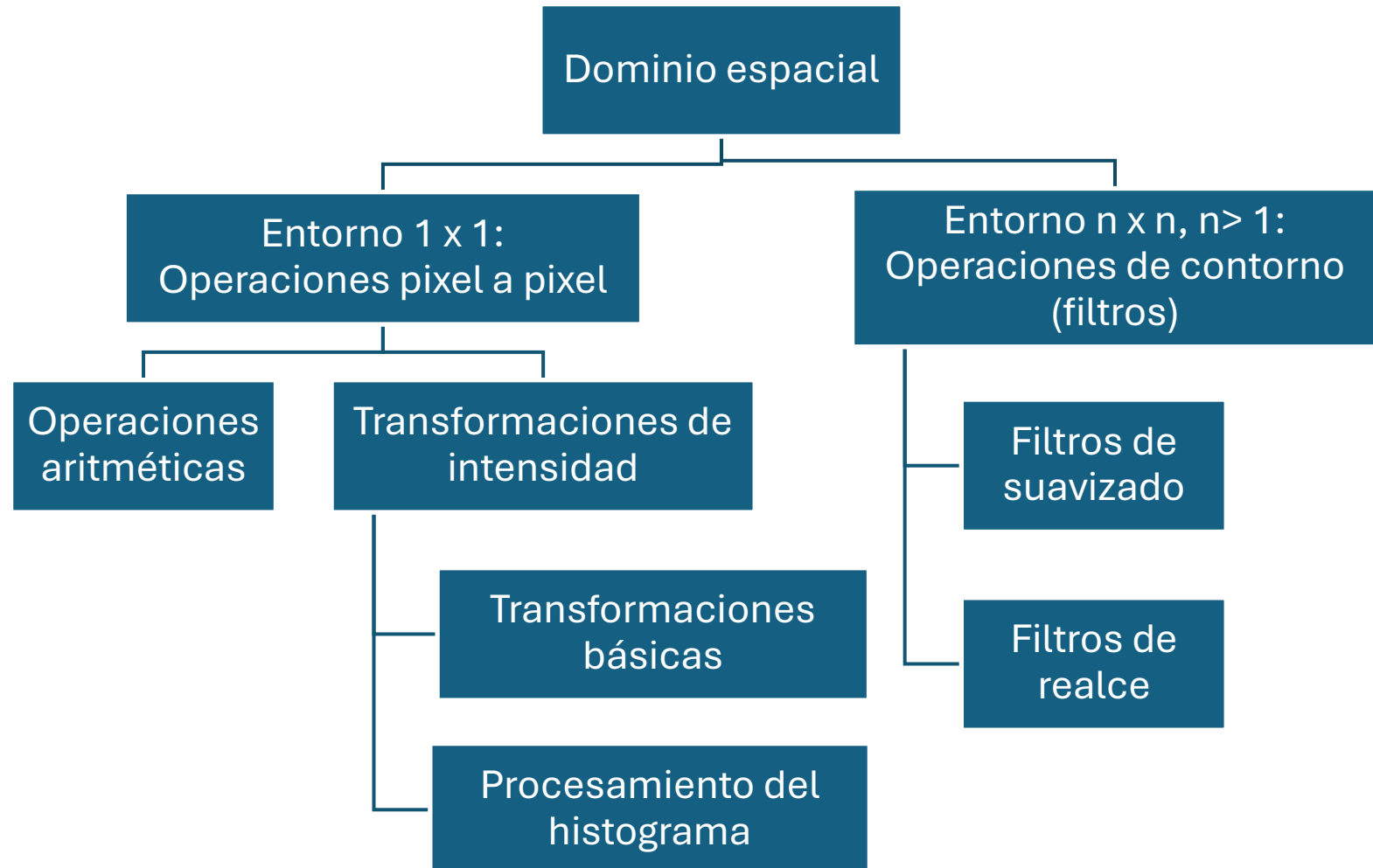
- **Ruido gaussiano vs ruido sal y pimienta:**



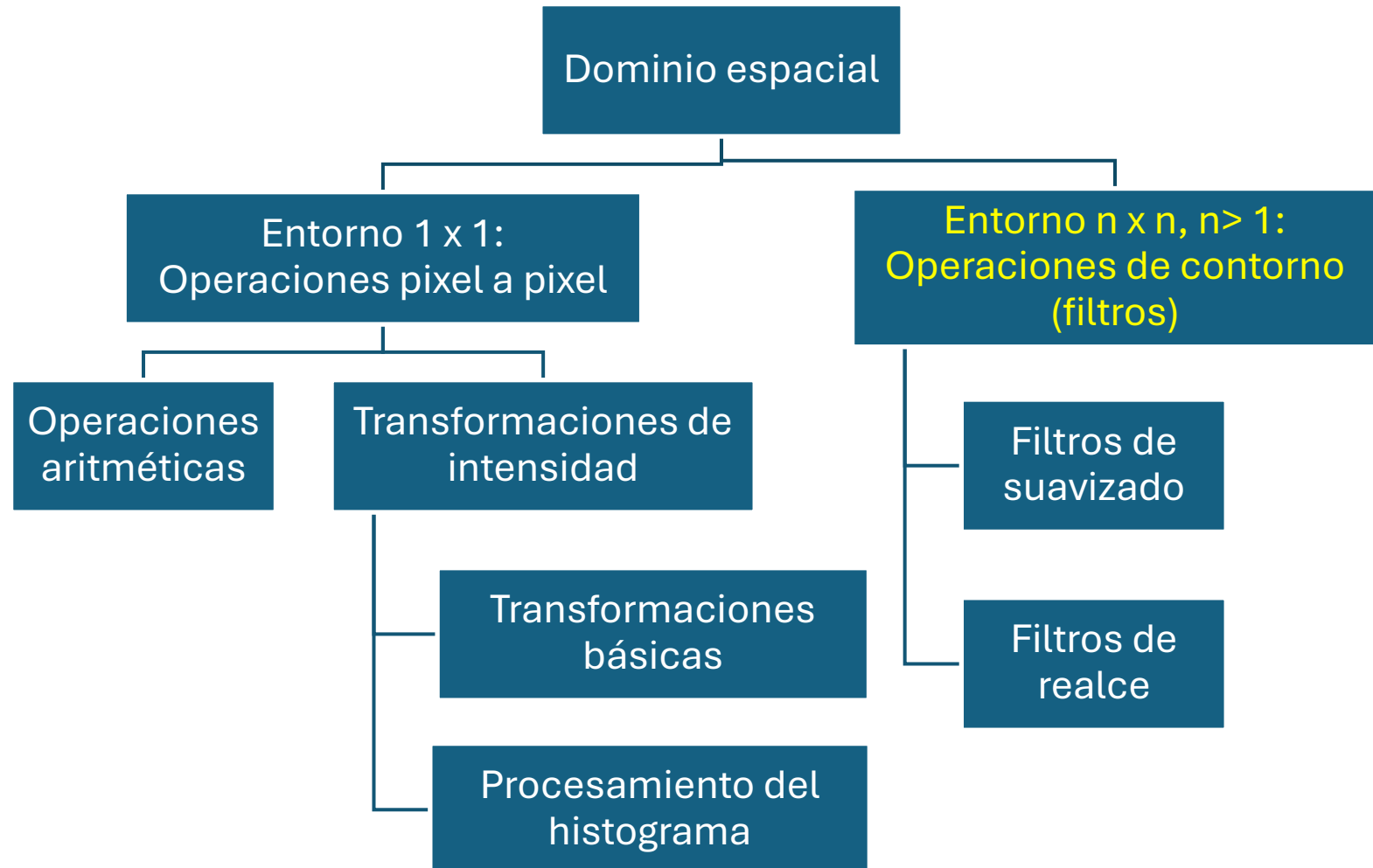
## Practica:

- Usa la función `randn(dst, mean, stddev)` ->dst de OpenCV para generar valores aleatorios de media “mean” y desviación estándar “stddev” para introducir ruido gaussiano en una imagen.
- Para introducir ruido sal y pimienta, se puede usar la función `randint()` de la librería `random` para decidir si un pixel se sustituye por blanco o negro con cierta probabilidad
- Ejemplo: <https://www.kaggle.com/code/chanduanilkumar/adding-and-removing-image-noise-in-python/notebook>

## Tema 3: Filtrado en el dominio espacial



## Tema 3: Filtrado en el dominio espacial





## Tema 3: Filtrado en el dominio espacial

- Las funciones de procesamiento de la imagen en el dominio espacial pueden expresarse como:

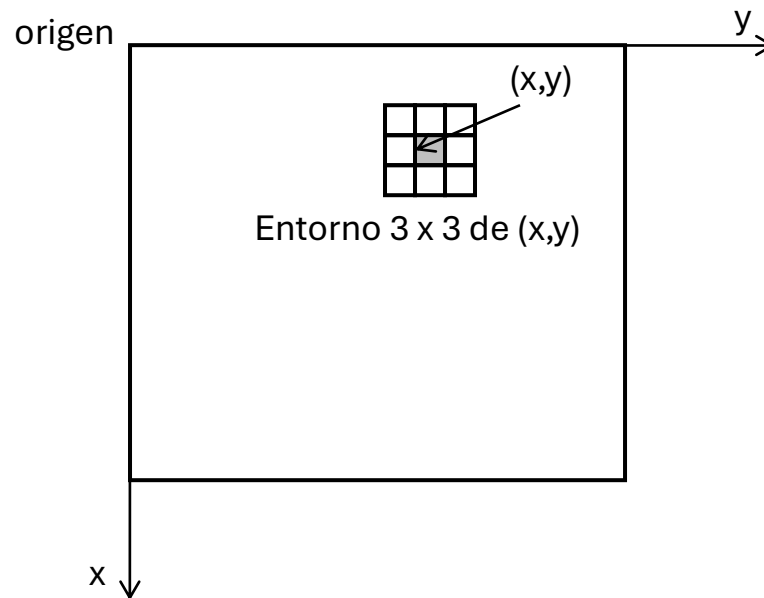
$$g(x,y) = T(f(x,y))$$

donde

- $f(x,y)$  es la imagen original
- $g(x,y)$  es la imagen procesada
- $T$  es un operador que actúa sobre  $f$  y se define en algún entorno de  $(x,y)$ .

## Tema 3: Filtrado en el dominio espacial

- **Entorno de un punto  $(x,y)$ :** subimagen cuadrada o rectangular centrada en el punto  $(x,y)$ .



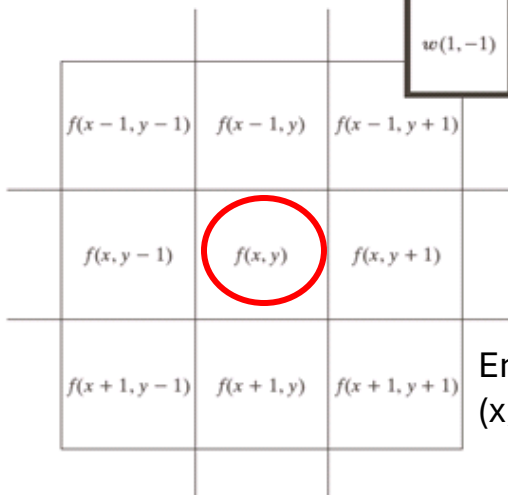
Pueden existir otros entornos, tales como aproximaciones de círculos, pero los cuadrados son los más comunes por ser los más fáciles de implementar.

## Tema 3: Filtrado en el dominio espacial

- **Técnica de filtrado en general:** Esquema del mecanismo de filtrado espacial usando un entorno 3x3

Máscara 3 x 3: matriz de coeficientes

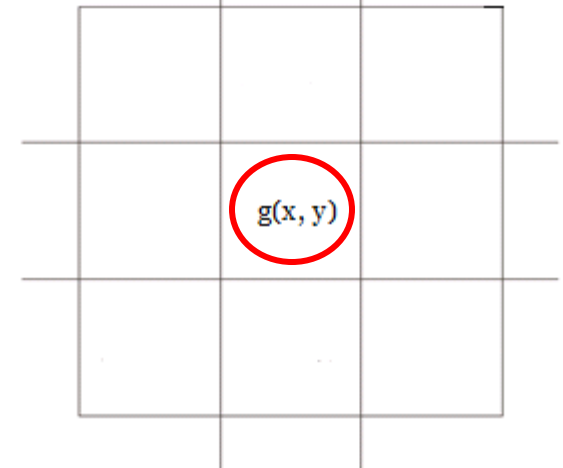
$w(-1, -1)$	$w(-1, 0)$	$w(-1, 1)$
$w(0, -1)$	$w(0, 0)$	$w(0, 1)$
$w(1, -1)$	$w(1, 0)$	$w(1, 1)$



Entorno 3 x 3 del pixel (x,y) con valor  $f(x,y)$



Resultado de interactuar la máscara con el entorno del píxel



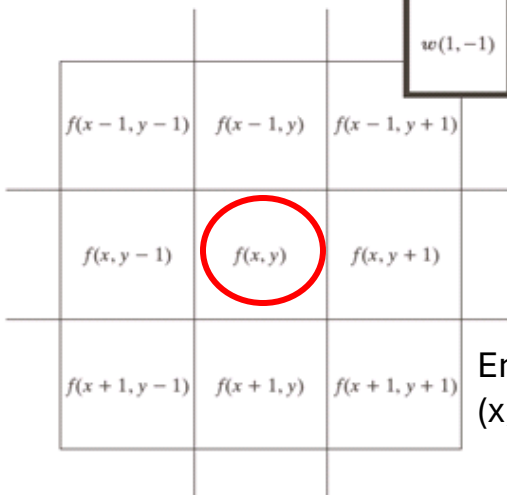
## Tema 3: Filtrado en el dominio espacial

- **Técnica de filtrado lineal:**

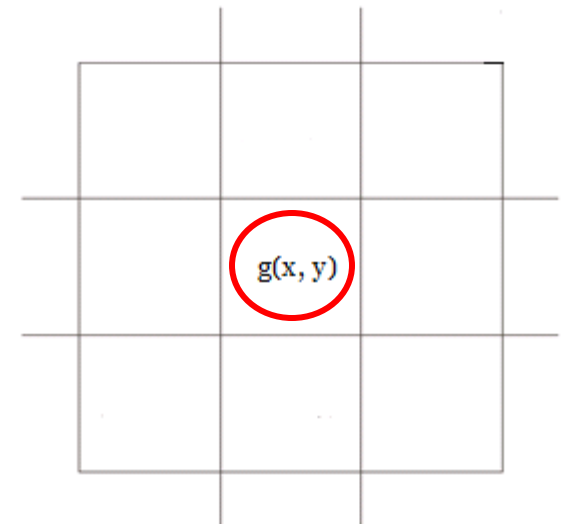
$$g(x, y) = \sum_{s=-1}^1 \sum_{t=-1}^1 w(s, t) f(x + s, y + t)$$

Máscara 3 x 3: matriz de coeficientes

$w(-1, -1)$	$w(-1, 0)$	$w(-1, 1)$
$w(0, -1)$	$w(0, 0)$	$w(0, 1)$
$w(1, -1)$	$w(1, 0)$	$w(1, 1)$



Entorno 3 x 3 del pixel (x,y) con valor  $f(x,y)$



## Tema 3: Filtrado en el dominio espacial

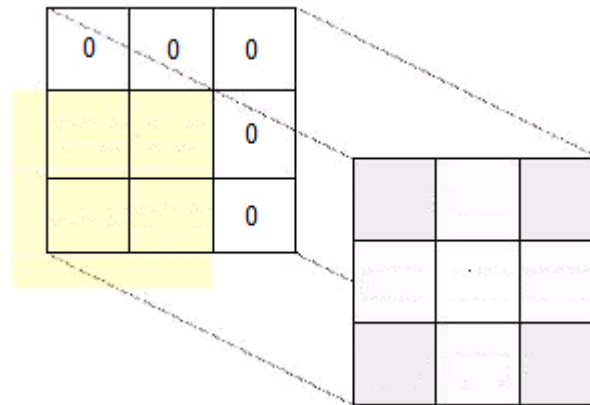
- **Técnicas de filtrado:**

### Practica:

- Image filtering in OpenCV  
[https://docs.opencv.org/4.x/d4/d86/group\\_\\_imgproc\\_\\_filter.html](https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html)
- Con la función `filter2D()` puedes crear tu propio filtro

## Tema 3: Filtrado en el dominio espacial

- **Límites de la imagen:** Podemos aplicar la máscara añadiendo un marco de ceros (padding) de la anchura adecuada. Esto puede tener efectos no deseados (p. ej., de difuminación en los límites de la imagen) pero, en general, poco significativos si la máscara es pequeña en relación con el tamaño de la imagen.



- Otras formas: duplicar el borde de la imagen, reflejarlo, no procesarlo.

## Practica:

- OpenCV tiene varias opciones para el argumento **BorderType**

[https://docs.opencv.org/4.x/d2/de8/group\\_core\\_array.html#gga209f2f4869e304c82d07739337eae7c5afe14c13a4ea8b8e3b3ef399013dbae01](https://docs.opencv.org/4.x/d2/de8/group_core_array.html#gga209f2f4869e304c82d07739337eae7c5afe14c13a4ea8b8e3b3ef399013dbae01)

# Filtros de suavizado



## Tema 3: Filtrado en el dominio espacial

- **FILTROS DE SUAVIZADO: ¿Para qué se usan?**
  - Emborronar la imagen:
    - Eliminar pequeños detalles antes de la segmentación de un objeto de interés.
    - Rellenar pequeños espacios.
  - Eliminar ruido.



## Tema 3: Filtrado en el dominio espacial

- **FILTROS DE SUAVIZADO: Clasificación**
  - Filtros lineales
    - Filtro de la media
    - Filtro gaussiano
  - Filtros no lineales
    - Filtros estadísticos ordenados

## Tema 3: Filtrado en el dominio espacial

- **FILTRO DE LA MEDIA (Mean filter):**

- **Operación:** Se reemplaza el valor de la cada píxel por la media de los valores de los píxeles vecinos. Se puede operar mediante convolución con una máscara determinada.

- **Máscara 3 x 3:**

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

- **Ejercicio:** ¿Cómo sería la máscara 5 x 5 para este filtro?

## Tema 3: Filtrado en el dominio espacial

- FILTRO DE LA MEDIA:**

- **Ejemplo:** (Sin procesar el marco exterior)

5	1	1	3	2	3
6	4	0	2	3	2
7	7	0	3	3	1
9	7	0	1	1	1
8	8	0	1	1	0
7	7	0	1	0	0

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



5	1	1	3	2	3
6	3	2	2	2	2
7	4	3	1	2	1
9	5	3	1	1	1
8	5	3	0	0	0
7	7	0	1	0	0

## Practica:

- Usa las funciones de OpenCV `blur()` y `boxFilter()`

## Tema 3: Filtrado en el dominio espacial

- **FILTRO DE LA MEDIA:**

- Los coeficientes deben ser todos iguales y sumar 1.
- Es el más simple, intuitivo y fácil de implementar.
- Puede crear nuevas intensidades de grises que no aparecían en la imagen.
- A mayor tamaño de la máscara, mayor emborronamiento.

## Tema 3: Filtrado en el dominio espacial

- **FILTRO DE LA MEDIA:**

- **Aplicaciones:**

- Suavizar imágenes, es decir, reducir la cantidad de variaciones de intensidad entre píxeles vecinos.

- Conseguir que las intensidades de los objetos pequeños se mezclen con el fondo con el fin de detectar los objetos de mayor tamaño.

- Eliminar ruido.

## Tema 3: Filtrado en el dominio espacial

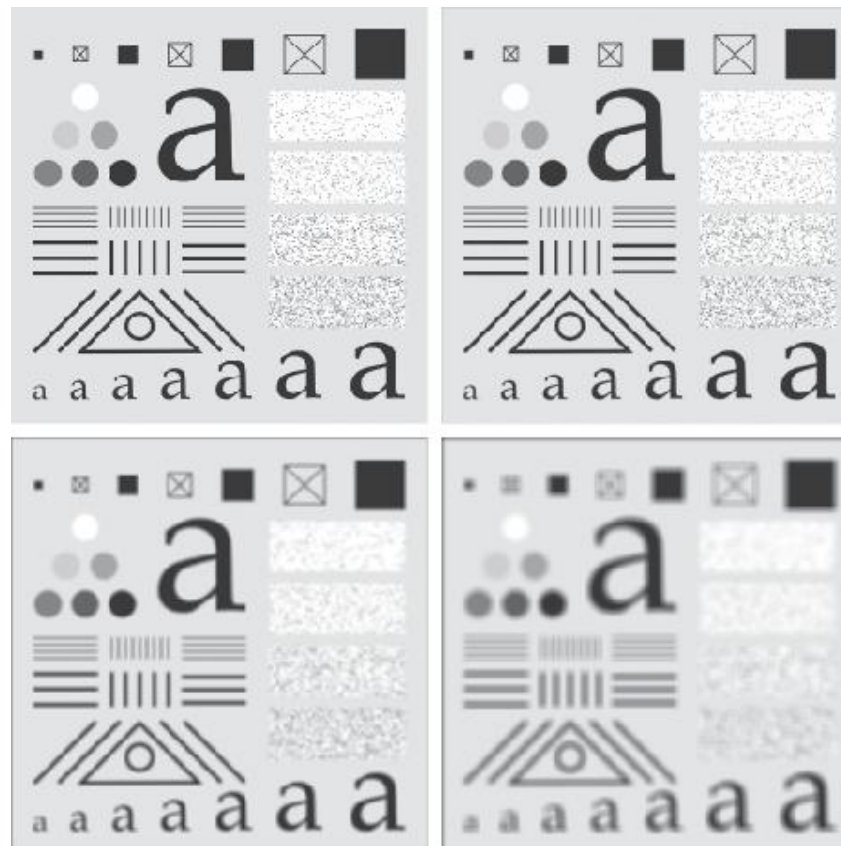
- FILTRO DE LA MEDIA:**

a b  
c d

**FIGURE 3.39**

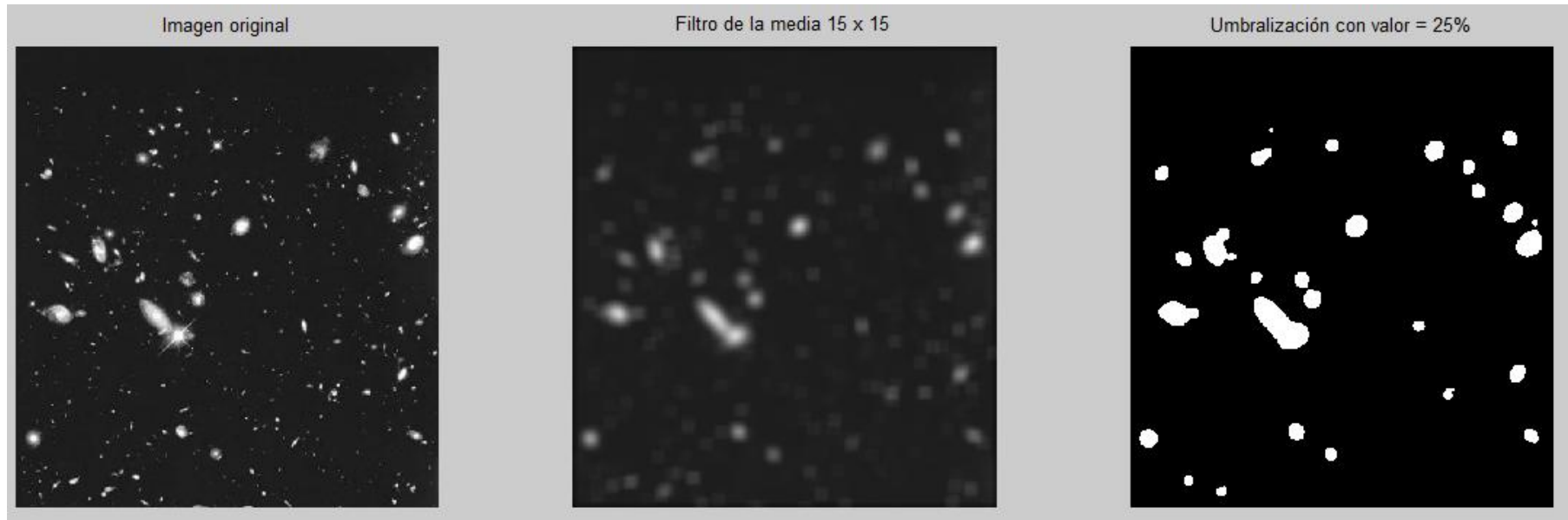
(a) Test pattern of size  $1024 \times 1024$  pixels.

(b)-(d) Results of lowpass filtering with box kernels of sizes  $3 \times 3$ ,  $11 \times 11$ , and  $21 \times 21$ , respectively.



## Tema 3: Filtrado en el dominio espacial

- FILTRO DE LA MEDIA:**



## Tema 3: Filtrado en el dominio espacial

Imagen original



Ruido gaussiano media 0 y varianza 0.01



Filtro de la media 3 x 3



Filtro de la media 5 x 5





## Tema 3: Filtrado en el dominio espacial

- **FILTRO GAUSSIANO (Gaussian filter):**

- **Operación:** Función gaussiana

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

- **Ejemplo de máscara 5 x 5 que aproxima la función gaussiana con  $\sigma = 1$ :**

1/273 x

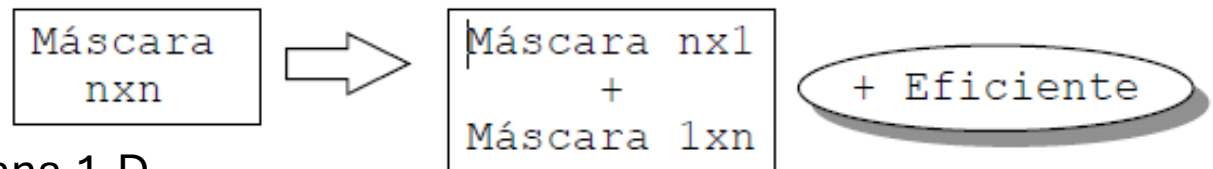
1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

## Tema 3: Filtrado en el dominio espacial

- **FILTRO GAUSSIANO:**

- **Características:**

- Efecto similar al filtro de la media, emborronando la imagen, pero con un resultado más “natural” ya que se le da más peso al píxel central.
- Todos los coeficientes siguen sumando 1.
- Es un *filtro separable*, es decir, en lugar de usar una máscara bidimensional, es posible usar dos máscaras 1-dimensionales; una vertical y otra horizontal, lo cual es más eficiente computacionalmente.



Ejemplo de máscara gaussiana 1-D con la que tendríamos el mismo resultado que con la máscara 5x5 anterior :

.006	.061	.242	.383	.242	.061	.006
------	------	------	------	------	------	------

## Tema 3: Filtrado en el dominio espacial

- FILTRO GAUSSIANO (Gaussian filter):**

- Operación:** Función gaussiana

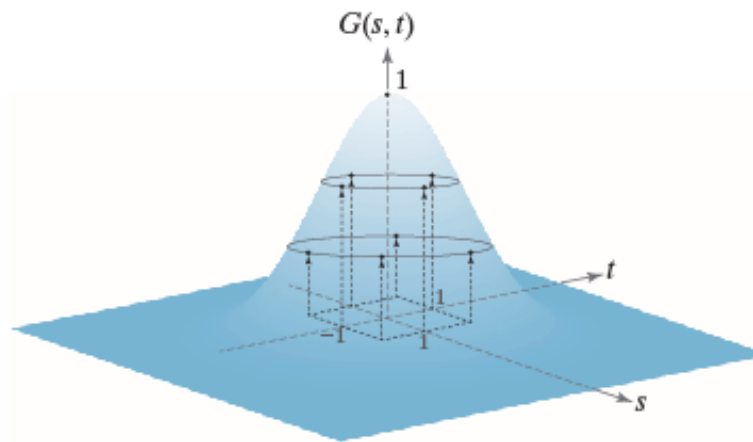
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Discretización**

a b

**FIGURE 3.41**

(a) Sampling a Gaussian function to obtain a discrete Gaussian kernel. The values shown are for  $K = 1$  and  $\sigma = 1$ . (b) Resulting  $3 \times 3$  kernel [this is the same as Fig. 3.37(b)].



$$\frac{1}{4.8976} \times$$

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

## Tema 3: Filtrado en el dominio espacial

- FILTRO GAUSSIANO (Gaussian filter):**

- Ejemplo de filtros 3 x 3:**

a b

**FIGURE 3.37**

Examples of smoothing kernels:  
(a) is a *box* kernel;  
(b) is a *Gaussian* kernel.

$$\frac{1}{9} \times$$

1	1	1
1	1	1
1	1	1

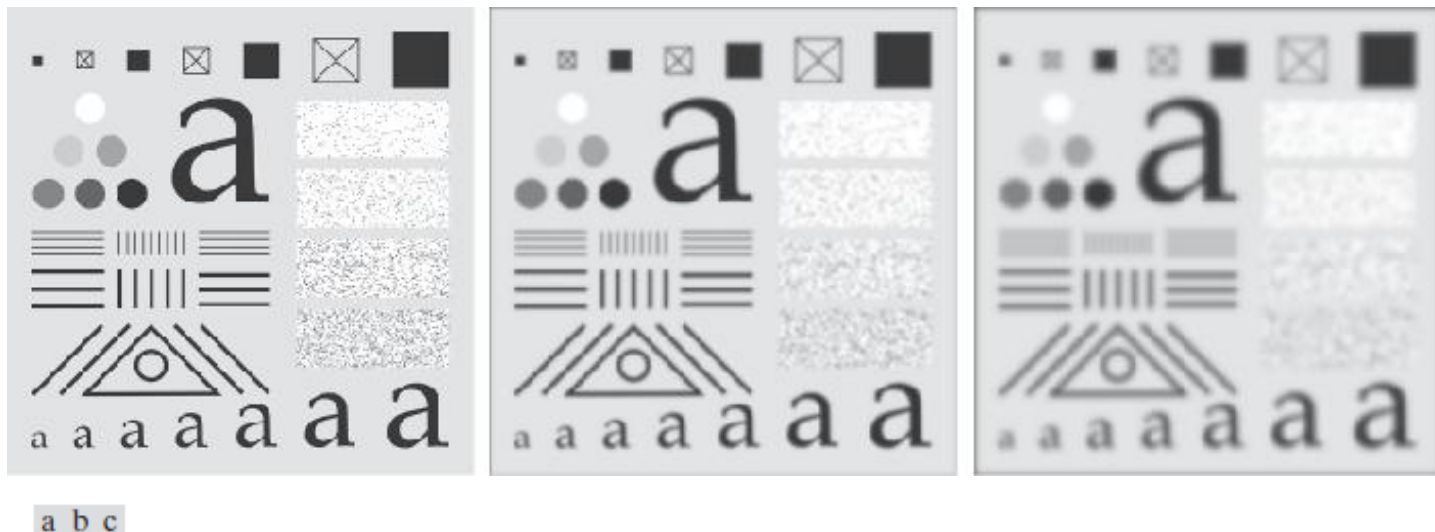
$$\frac{1}{4.8976} \times$$

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

## Tema 3: Filtrado en el dominio espacial

- FILTRO GAUSSIANO:**

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



**FIGURE 3.42** (a) A test pattern of size  $1024 \times 1024$ . (b) Result of lowpass filtering the pattern with a Gaussian kernel of size  $21 \times 21$ , with standard deviations  $\sigma = 3.5$ . (c) Result of using a kernel of size  $43 \times 43$ , with  $\sigma = 7$ . This result is comparable to Fig. 3.39(d). We used  $K = 1$  in all cases.

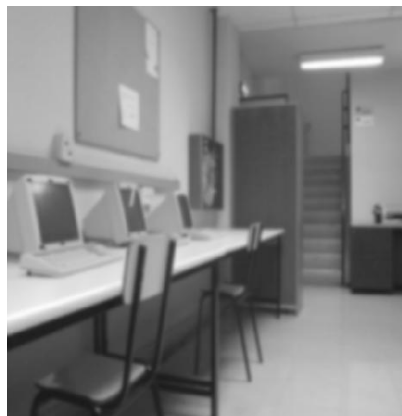
## Tema 3: Filtrado en el dominio espacial

- FILTRO GAUSSIANO:**

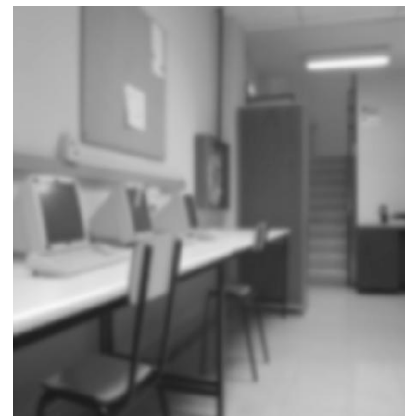
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



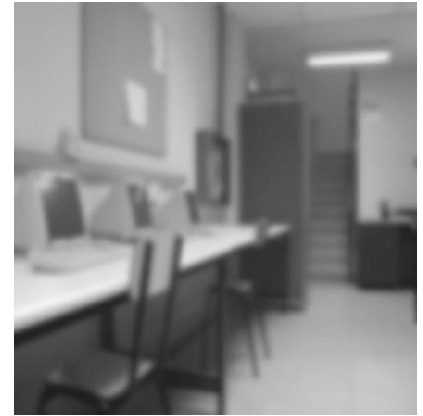
$\sigma = 1$



$\sigma = 2$



$\sigma = 3$



$\sigma = 4$

## Practica:

- Usa la función de OpenCV GaussianBlur()

## Tema 3: Filtrado en el dominio espacial

- **FILTROS ESTADÍSTICOS ORDENADOS:**
  - **Operación no lineal:** Ordenar los valores en la vecindad de cada píxel de menor a mayor y obtener algún valor a partir de la lista.
  - **Tipos:**
    - Máximo
    - Mínimo
    - Mediana

## Tema 3: Filtrado en el dominio espacial

- FILTROS ESTADÍSTICOS ORDENADOS:**

- **Operación no lineal:** Ordenar los valores en la vecindad de cada píxel de menor a mayor y seleccionar el valor en la posición primera (mínimo), intermedia (mediana) o final (máximo)

- Ejemplo:

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Valores ordenados de menor a mayor:

{115, 119, 120, 123, 124, 125, 126, 127, 150}

Mínimo

Mediana

Máximo



## Tema 3: Filtrado en el dominio espacial

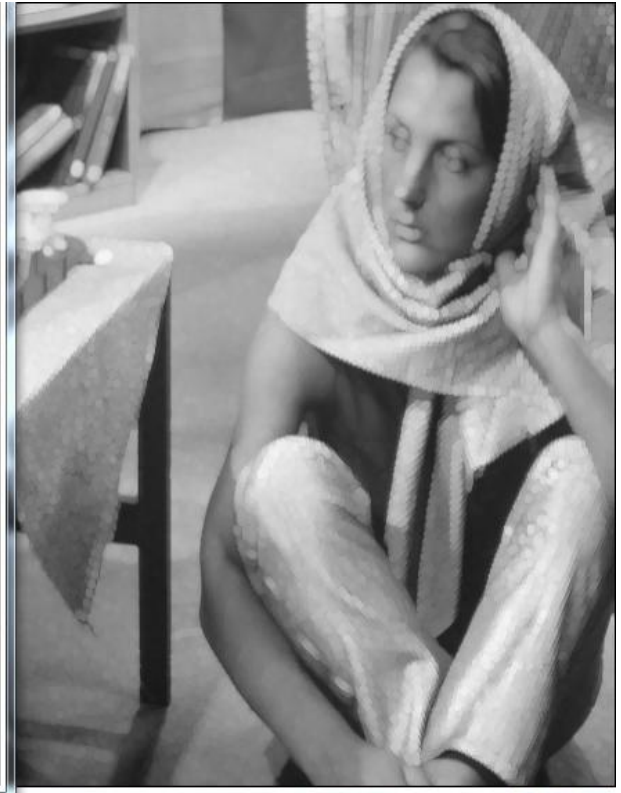
- **FILTROS ESTADÍSTICOS ORDENADOS: Mínimo**
  - Selecciona el menor valor de dentro de una ventana ordenada de valores de píxeles.
  - **Ventaja:** Elimina el ruido sal (píxeles blancos).
  - **Inconvenientes:**
    - Sólo funciona cuando el ruido es exclusivamente tipo sal.
    - Tiende a oscurecer la imagen.

## Tema 3: Filtrado en el dominio espacial

- **FILTROS ESTADÍSTICOS ORDENADOS: Máximo**
  - Selecciona el mayor valor dentro de una ventana ordenada de valores de nivel de gris.
  - **Ventaja:** Elimina el ruido pimienta (píxeles negros).
  - **Inconvenientes:**
    - Sólo funciona cuando el ruido es exclusivamente tipo pimienta.
    - Tiende a aclarar la imagen.

## Tema 3: Filtrado en el dominio espacial

- **FILTROS ESTADÍSTICOS ORDENADOS: Mínimo y máximo**



## Tema 3: Filtrado en el dominio espacial

- **FILTRO DE LA MEDIANA (median filter):**

- **Operación no lineal:** Ordenar los valores en la vecindad de cada píxel de menor a mayor y seleccionar el valor en la posición intermedia (mediana).

- Ejemplo:

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Valores ordenados de menor a mayor:  
{115, 119, 120, 123, 124, 125, 126, 127, 150}

Mediana = 124.

## Tema 3: Filtrado en el dominio espacial

- **FILTROS DE LA MEDIANA:**

- **Ventajas:**

- Atenúa el ruido impulsivo (sal y pimienta).
    - Preserva mejor los bordes de la imagen.

- **Inconvenientes:**

- No es lineal:  $\text{mediana}(f+g) \neq \text{mediana}(f) + \text{mediana}(g)$
    - Pierde detalles (puntos, líneas finas).
    - Redondea las esquinas de los objetos.

## Tema 3: Filtrado en el dominio espacial

- FILTROS DE LA MEDIANA:**



Imagen original



Imagen con ruido  
sal y pimienta.



Imagen tras aplicar  
filtro de la mediana  
3 x 3.

## Tema 3: Filtrado en el dominio espacial

- FILTROS DE LA MEDIANA:**



Imagen original



Imagen con ruido  
sal y pimienta.



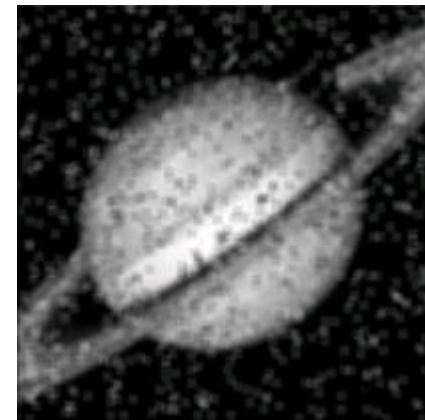
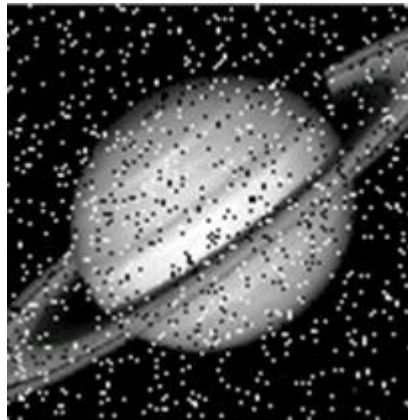
Imagen tras aplicar  
filtro de la mediana  
5 x 5.

## Practica:

- Usa la función de OpenCV `medianBlur()`

## Tema 3: Filtrado en el dominio espacial

- FILTROS DE LA MEDIA Y LA MEDIANA:**



Filtro de la media 3 x 3.

Filtro de la mediana 3 x 3.





## Tema 3: Filtrado en el dominio espacial

- **FILTROS DE LA MEDIANA PONDERADA:**

- **Operación no lineal:** La nueva imagen se genera a base de hallar la mediana de los valores del entorno del píxel, repetidos tantas veces como indique una determinada máscara.

- **Máscara 3 x 3 (ej):**

1	2	1
2	4	2
1	2	1

## Tema 3: Filtrado en el dominio espacial

- FILTROS DE LA MEDIANA PONDERADA:**

1	2	1
2	4	2
1	2	1

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Valores ordenados de menor a mayor:

{115, 119, 120, 123, 124, 125, 126, 127, 150}

Aplicando máscara:

{115, 119, 119, 120, 120, 123, 124, 125, 125, 126, 126, 127, 150, 150, 150, 150}

# Filtros de realce



## Tema 3: Filtrado en el dominio espacial

- **FILTROS DE REALCE:**

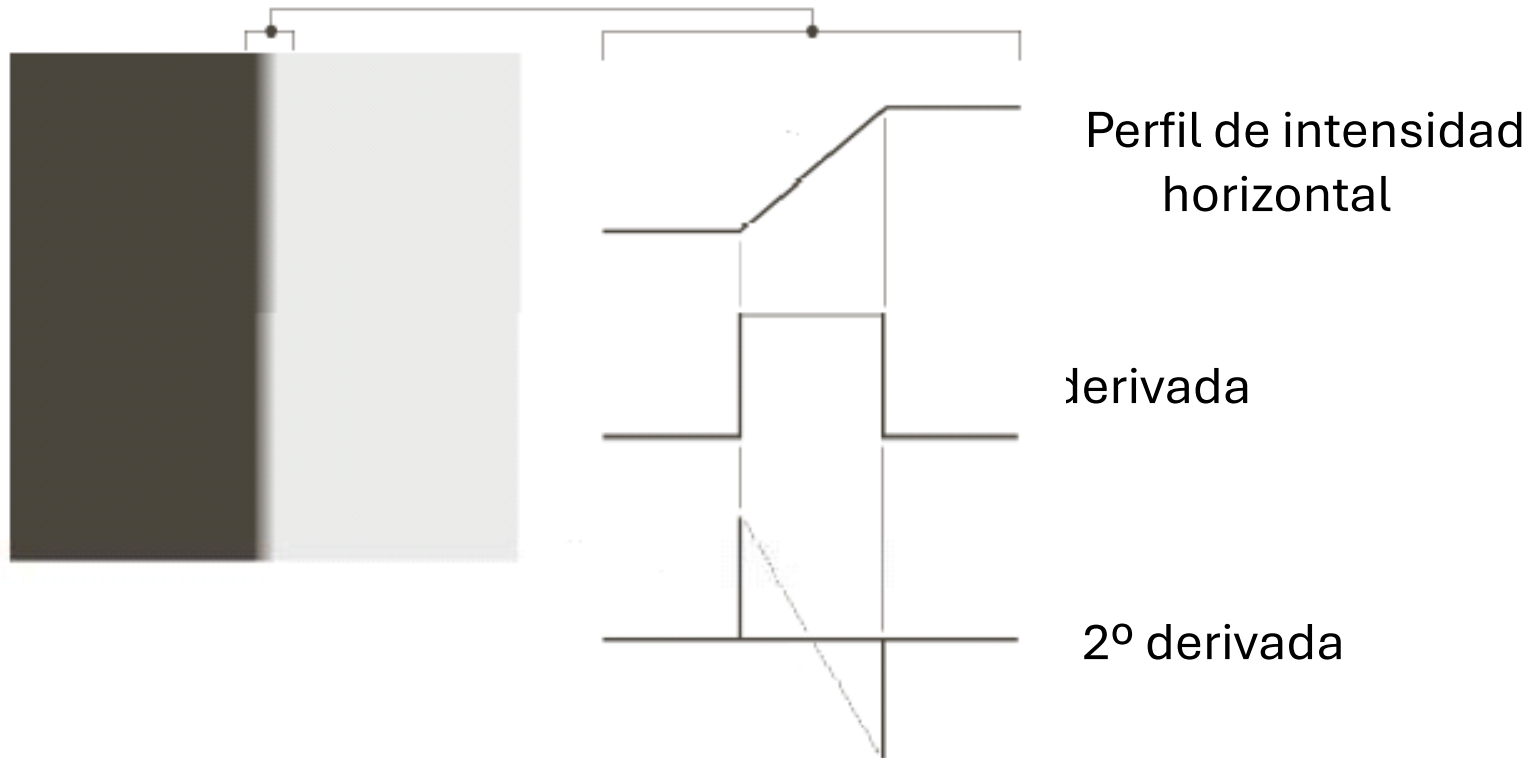
- **Objetivo:** Realzar los detalles de una imagen que hayan podido quedar emborronados. Estos filtros están asociados, por tanto, con la detección de lados o bordes (edges).
- La **idea** que subyace en la mayor parte de las técnicas de detección de bordes es el cálculo de un operador local de derivación ya que un píxel pertenece a un borde si se produce un cambio brusco entre niveles de grises con sus vecinos. Mientras más brusco sea el cambio, más fácil es detectar el borde.
- **Inconveniente:** El ruido es colateralmente realzado.

## Tema 3: Filtrado en el dominio espacial

- **Derivada digital:**
  - La derivada de una función digital se mide en términos de variaciones entre píxeles adyacentes. Existen varias formas de definir estas diferencias pero todas deben cumplir:
    - La *1º derivada* debe ser 0 en zonas de intensidad constante y distinta de 0 en zonas de variaciones (escalones o rampas).
    - La *2º derivada* debe ser 0 en zonas de intensidad constante y a lo largo de rampas con intensidad constante y debe ser distinta de 0 en escalones y comienzo y fin de rampa.

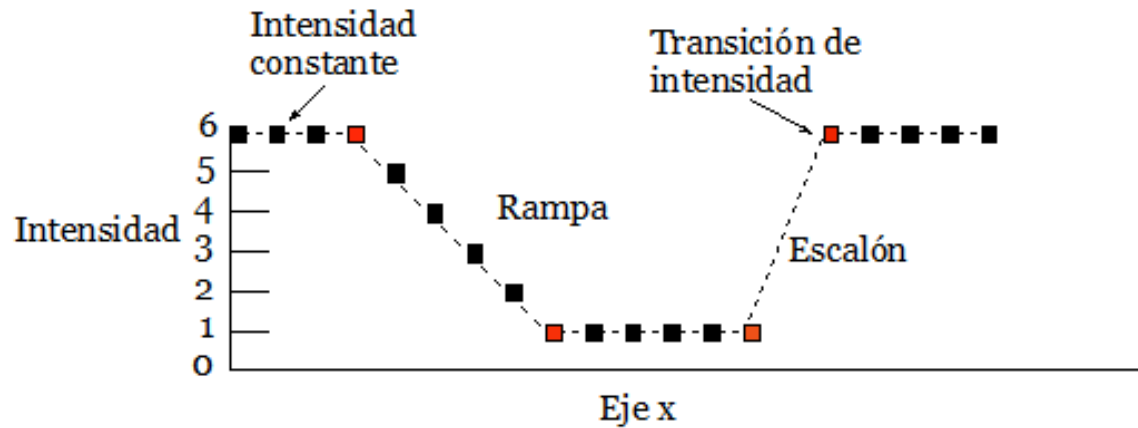
## Tema 3: Filtrado en el dominio espacial

- Derivada digital:



## Tema 3: Filtrado en el dominio espacial

- Derivada digital:**



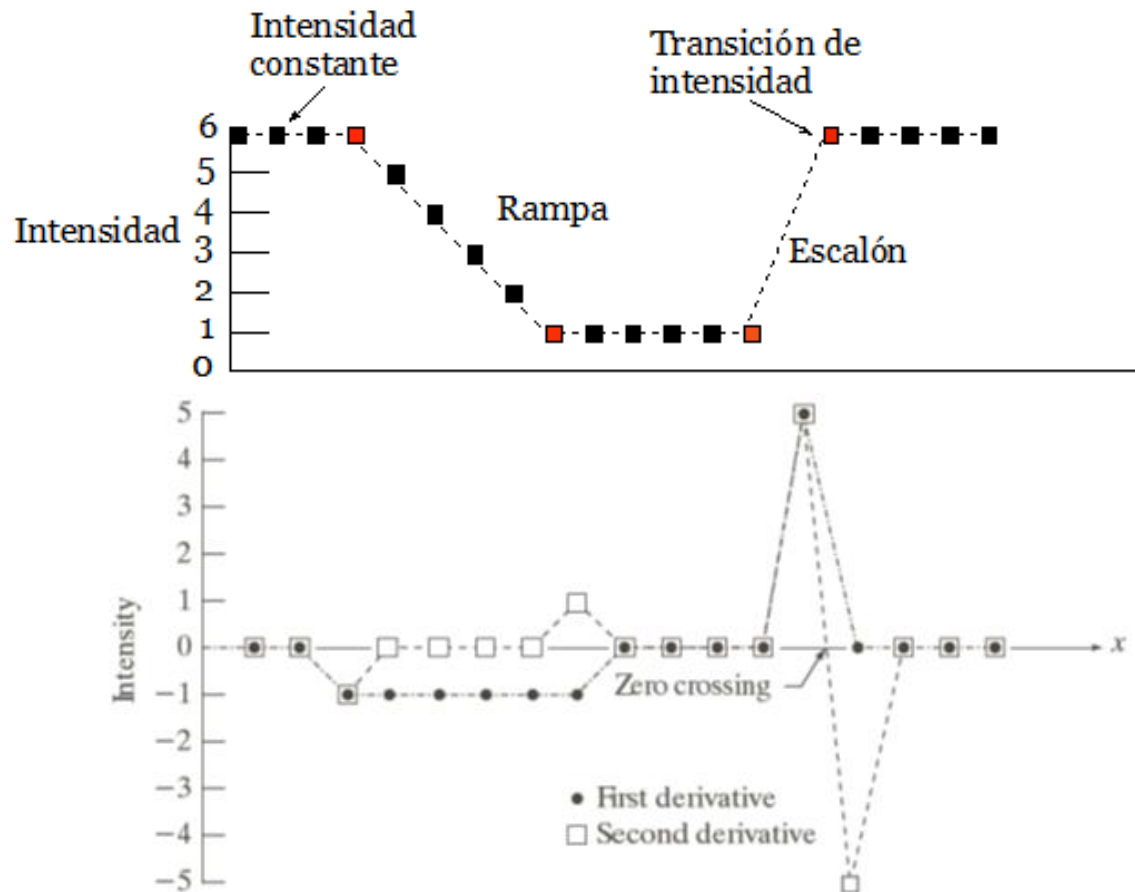
6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0	5	0	0	0
---	---	---	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

0	0	-1	0	0	0	0	1	0	0	0	0	5	-5	0	0	0
---	---	----	---	---	---	---	---	---	---	---	---	---	----	---	---	---

## Tema 3: Filtrado en el dominio espacial

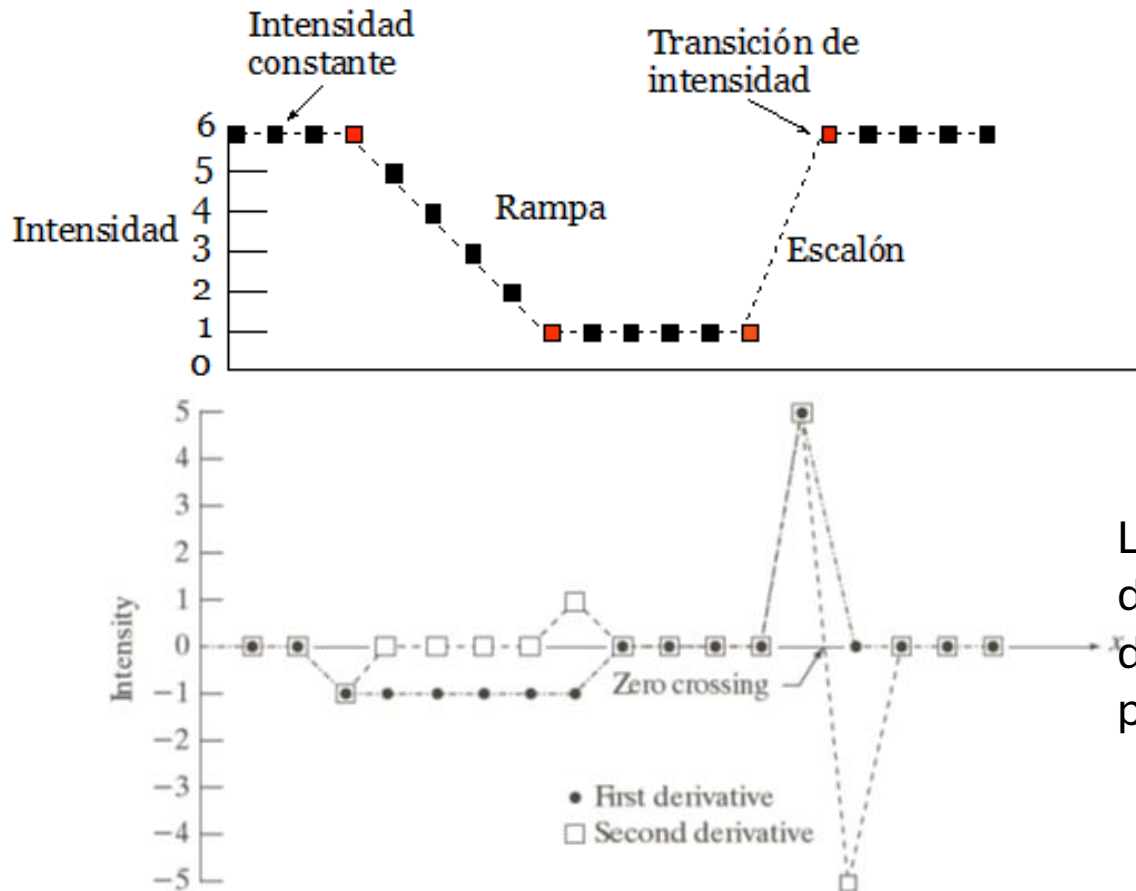
- Derivada digital:**





## Tema 3: Filtrado en el dominio espacial

- Derivada digital:**



La segunda derivada detecta mejor los detalles finos que la primera derivada

## Tema 3: Filtrado en el dominio espacial

- **GRADIENTE DIGITAL:**

- Una aproximación del gradiente de una imagen  $f(x,y)$  sería:

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \quad \begin{aligned} g_x &= \frac{\partial f}{\partial x} = f(x, y) - f(x - 1, y) \\ g_y &= \frac{\partial f}{\partial y} = f(x, y) - f(x, y - 1) \end{aligned}$$

- Estas operaciones pueden usarse en forma de convolución usando las siguientes máscaras, respectivamente:

0	0	0
-1	1	0
0	0	0

0	-1	0
0	1	0
0	0	0

## Tema 3: Filtrado en el dominio espacial

- **GRADIENTE DIGITAL: Operador de Sobel.**

- Se usan dos máscaras 3 x 3 para aproximar el gradiente:

1	2	1
0	0	0
-1	-2	-1

$g_x$

-1	0	1
-2	0	2
-1	0	1

$g_y$

- Observemos que tienen más pesos los píxeles situados en la vertical y la horizontal (con respecto al píxel central) que los píxeles de la diagonal.

## Tema 3: Filtrado en el dominio espacial

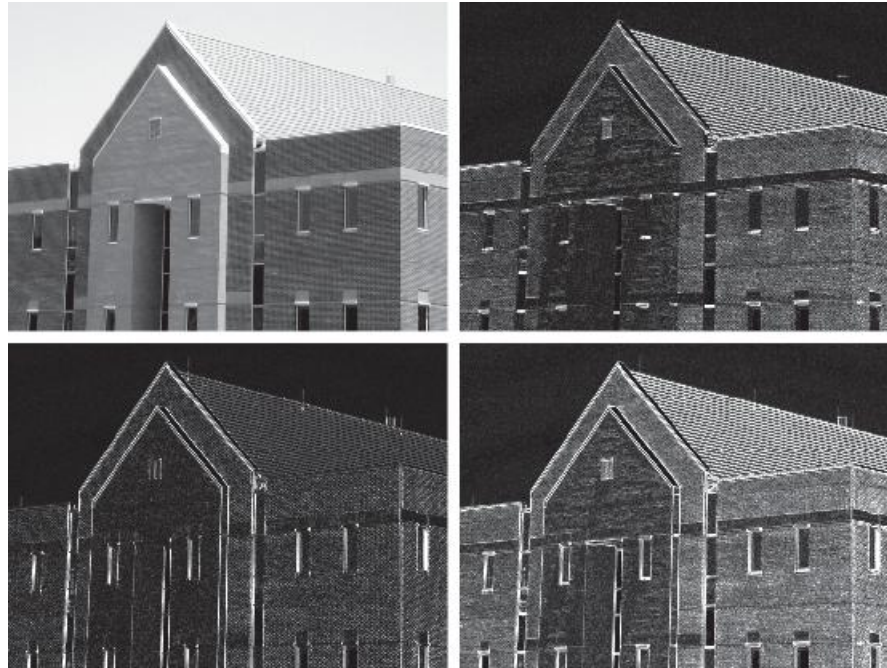
- GRADIENTE DIGITAL: Operador de Sobel.**

Imagen original  $g(x,y)$

a	b
c	d

**FIGURE 10.16**

(a) Image of size  $834 \times 1114$  pixels, with intensity values scaled to the range  $[0,1]$ .  
 (b)  $|g_x|$ , the component of the gradient in the  $x$ -direction, obtained using the Sobel kernel in Fig. 10.14(f) to filter the image.  
 (c)  $|g_y|$ , obtained using the kernel in Fig. 10.14(g).  
 (d) The gradient image,  $|g_x| + |g_y|$ .



$|g_x|$

$$|\nabla f| \approx |g_x| + |g_y|$$

$|g_y|$

## Tema 3: Filtrado en el dominio espacial

- **GRADIENTE DIGITAL: Operador de Sobel.**

- Se usan dos máscaras 3 x 3 para aproximar el gradiente:

1	2	1
0	0	0
-1	-2	-1

$g_x$

-1	0	1
-2	0	2
-1	0	1

$g_y$

## Practica:

- Usa la función de OpenCV Sobel(). Estudia sus argumentos. ¿Cuáles se usan para cambios en horizontal? ¿Y en vertical?

## Tema 3: Filtrado en el dominio espacial

- **LAPLACIANO DIGITAL:**

- El Laplaciano de una función bidimensional  $f$  es un operador definido por:

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

- El Laplaciano puede implementarse en forma digital de varias maneras. La más frecuente en la práctica es:

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

0	1	0
1	-4	1
0	1	0

## Tema 3: Filtrado en el dominio espacial

- **LAPLACIANO DIGITAL:**

- Otras máscaras 3 x 3 para implementar en Laplaciano digital:

1	1	1
1	-8	1
1	1	1

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

- Nótese que la suma de los coeficientes de la máscara debe ser cero, lo que es coherente en el caso de que el punto en cuestión y sus vecinos tengan el mismo valor.

- Los píxeles a ambos lados de un borde darán como respuesta un número positivo y negativo “grande” en valor absoluto.

## Tema 3: Filtrado en el dominio espacial

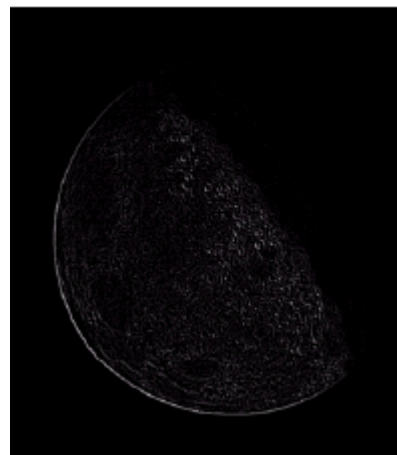
- **LAPLACIANO DIGITAL:**

- El Laplaciano no suele usarse directamente en la práctica por ser muy sensible al ruido. Se suele usar sumado o restado con la imagen original para realzar detalles.

$$g(x,y)=f(x,y)+c \nabla^2 f (x,y)$$



Imagen original



Laplaciano  
(con la primera máscara )



Imagen original  
-Laplaciano

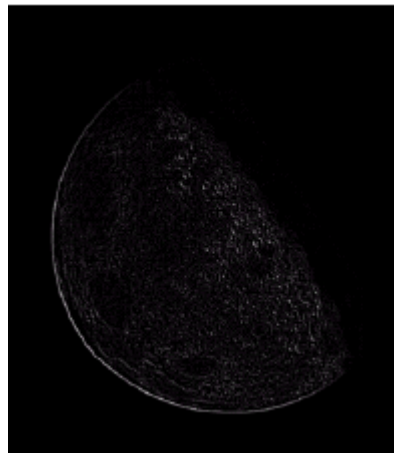


## Tema 3: Filtrado en el dominio espacial

- LAPLACIANO DIGITAL:**



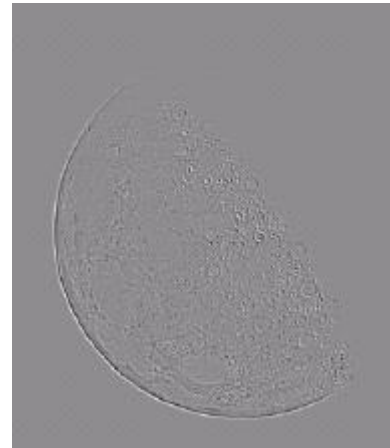
Imagen original



Laplaciano



Secciones grandes negras ya que los valores negativos del Laplaciano se truncan en 0 (negro).



Reescalado  
del laplaciano



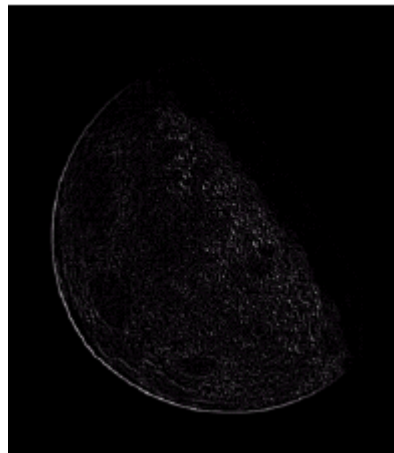
Para que los valores estén entre  $[0, L-1]$  (transformación de intensidad)

## Tema 3: Filtrado en el dominio espacial

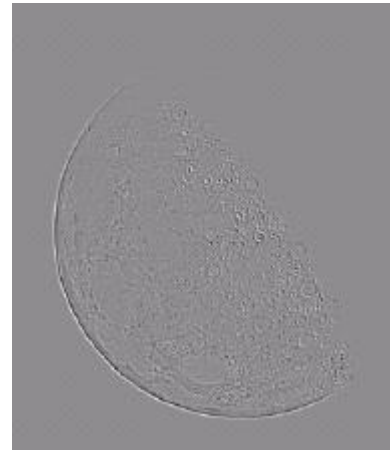
- **LAPLACIANO:**



Imagen original



Laplaciano



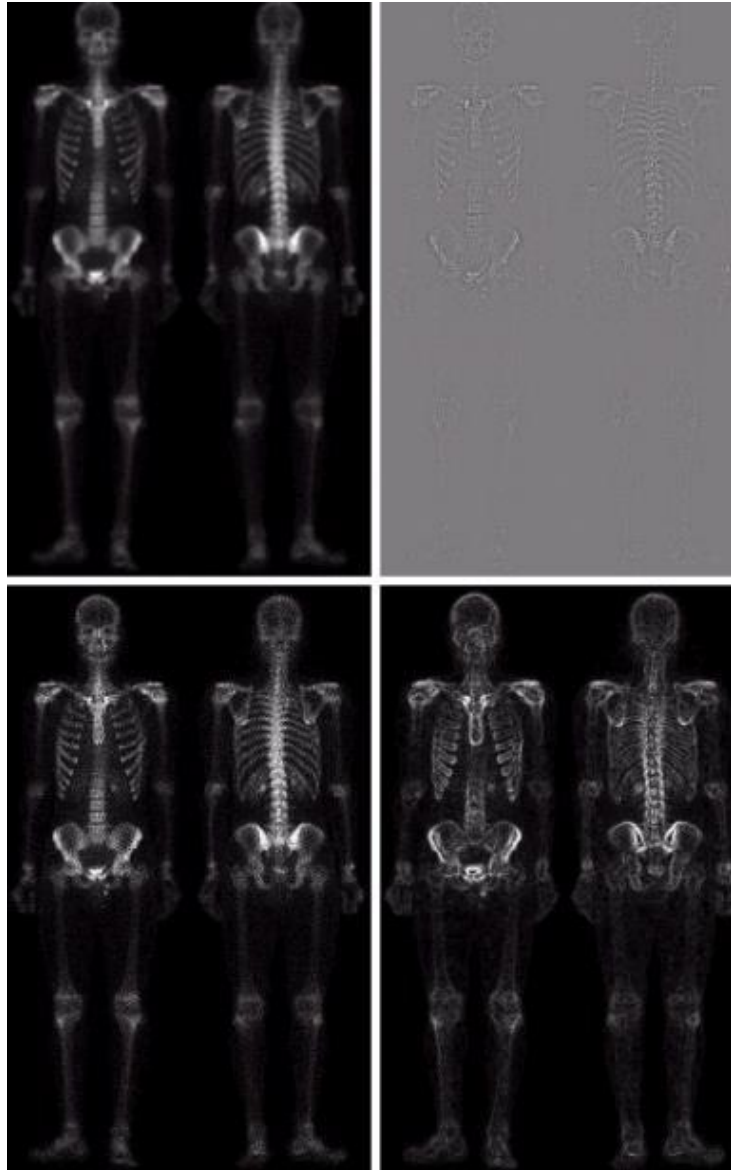
Reescalado  
del laplaciano

## Practica:

- Usa la función de OpenCV `Laplacian()`. Estudia sus argumentos. ¿Qué máscara implementa? ¿Proporciona un resultado reescalado?

## Tema 3: Filtrado en el dominio espacial

Ejemplo:

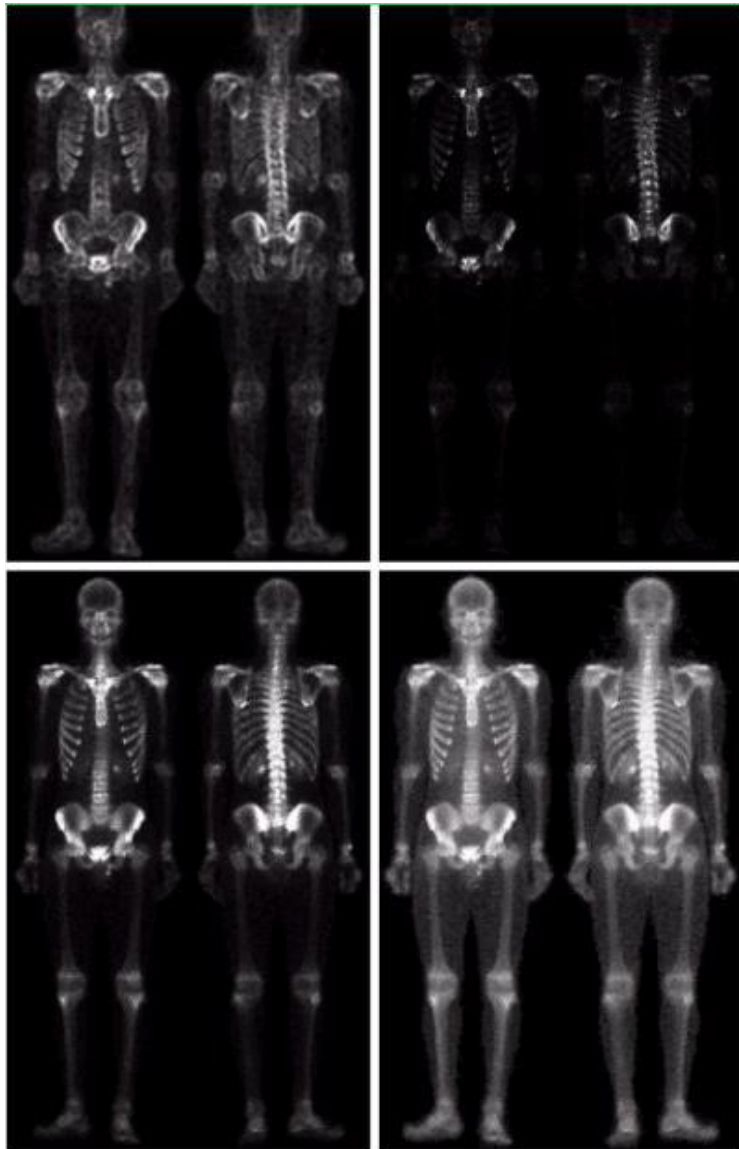


a	b
c	d

**FIGURE 3.46**

(a) Image of whole body bone scan.  
 (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).

## Tema 3: Filtrado en el dominio espacial



e	f
g	h

**FIGURE 3.46**

*(Continued)*

(e) Sobel image smoothed with a  $5 \times 5$  averaging filter. (f) Mask image formed by the product of (c) and (e).

(g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

## Tema 3: Filtrado en el dominio espacial

- **Bibliografía**
- R.C. González, R.E. Woods. *Digital Image Proccesing*, 4th edition. Pearson, 2018.  
**(La mayor parte de las imágenes han sido extraídas de esta publicación)**
- Mark S. Nixon, Alberto S. Aguado. *Feature Extraction and Image Processing for Computer Vision*. Elsevier, 2020.