

# Procesamiento Digital de Imágenes

Capítulo X— Procesamiento basado en el vecindario

[rms@decsai.ugr.es](mailto:rms@decsai.ugr.es)

<http://decsai.ugr.es/~rms>

# Contenidos

- Procesamiento basado en el vecindario
- Convolución y Correlación
- Alisamiento de imágenes (filtro paso bajo)
- Realce de imágenes (filtro paso alto)
- Procesamiento de regiones de interés
- Combinación de métodos de mejora de imágenes
- Bibliografía

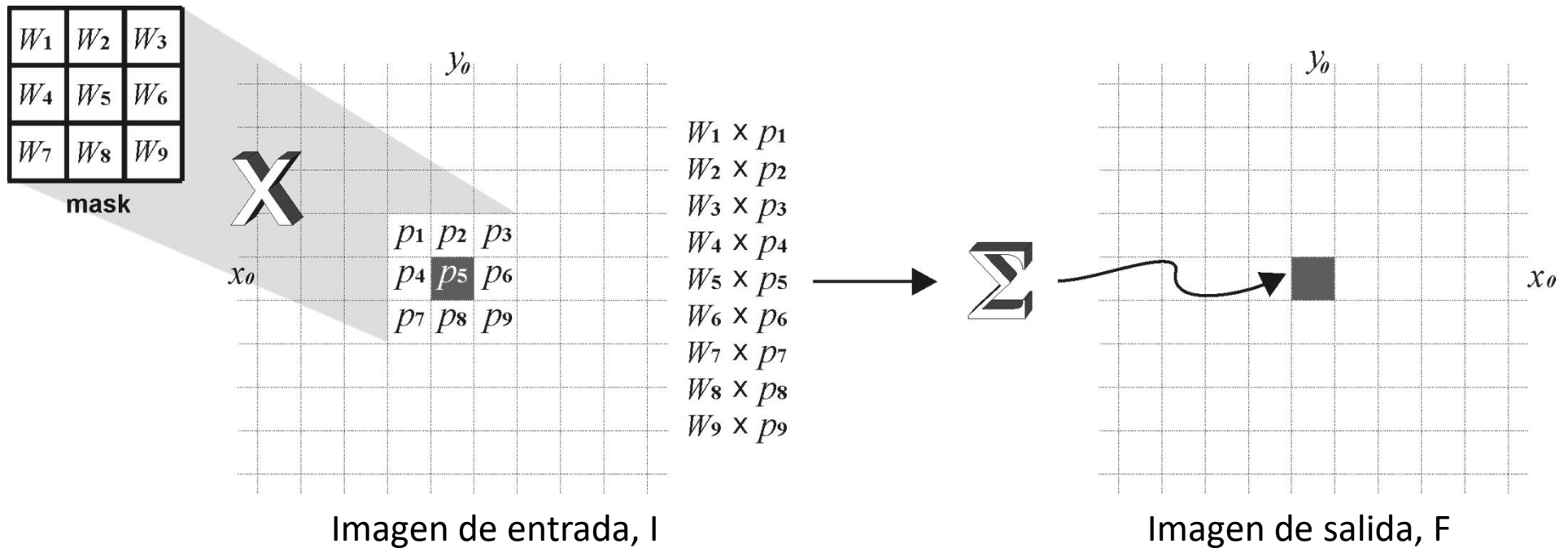
# ¿Qué vamos a aprender en este tema?

- ¿Qué es el procesamiento basado en vecinos y en qué se diferencia del puntual?
- ¿Qué es la convolución y cómo se utiliza para procesar imágenes?
- ¿Qué es un filtro lineal paso-bajo, cómo se utiliza y cómo se implementa usando convolución 2D?
- ¿Qué es un filtro de mediana y cómo se usa?
- ¿Qué es un filtro lineal paso-alto, cómo se utiliza y cómo se implementa usando convolución 2D?

# Procesamiento basado en vecinos

- Pasos Fundamentales
  - Definir un punto de referencia en la imagen de entrada  $f(x_0, y_0)$
  - Realizar una operación que involucre solo a píxeles vecinos del píxel de entrada
  - Almacenar en  $g(x_0, y_0)$  el resultado
  - Repetir el proceso en todos los píxeles de la imagen

# Procesamiento basado en vecinos (filtro lineal)



# Procesamiento por vecinos

- **Filtros lineales:** donde la salida resultante se computa como una suma de productos de los valores de los píxeles y los coeficientes de una máscara del mismo tamaño que el vecindario.
  - Ejemplo: filtro de media
- **Filtros no lineales:** la salida del filtro selecciona un valor de los niveles del gris ordenados de él y los vecinos
  - Ejemplo: filtro de mediana

# Convolución y Correlación


- La **convolución** y la **correlación** son dos operaciones matemáticas que se utilizan para procesamiento lineal de imágenes basado en vecinos.
  - Las dos operaciones son (ligeramente) diferentes.
- La **convolución** y la **correlación** procesan una imagen calculando, en cada píxel, una suma ponderada de los valores de ese píxel y sus vecinos.
  - Dependiendo de los pesos se realizan diferentes tipos de procesamiento sobre la imagen.

# Convolución y Correlación

- Convolución Bidimensional

$$g(x, y) = \sum_{j=-n_2}^{n_2} \sum_{k=-m_2}^{m_2} h(j, k) f(x - j, y - k)$$

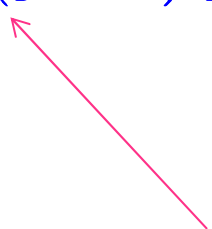
Filtro de convolución



- Correlación Bidimensional

$$g(x, y) = \sum_{j=-n_2}^{n_2} \sum_{k=-m_2}^{m_2} h(j, k) f(x + j, y + k)$$

Filtro de correlación





# Convolución y Correlación

## Ejemplo de convolución

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

\*

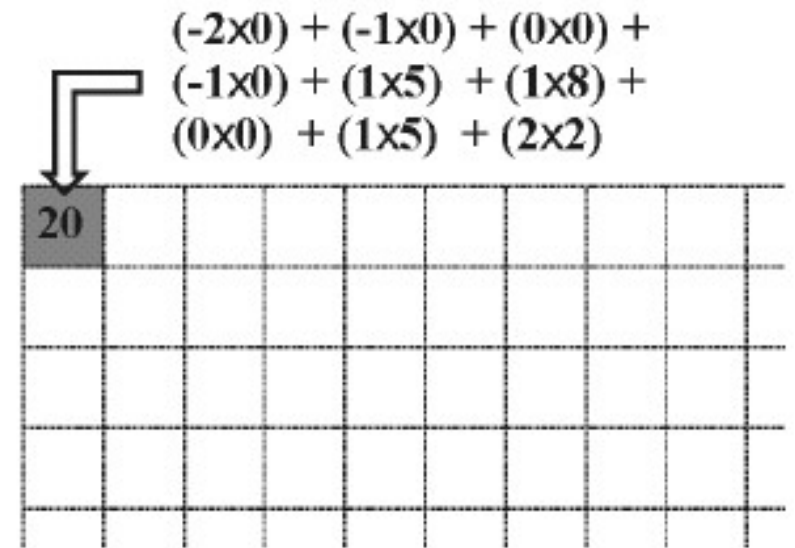
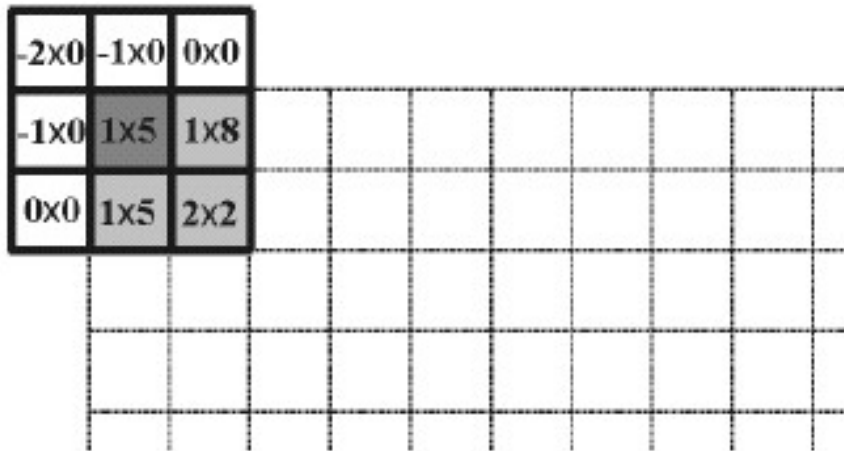
5	8	3	4	6	2	3	7
3	2	1	1	9	5	1	0
0	9	5	3	0	4	8	3
4	2	7	2	1	9	0	6
9	7	9	8	0	4	2	4
5	2	1	8	4	1	0	9
1	8	5	4	9	2	3	8
3	7	1	2	3	4	4	6

=

20	10	2	26	23	6	9	4
18	1	-8	2	7	3	3	-11
14	22	5	-1	9	-2	8	-1
29	21	9	-9	10	12	-9	-9
21	1	16	-1	-3	-4	2	5
15	-9	-3	7	-6	1	17	9
21	9	1	6	-2	-1	23	2
9	-5	-25	-10	-12	-15	-1	-12

# Convolución y Correlación

- Convolución 2D



# Convolución y Correlación

- Convolución con diferentes máscaras
  - La convolución es un herramienta muy versátil
  - Dependiendo de la máscara utilizada los resultados son muy diferentes

Filtro paso bajo	Filtro paso alto	Detector fronteras horizontales
$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

# Convolución y Correlación

- Convolución con diferentes máscaras

- (a) original, (b)-(d) resultado tras aplicar las máscaras en la tabla anterior



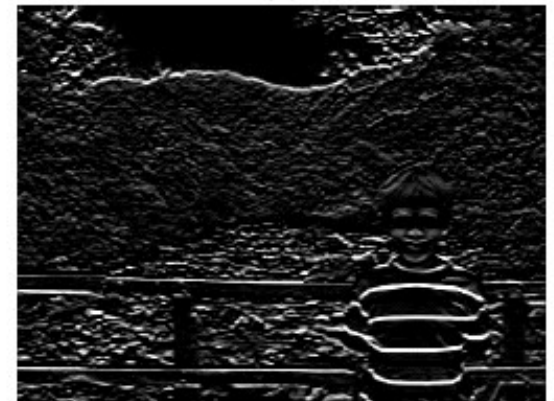
(a)



(b)



(c)



(d)

# Convolución y Correlación

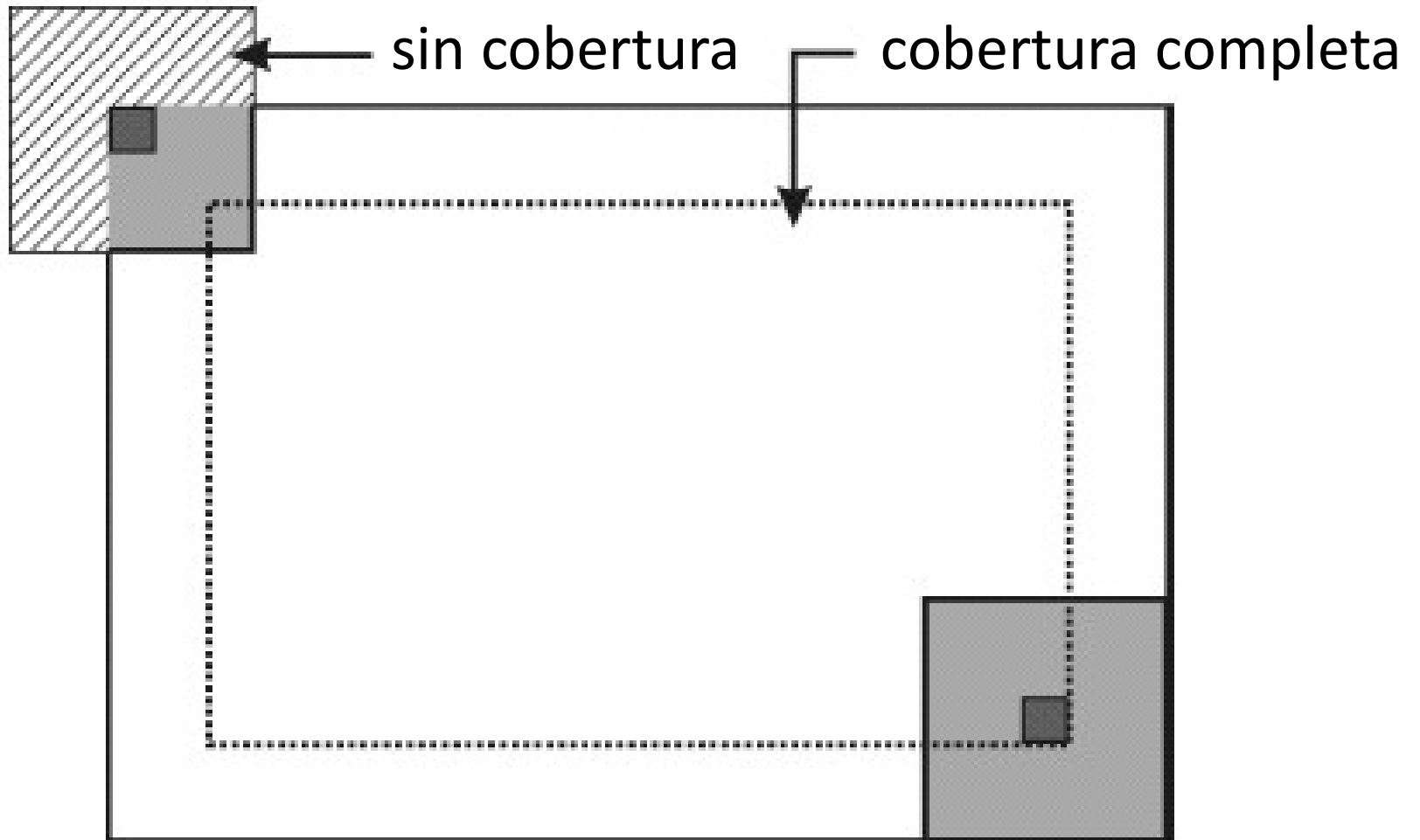
- Correlación
  - La correlación es lo mismo que la convolución pero sin girar la máscara antes de realizar las sumas y productos.
  - La diferencia entre la convolución y la correlación 2D es muchas veces irrelevante ya que muchas de las máscaras que se utilizan son simétricas alrededor del origen.

# Convolución en Matlab

- **conv2**: calcula la convolución bidimensional entre dos matrices. Además de las matrices, esta función tiene como parámetro adicional el tamaño de la salida.
- **filter2**: Implementa la correlación. También tiene parámetros adicionales.
- **imfilter**: implementa convolución y correlación bidimensionales.

Ten cuidado con el orden de las matrices cuando utilices estas funciones. Fíjate en el tipo de dato que aceptan.

# Tratamiento de las fronteras



# Tratamiento de fronteras: opciones

- Ignorar las fronteras
  - No procesar los píxeles donde la máscara no puede ser incluida completamente en la imagen
  - Sustituir estos píxeles con un valor dado
- Extender la imagen con ceros
- Extender la imagen con un valor dado
- Realizar un espejo de las imágenes
- Realizar una extensión periódica de la imagen
- Examina en Matlab **boundary\_options** en **imfilter**



## Alisamiento de Imágenes (Filtro Paso Bajo, LPF)

- Son filtros cuyo efecto es atenuar las componentes de alta frecuencia (detalles finos) en la imagen y mantener las componentes de baja frecuencia (regiones homogéneas y detalles más gruesos).
- LPFs se implementan usando convoluciones con coeficientes no negativos que suman uno.
- LPFs se utilizan para emborronar una imagen o suprimir ruido en ella.
- En Matlab **`imfilter`** y **`fspecial`**.

## Filtro de Media (Promedio)

- Es el filtro de alisamiento espacial más común.
- Los coeficientes de la máscara son 1 divididos por el número de vecinos.
- También se llama filtro de caja.

$$h(x, y) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

# Filtro de Media (Promedio)



(a)



(b)



(c)



(d)

De izquierda a derecha y de arriba abajo: original y convolución con filtros de medias de tamaño 7x7, 15x15 y 31x31

# Filtro de Media (Variaciones)

- Coeficientes de la máscara modificados

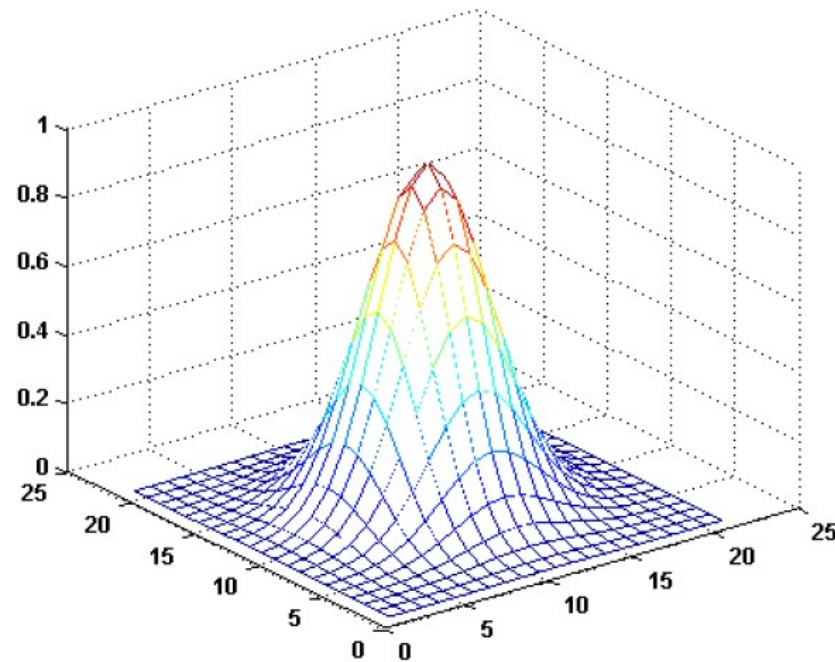
$$h(x, y) = \begin{bmatrix} 0.075 & 0.125 & 0.075 \\ 0.125 & 0.200 & 0.125 \\ 0.075 & 0.125 & 0.075 \end{bmatrix}$$

- Promedios direccionales
- Aplicación selectiva del cálculo de promedios
- Eliminación de outliers antes de hacer el promedio

# Filtro de emborronamiento gaussiano

- Es el LPF más conocido con un núcleo no uniforme.
- Los coeficientes de la máscara son puntos de la función gaussiana bidimensional (normalizados para que sumen uno).

$$h(x, y) \propto \exp \left[ \frac{-(x^2 + y^2)}{2\sigma^2} \right]$$



# Filtro de emborronamiento gaussiano

- Propiedades
  - Es simétrico en rotación y por tanto no produce sesgo en ninguna dirección.
  - Es separable y por tanto puede ser implementable muy eficientemente.
  - Los coeficientes disminuyen conforme nos alejamos del centro.
  - Su transformada de Fourier es otro filtro gaussiano.
  - La convolución de dos gaussianas es otra gaussiana.

# Filtro de emborronamiento gaussiano

- Propiedades

```
I = imread('cameraman.tif');  
  
h1 = fspecial('gaussian', [5 5], 1);  
h2 = fspecial('gaussian', [13 13], 1);  
h3 = fspecial('average', [13 13]);  
  
J1 = imfilter(I, h1);  
J2 = imfilter(I, h2);  
J3 = imfilter(I, h3);  
  
subplot(2,2,1);imshow(I);  
title('imagen original');  
  
subplot(2,2,2); imshow(J1);  
title('Filtro gaussiano 5x5, desv=1');  
  
subplot(2,2,3);imshow(J2);  
title('Filtro gaussiano 13x13, desv=1');  
  
subplot(2,2,4); imshow(J3);  
title('Filtro media 13x13');
```

Observa el efecto de las fronteras

imagen original



Filtro gaussiano 5x5, desv=1



Filtro gaussiano 13x13, desv=1



Filtro media 13x13



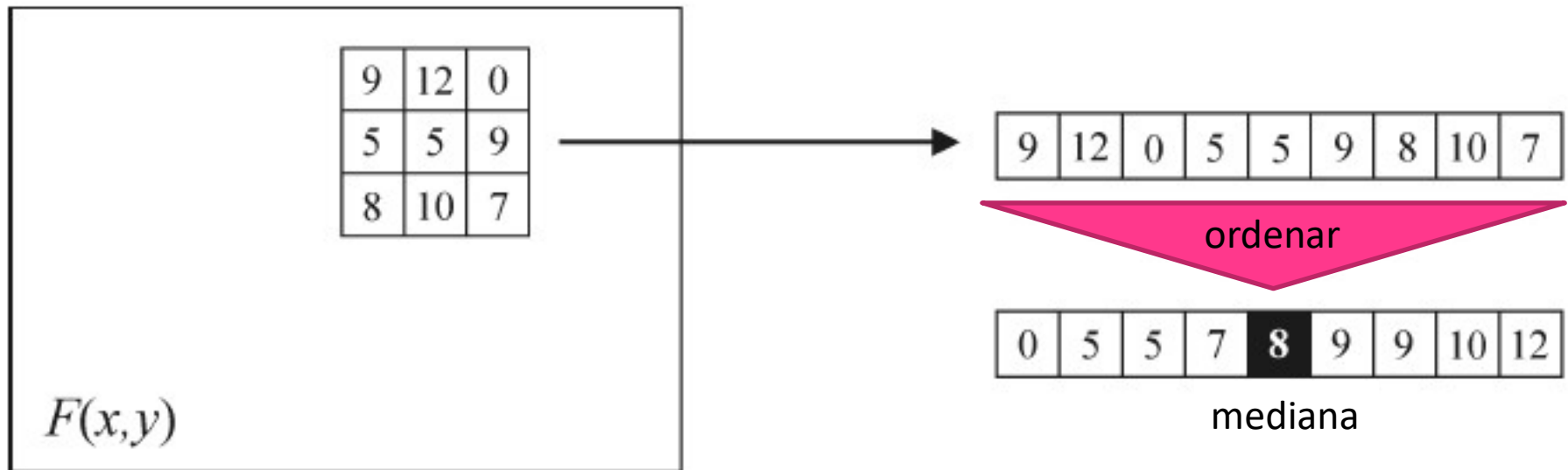
# La Mediana y otros filtros no lineales

- Los filtros no lineales también funcionan a nivel de vecindario pero no procesan los píxeles utilizando operadores de convolución/correlación
  - En su lugar, ordenan los valores de los píxeles en los vecinos y seleccionan un valor de la lista ordenada
  - A veces reciben el nombre de filtros de orden
  - Ejemplos: la mediana, el mínimo, el máximo (los veremos más adelante)



# Filtro de mediana

- Funciona ordenando el vecindario de un tamaño dado. Encuentra la mediana y reemplaza el píxel original con la mediana del vecindario



# Filtro de mediana

- Ejemplo  
(reducción ruido  
sal y pimienta)

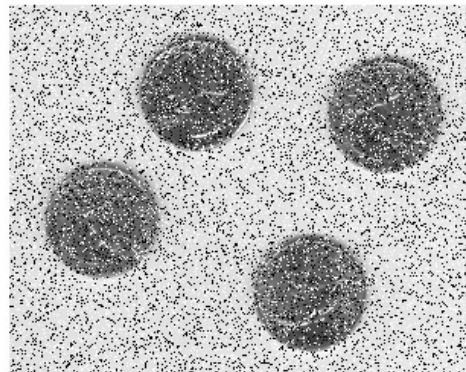
```
I = imread('eight.tif');  
J = imnoise(I,'salt &  
pepper',0.02);  
K = medfilt2(J);  
  
figure, imshow(J), %en (a)  
figure, imshow(K) %en (b)  
  
J = imnoise(I,'salt &  
pepper',0.2);  
K = medfilt2(J);  
figure, imshow(J), %en (c)  
figure, imshow(K) %en (d)
```



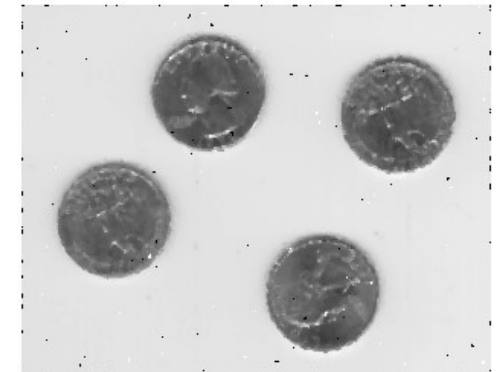
(a)



(b)



(c)



(d)

# Realce de Imágenes (Filtro paso Alto, HPF)

- Son filtros en el dominio espacial cuyo efecto en la salida es equivalente a preservar o enfatizar las componentes de frecuencia alta (detalles, puntos, líneas, fronteras), y realzar las transiciones en las intensidades en la imagen.
- Los HPFs pueden implementarse usando máscaras de convolución 2D con coeficientes positivos y negativos que aproximan el laplaciano, una derivada de segundo orden isotrópica que responde a los cambios de intensidad en cualquier dirección.

# Realce de Imágenes (HPF)

- El laplaciano
  - El Laplaciano de una imagen  $f(x,y)$  se define como

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

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

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

# Realce de Imágenes (HPF)

- Máscara laplaciana compuesta

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

- Para  $c=-1$

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

# Realce de Imágenes (HPF)

- Ejemplo



```
I = imread('escudo.png'); I=rgb2gray(I);  
h = fspecial('laplacian', 0)  
I1 = im2double(I);  
J = imfilter(I1,h);  
K = I1-J; K=max(min(K,1),0); % ¿Es necesaria la segunda orden?  
h8 = [1 1 1; 1 -8 1; 1 1 1]  
K8 = I1 - imfilter(I1,h8,'replicate'); K8=max(min(K8,1),0);  
% ¿Es necesaria la segunda orden?  
subplot(1,3,1);imshow(I);  
subplot(1,3,2);imshow(K);  
subplot(1,3,3);imshow(K8);
```

# Filtros de diferencias direccionales

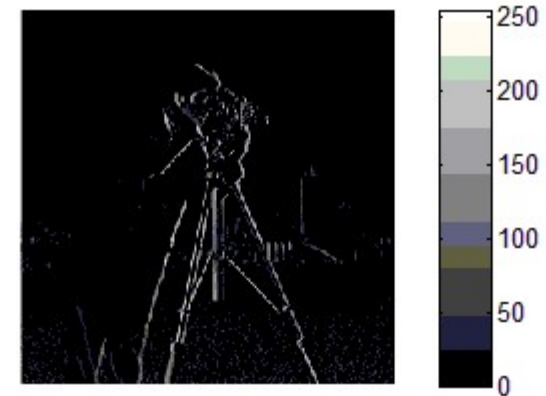
- Es similar al laplaciano de altas frecuencias.
  - La principal diferencia es la selección de una dirección específica a enfatizar
- Ejemplos de máscaras de diferencias direccionales.
- Estos filtros volverán a aparecer en la detección de fronteras.

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

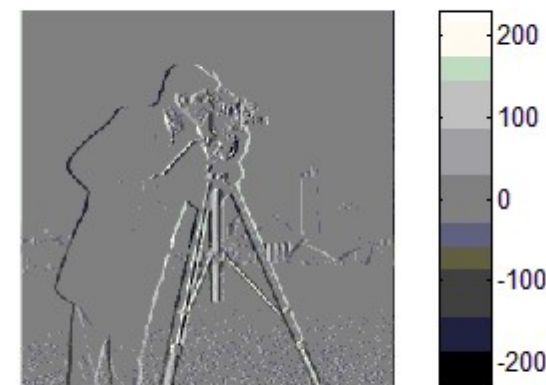
# Filtros de diferencias direccionales

Ten cuidado con el cálculo basado en filtros. Utiliza el tipo de dato apropiado:

```
a=imread('cameraman.tif');  
b=imfilter(a, [-1 0 1], ...  
    'corr', 'same');  
imshow(b); colorbar
```



```
b=imfilter(double(a), [-1 0 1], ...  
    'corr', 'same');  
imshow(b, []); colorbar
```





# Máscaras Unsharp

- La idea es calcular la diferencia entre la imagen de entrada y una versión borrosa de la imagen de entrada.
- Se aumenta la cantidad de alta frecuencia (detalles finos) reduciendo la importancia de las componentes de baja frecuencia.
- Veremos diferentes aproximaciones en los guiones de prácticas.

## Filtrado High-Boost

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

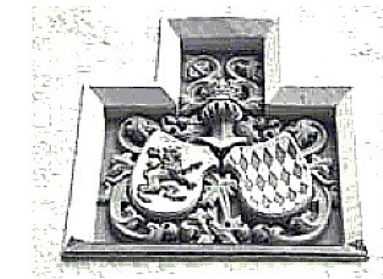
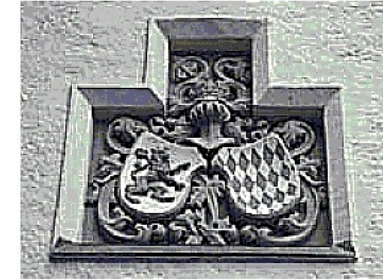
- donde  $c (>8)$  es un factor de amplificación que controla cuánto peso se le da a la imagen original y a la versión filtrada paso alto de la imagen
  - $c=8$ , el negativo de esta máscara es el laplaciano
  - Valores mayores de  $c$  producen más realce

# Filtrado High-Boost

```
a=imread('escudo.png'); a=rgb2gray(a);  
a=double(a);  
b=imfilter(a,[-1 -1 -1;-1 9 -1;-1 -1 -1]);  
imshow(uint8(b))
```

```
a=imread('escudo.png'); a=rgb2gray(a);  
a=double(a);  
b=imfilter(a,[-1 -1 -1;-1 8.5 -1;-1 -1 -1]);  
imshow(uint8(b))
```

```
a=imread('escudo.png'); a=rgb2gray(a);  
a=double(a);  
b=imfilter(a,[-1 -1 -1;-1 9.5 -1;-1 -1 -1]);  
imshow(uint8(b))
```

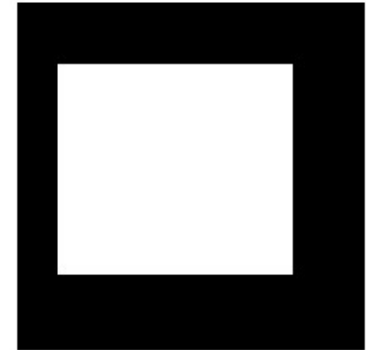


# Procesamiento de Regiones de Interés (ROI)

- Las operaciones de filtrado se realizan a veces en una parte de la imagen (conocida como región de interés (ROI)) que normalmente se especifica mediante una máscara.
  - El enmascarado de una imagen es el proceso de extraer la ROI para su posterior procesamiento
- En Matlab
  - Se realiza mediante la combinación de dos funciones **roipoly** (ya la conocemos) y **roifilt2**

# Procesamiento de ROI

```
I=imread('lena.png');  
I=double(I);  
r= [91 400 400 91];  
c= [60 60 407 407];  
BW= roipoly(I,c,r);
```



```
h=fspecial('gaussian',[15 15],5);  
J=roifilt2(h,I,BW);  
imshow(uint8(J))
```



```
h4=[0 -1 0; -1 5 -1; 0 -1 0]  
L = roifilt2(h4,I,BW);  
figure; imshow(uint8(L));
```



# Procesamiento de ROIs



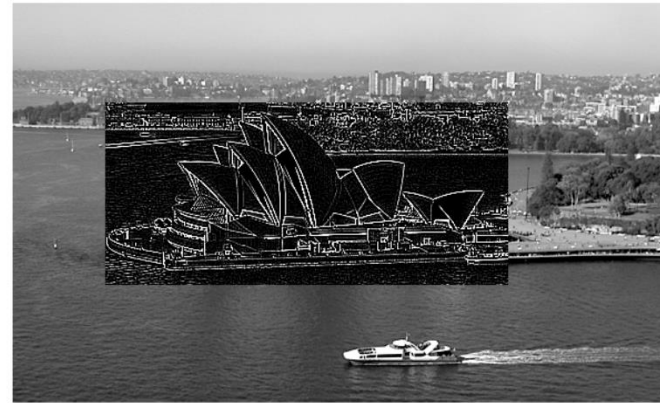
(a)



(b)



(c)



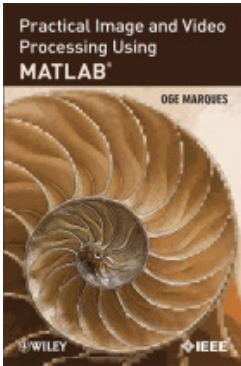
(d)

De izquierda a derecha y de arriba abajo: (a) original, (b) filtrado gaussiano sobre región de interés, (c) HPF sobre región de interés, (d) Laplaciana sobre región de interés

# Combinación de métodos de mejora

- En problemas reales ¿qué técnica se debe usar?
- No existe respuesta universal a esta pregunta
  - La mayoría de las soluciones de procesamiento de imágenes son específicas para problemas concretos y pretenden alcanzar un objetivo determinado.
  - Normalmente la selección de un algoritmo y ajuste de los parámetros es un proceso de prueba y error.

# Bibliografía



Oge Marques, Practical Image and Video Processing Using Matlab, 2011

<http://www.ogemarques.com/>