

Filtros y Convolución

Métodos Avanzados de Procesamiento y Síntesis de Imágenes

Departamento de Computación

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

4to bimestre 2024

Operadores Puntuales

Métodos en el dominio espacial

Sean f y g dos imágenes:

$$f : V \rightarrow W$$

$$g : V \rightarrow W$$

con $V \subset \mathbb{Z}^2$ y $W \subseteq [0, \dots, L - 1]$, donde L es la cantidad de niveles de gris

Sean r y s los niveles de gris de f y g respectivamente:

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

$$g(x, y) = s$$

Operadores Puntuales

Un operador de procesamiento en el dominio espacial es tal que se aplica a una imagen de entrada f y produce una imagen de salida g :

$$T : W \rightarrow W$$

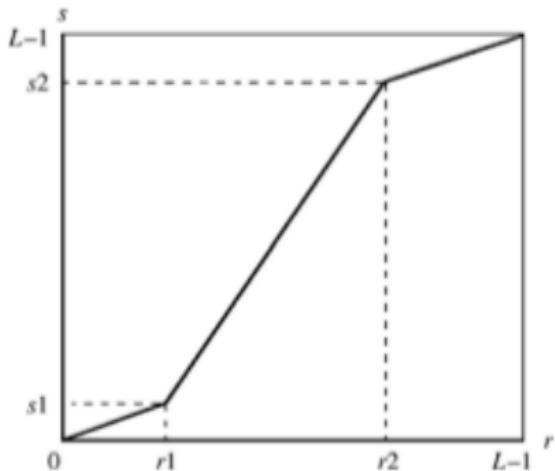
$$f(x, y) \rightarrow \boxed{T} \rightarrow g(x, y)$$

$$s = T(r).$$

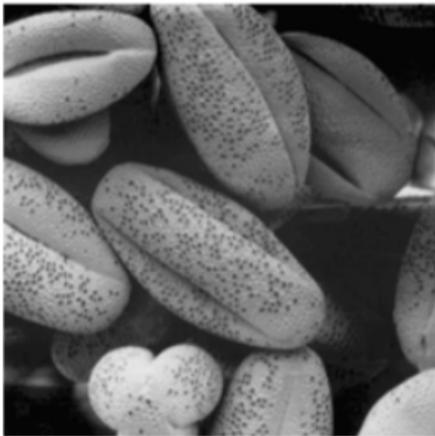
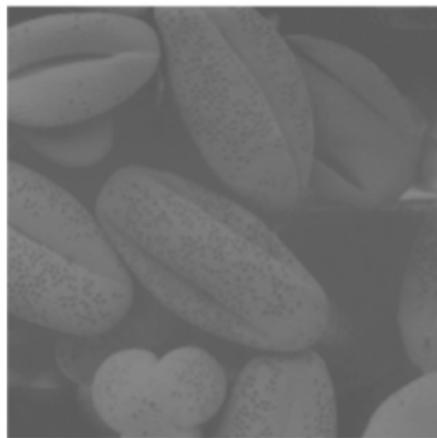
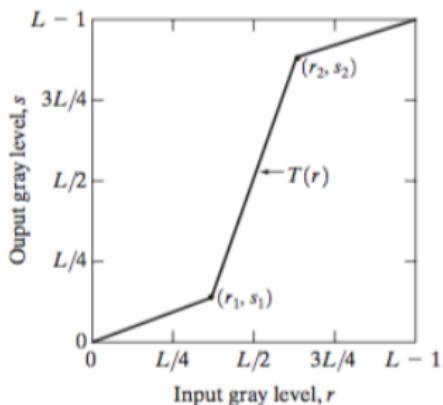
La transformación T también puede operar sobre un conjunto de imágenes, por ejemplo, en el caso de sumar o promediar imágenes.

Aumento del contraste

Este es un ejemplo donde se oscurece la imagen para los valores $r \leq r1$ y se ilumina para los valores de $r \geq r2$.

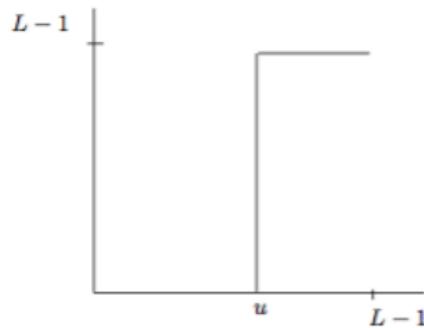


$$T(r) = \begin{cases} f1 & \text{si } 0 \leq r < r1 \\ f2 & \text{si } r1 \leq r \leq r2 \\ f3 & \text{si } r > r2 \end{cases}$$

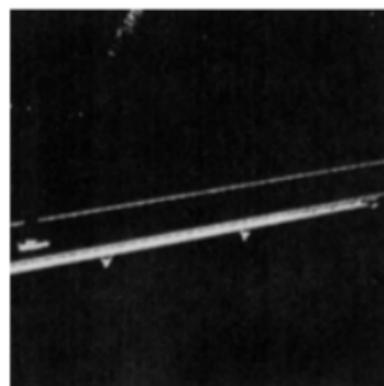
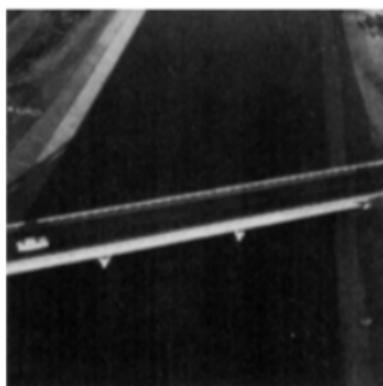
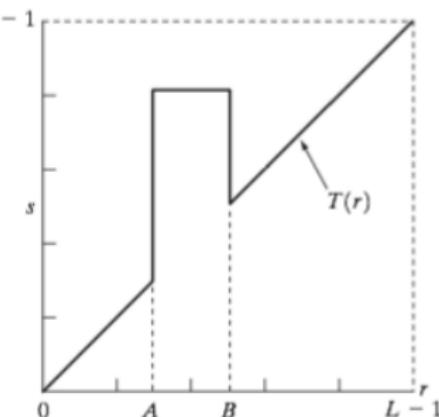
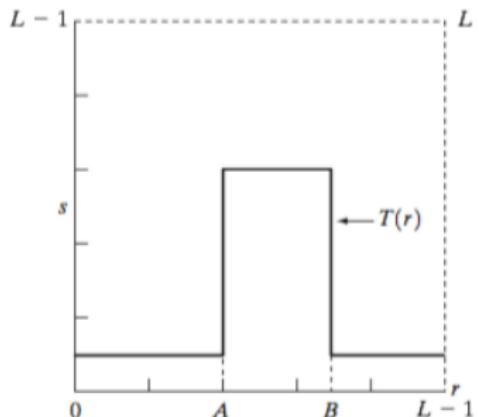


Umbralización

La transformación de umbralización convierte la imagen de entrada en una imagen binaria, utilizando un umbral u .



$$T(r) = \begin{cases} 0 & \text{si } r \leq u \\ 255 & \text{en otro caso} \end{cases}$$



Filtros y Convolución

Negativos de imágenes

La transformación de la Figura 3 calcula el negativo de la imagen de entrada



$$T(r) = -r + L - 1$$

Compresión de rango dinámico

Cuando los niveles de gris son muy bajos y la imagen es muy oscura es conveniente utilizar una transformación logarítmica, que logra que los niveles de gris cercanos a cero se vean más brillantes.

$$T(r) = c \cdot \log(1 + r)$$

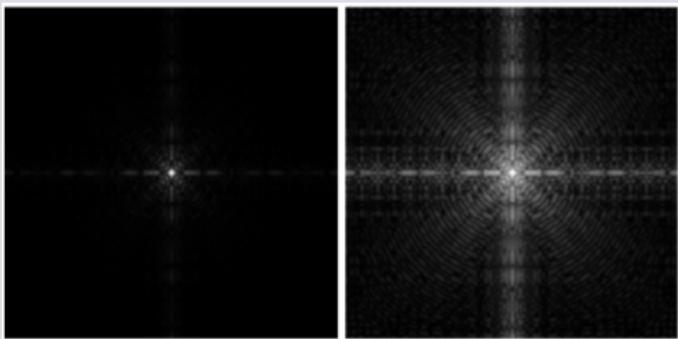
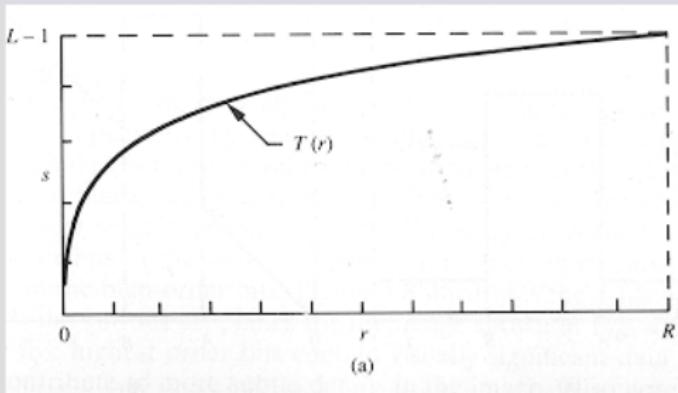
tal que

$$T(0) = c \cdot \log(1) = 0$$

$$T(R) = c \cdot \log(1 + R) = L - 1$$

entonces la constante c viene dada por:

$$c = \frac{L - 1}{\log(1 + R)}$$



¿Qué es un filtro?

- Principal diferencia con operadores puntuales: los filtros generalmente utilizan más de un píxel de la imagen de entrada para obtener cada píxel de la imagen de salida.
- El filtrado espacial modifica una imagen reemplazando el valor de cada píxel por una función de los valores del píxel y sus vecinos.
- Si la función aplicada es lineal entonces es un filtro lineal espacial.

Filtros

Ejemplo: suavizado de una imagen

- No puede obtenerse a partir de operadores puntuales



Filtros y Convolución

Filtros

Ejemplo: suavizado de una imagen

- No puede obtenerse a partir de operadores puntuales
- Para determinar un píxel $I'(u, v)$ en la imagen de salida I' , utilizamos el píxel original $I(u, v) = p_0$ de la imagen de entrada I más sus vecinos p_1, p_2, \dots, p_8 .



Filtros y Convolución

Filtros

Ejemplo: suavizado de una imagen

- No puede obtenerse a partir de operadores puntuales
- Para determinar un píxel $I'(u, v)$ en la imagen de salida I' , utilizamos el píxel original $I(u, v) = p_0$ de la imagen de entrada I más sus vecinos p_1, p_2, \dots, p_8 .
- Idea: reemplazar cada píxel por un promedio de píxeles vecinos

$$I'(u, v) \leftarrow \frac{p_0 + p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8}{9}$$



Filtros y Convolución

Filtros

Ejemplo: suavizado de una imagen

- De forma más compacta:

$$I'(u, v) \leftarrow \frac{1}{9} \cdot \sum_{i=-1}^1 \sum_{j=-1}^1 I(u + i, v + j)$$

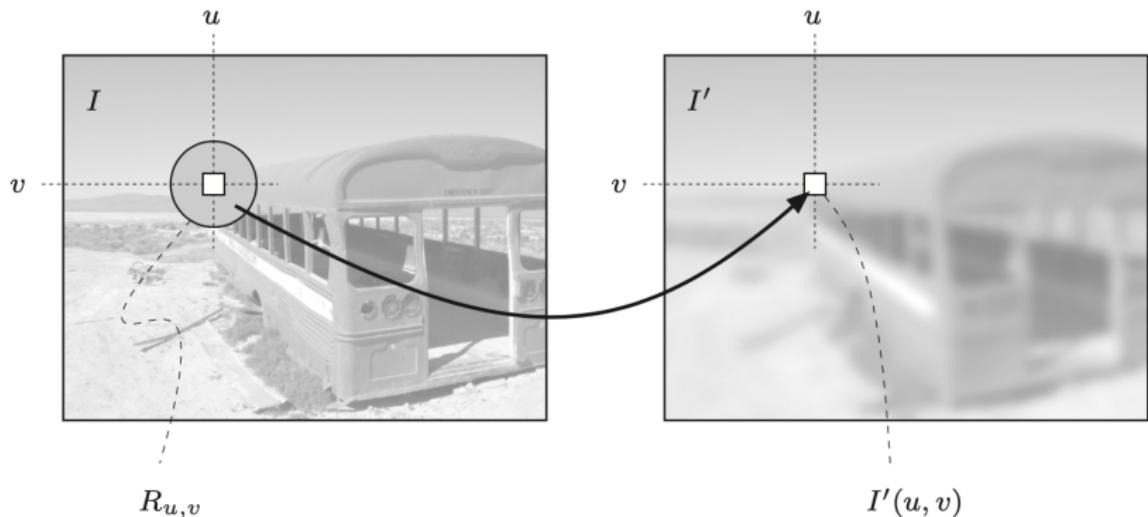
- Filtro lineal: realiza una combinación lineal de los vecinos.



Filtros

Ejemplo: suavizado de una imagen

$$I'(u, v) \leftarrow \frac{1}{|R_{u,v}|} \cdot \sum_{(i,j) \in R_{u,v}} I(u + i, v + j)$$



$R_{u,v}$

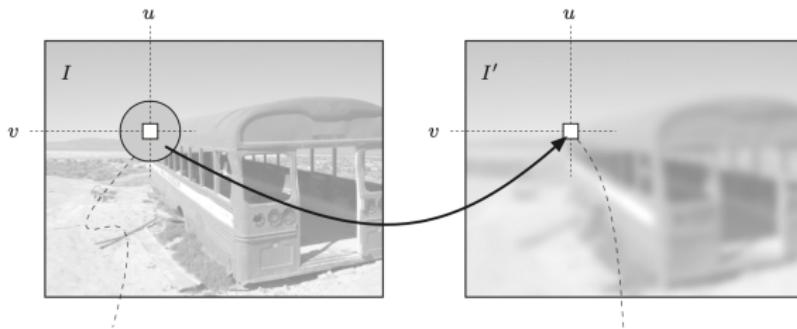
$I'(u, v)$

Filtros

Ejemplo: suavizado de una imagen

$$I'(u, v) \leftarrow \frac{1}{9} \sum_{(i,j) \in R_{u,v}} I(u+i, v+j)$$

- Región de 3×3 pixeles de tamaño. Otros tamaños?
- Forma rectangular. Otras formas?
- Podrían usarse otros pesos para la combinación lineal?



Filtros y Convolución

Matriz H del filtro

- Para cualquier filtro lineal, el tamaño y forma de la región de soporte se especifica por una matriz H del filtro.
- En nuestro ejemplo:

$$H = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Aplicación del filtro

- La matriz de filtro H se mueve sobre la imagen original I de manera que su origen $H(0, 0)$ coincida con la posición actual de la imagen (u, v) .
- Todos los coeficientes de filtro $H(i, j)$ se multiplican por el elemento de imagen correspondiente $I(u + i, v + j)$ y se suman los resultados.
- Finalmente, la suma resultante se almacena en la posición actual en la nueva imagen $I'(u, v)$.
- Más formalmente, siendo R_H el conjunto de coordenadas cubiertas por el filtro H :

$$I'(u, v) = \sum_{(i,j) \in R_H} I(u + i, v + j) \cdot H(i, j)$$

Filtros

Aplicación del filtro

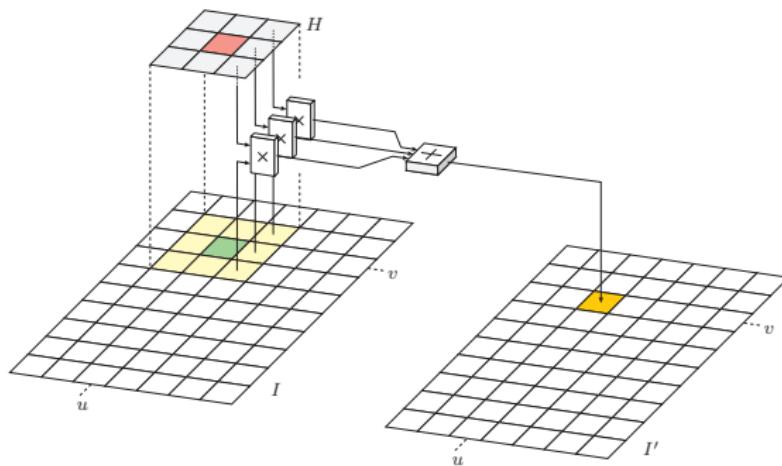
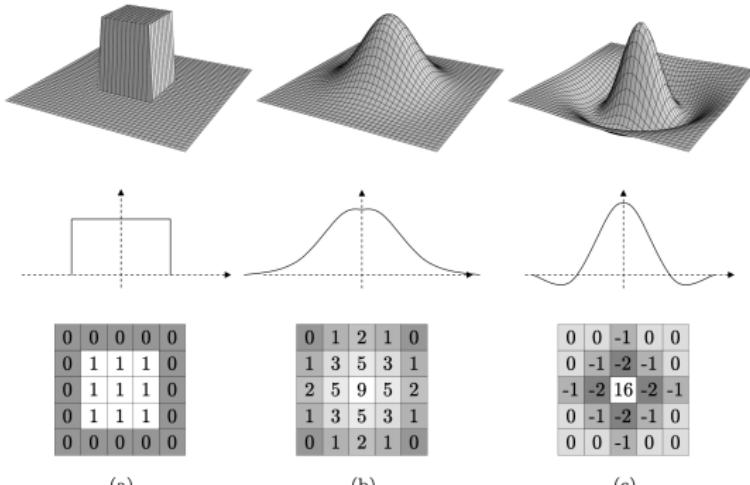


Figure 1: La matriz del filtro H se coloca con su origen en la posición (u, v) en la imagen I . Cada coeficiente de filtro $H(i, j)$ se multiplica por el píxel de imagen correspondiente $I(u + i, v + j)$, los resultados se suman y la suma final se inserta como el nuevo valor de píxel $I'(u, v)$.

Filtros

Ejemplos de filtros

- (a) '*Box filter*'
- (b) Filtro Gaussiano de suavizado
- (c) Filtro de diferencias '*Mexican hat*'

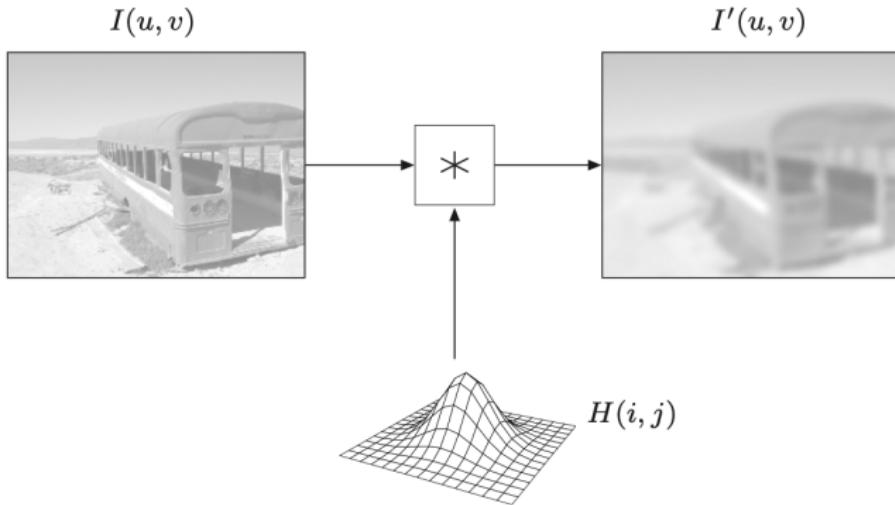


Filtros y Convolución

Convolución

Operador “*” de convolución: $I' = I * H$

$$I'(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(u - i, v - j) \cdot H(i, j)$$



Filtros y Convolución

Convolución

Propiedades

Commutatividad

$$H * I = I * H$$

Linealidad

$$\begin{aligned}(s \cdot I) * H &= I * (s \cdot H) = s \cdot (I * H) \\ (I_1 + I_2) * H &= (I_1 * H) + (I_2 * H) \\ (b + I) * H &\neq b + (I * H)\end{aligned}$$

Asociatividad

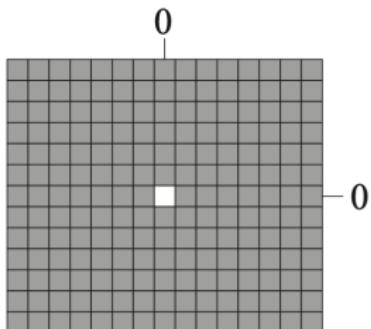
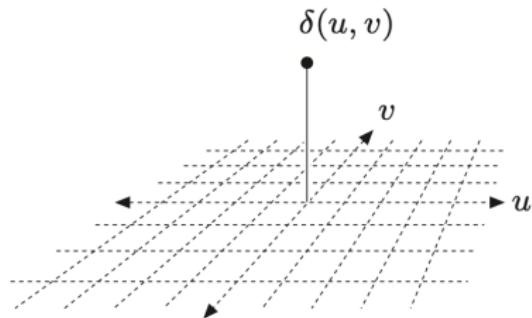
$$(A * B) * C = A * (B * C)$$

Respuesta al impulso

Función impulso

Consideremos la función discreta bidimensional:

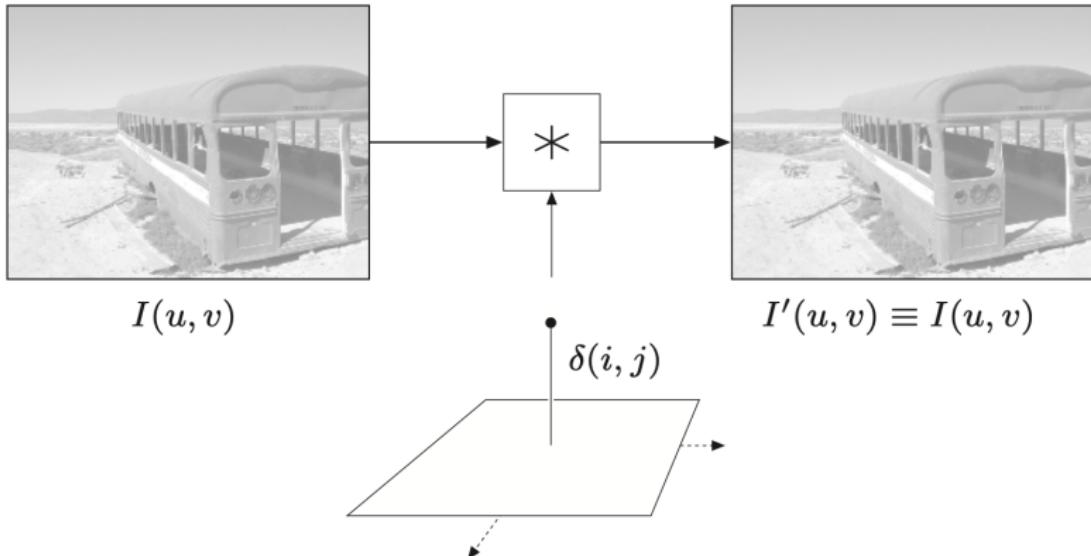
$$\delta(u, v) = \begin{cases} 1 & \text{si } u = v = 0 \\ 0 & \text{en otro caso} \end{cases}$$



Respuesta al impulso

Función impulso

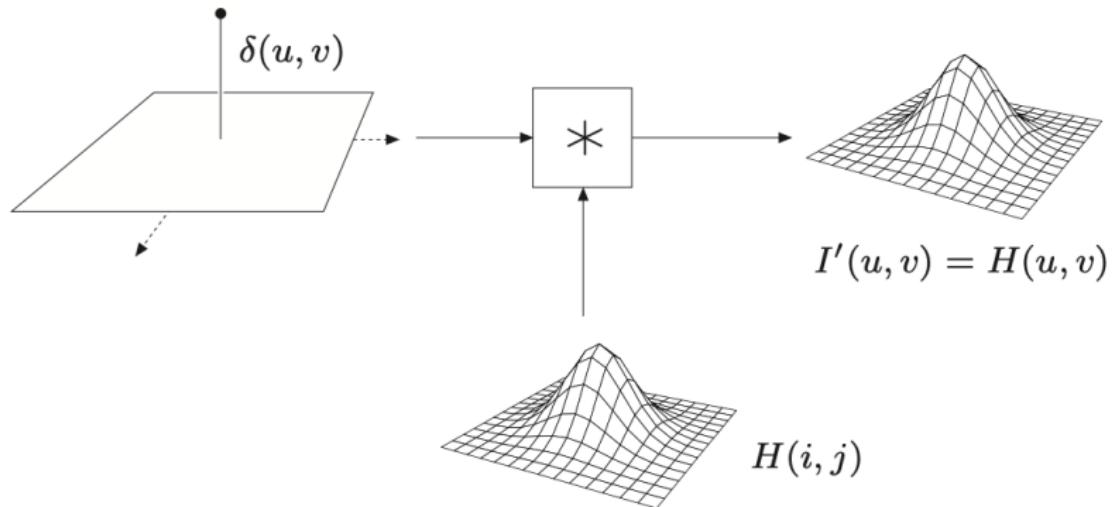
Elemento neutro de la operación de convolución: $\delta * I = I * \delta = I$



Respuesta al impulso

Función impulso

Elemento neutro de la operación de convolución: $\delta * I = I * \delta = I$



Filtros

Ejemplos convolución 1D - Fig 3.29 Gonzalez-Woods

Correlation			Convolution		
(a)	$\begin{array}{cccccc} \swarrow & \text{Origin} & f & & w \\ 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 & \textcolor{teal}{1} & 2 & 4 & 2 & 8 \end{array}$		$\begin{array}{cccccc} \swarrow & \text{Origin} & f & & w \text{ rotated } 180^\circ \\ 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 & \textcolor{teal}{8} & 2 & 4 & 2 & 1 \end{array}$	(i)	
(b)	$\begin{array}{cccccc} \downarrow & & & & & \\ 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 \\ \textcolor{teal}{1} & 2 & 4 & 2 & 8 \end{array}$		$\begin{array}{cccccc} & & & & & \\ 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 \\ \textcolor{teal}{8} & 2 & 4 & 2 & 1 \end{array}$	(j)	
	Starting position alignment		Starting position alignment		
(c)	$\begin{array}{cccccc} \swarrow & \text{Zero padding} & \searrow \\ \boxed{0} & 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \textcolor{teal}{1} & 2 & 4 & 2 & 8 \end{array}$		$\begin{array}{cccccc} \swarrow & \text{Zero padding} & \searrow \\ \boxed{0} & 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \textcolor{teal}{8} & 2 & 4 & 2 & 1 \end{array}$	(k)	
	Starting position		Starting position		
(d)	$\begin{array}{cccccc} & & & & & \\ 0 & 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \textcolor{teal}{1} & 2 & 4 & 2 & 8 \end{array}$		$\begin{array}{cccccc} & & & & & \\ 0 & 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \textcolor{teal}{8} & 2 & 4 & 2 & 1 \end{array}$	(l)	
	Position after 1 shift		Position after 1 shift		
(e)	$\begin{array}{cccccc} & & & & & \\ 0 & 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \textcolor{teal}{1} & 2 & 4 & 2 & 8 \end{array}$		$\begin{array}{cccccc} & & & & & \\ 0 & 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \textcolor{teal}{8} & 2 & 4 & 2 & 1 \end{array}$	(m)	
	Position after 3 shifts		Position after 3 shifts		
(f)	$\begin{array}{cccccc} & & & & & \\ 0 & 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \textcolor{teal}{1} & 2 & 4 & 2 & 8 \end{array}$		$\begin{array}{cccccc} & & & & & \\ 0 & 0 & 0 & 0 & \textcolor{teal}{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \textcolor{teal}{8} & 2 & 4 & 2 & 1 \end{array}$	(n)	
	Final position \uparrow		Final position \uparrow		
Correlation result			Convolution result		
(g)	$0 \ 8 \ 2 \ 4 \ 2 \ 1 \ 0 \ 0 \ 0$		$0 \ 1 \ 2 \ 4 \ 2 \ 8 \ 0 \ 0$		(o)
Extended (full) correlation result			Extended (full) convolution result		
(h)	$0 \ 0 \ 0 \ 8 \ 2 \ 4 \ 2 \ 1 \ 0 \ 0 \ 0 \ 0$		$0 \ 0 \ 0 \ 1 \ 2 \ 4 \ 2 \ 8 \ 0 \ 0 \ 0 \ 0$		(p)

Filtros

Ejemplos convolución 2D - Fig 3.30 Gonzalez-Woods

		Padded f							
Origin f		0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	w	0	0	0	1
0	0	1	2	3		0	0	0	0
0	0	4	5	6		0	0	0	0
0	0	7	8	9		0	0	0	0

(a)

(b)

		Correlation result				Full correlation result			
Initial position for w		0	0	0	0	0	0	0	0
1	2	3	0	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(c)

(d)

(e)

		Convolution result				Full convolution result			
Rotated w		0	0	0	0	0	0	0	0
9	8	7	0	0	0	0	0	0	0
6	5	4	0	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(f)

(g)

(h)

Filtros y Convolución

Separabilidad

Filtros y funciones separables

- Una función bidimensional $G(x, y)$ se dice *separable* si puede expresarse como el producto de dos funciones unidimensionales: $G(x, y) = G_1(x) \cdot G_2(y)$.
- Una matriz de filtro H es *separable* si puede expresarse como el producto externo de dos vectores h_1 y h_2 : $H = h_1 h_2^t$

Ejemplo de nuestro filtro suavizante

$$H_{xy} = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix} = h_y h_x$$

Propiedad: $H_{xy} = h_y h_x = h_y * h_x = h_x * h_y$

Aplicación secuencial: $I' \leftarrow (I * h_x) * h_y = I * \underbrace{(h_x * h_y^t)}_{H_{xy}}$

Filtros Pasa bajo

Suavizan la imagen, dejan pasar las frecuencias bajas y atenúan las frecuencias altas.

- media

$$A = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad B = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad C = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

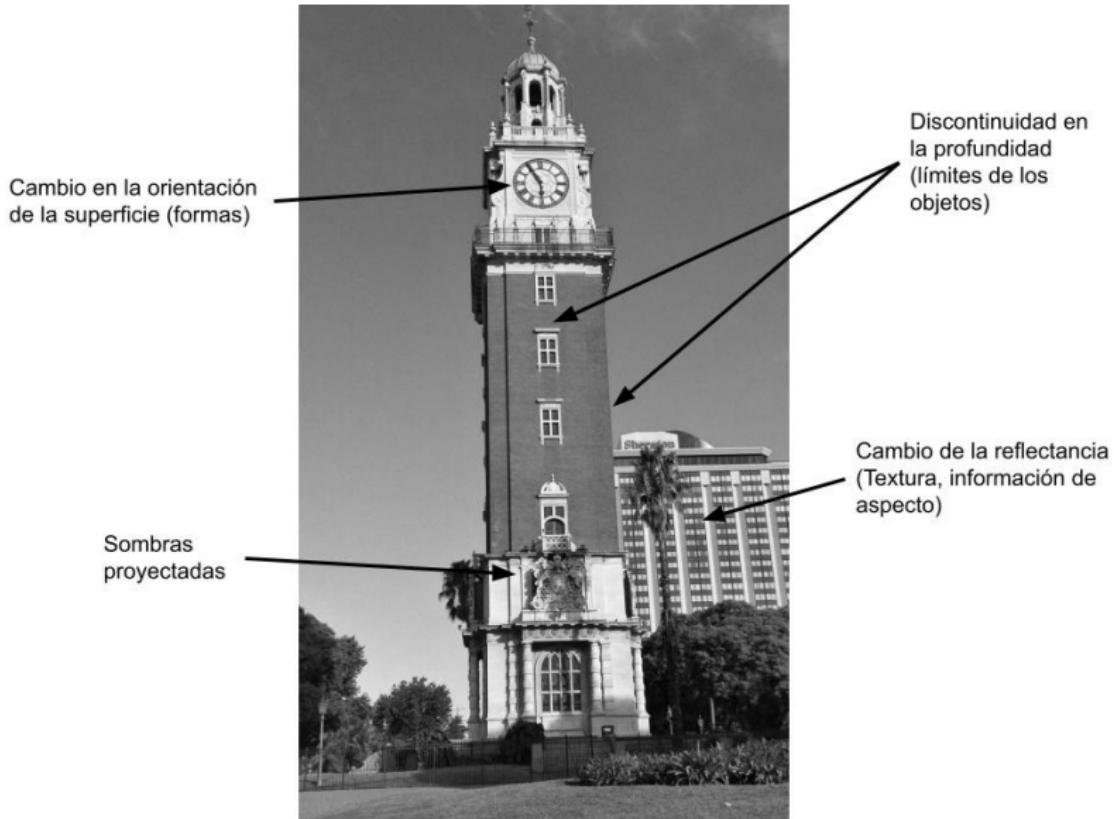
- mediana: filtro no-lineal, ordena la vecindad del pixel y lo reemplaza por el valor central.

Detección de Bordes

¿Cuál es el objetivo de detectar bordes?

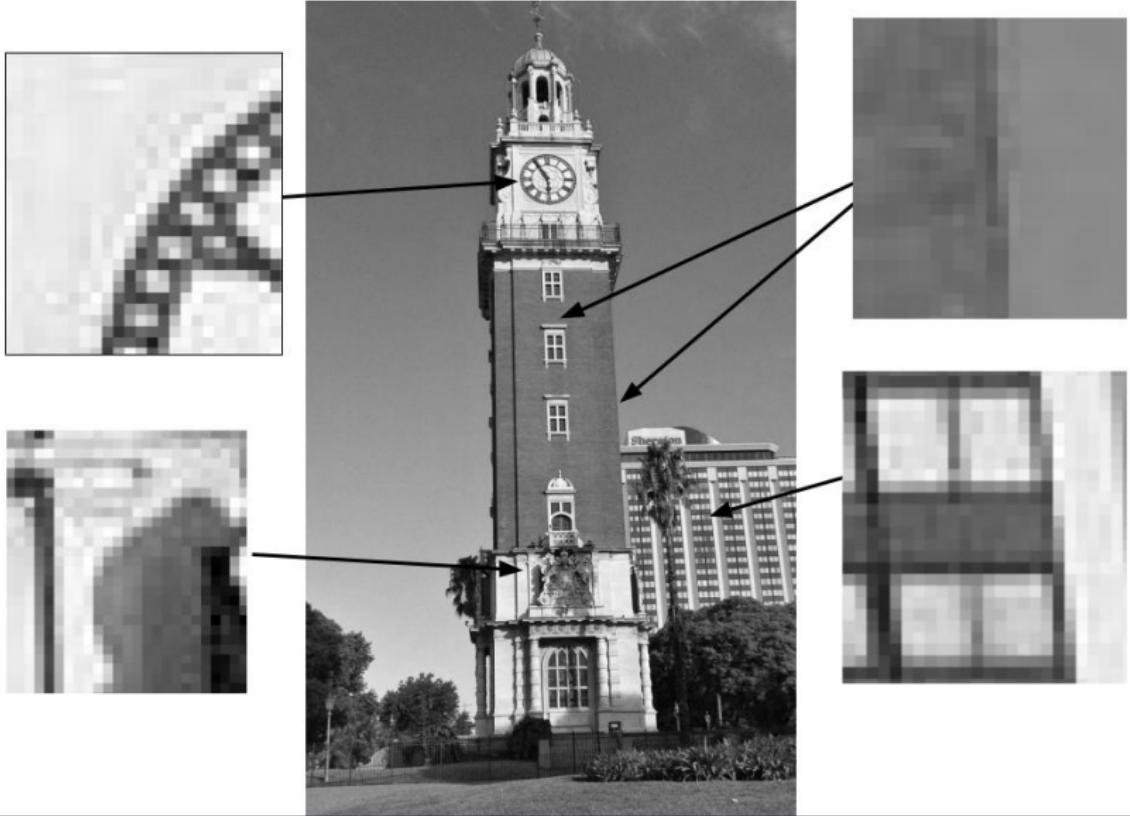
- mapear una imagen 2D a un conjunto de curvas o segmentos de línea o contornos.
- IDEA: buscar segmentos definidos, líneas, curvas posterior al procesamiento.

Qué hace que en una imagen haya bordes?



Filtros y Convolución

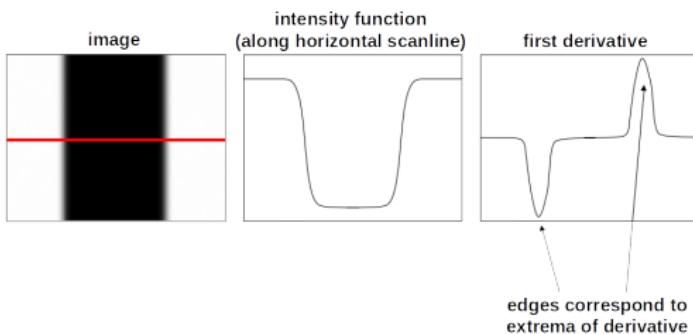
Dónde se distinguen bordes y dónde no?



Filtros y Convolución

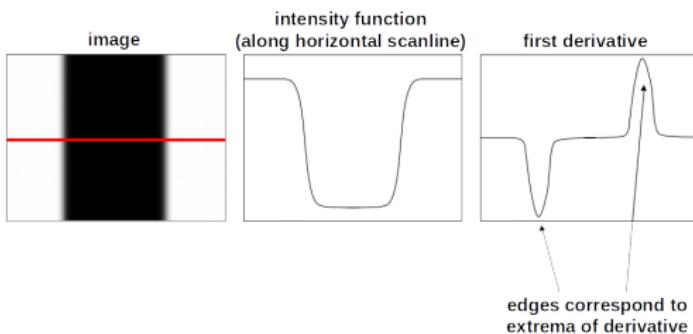
Derivadas y Bordes

- Un borde es el límite entre dos regiones con propiedades de nivel de gris relativamente distintas.
- La base de la mayoría de las técnicas de detección de bordes es el cálculo de un operador derivado local.



Derivadas y Bordes

- Un borde es el límite entre dos regiones con propiedades de nivel de gris relativamente distintas.
- La base de la mayoría de las técnicas de detección de bordes es el cálculo de un operador derivado local.
- La magnitud de la primera derivada calculada dentro de una vecindad alrededor del píxel de interés, se puede utilizar para detectar la presencia de un borde en una imagen.



Métodos de basados en el cálculo del gradiente

- Se trata de localizar los cambios abruptos en la función intensidad de la imagen.
- Los cambios en las funciones continuas se describen mediante las derivadas.
- En una imagen se considera el gradiente que describe la variación de la función.

Gradiente

-

$$\nabla f = \begin{pmatrix} f_x \\ f_y \end{pmatrix} = \begin{pmatrix} f(x+1, y) - f(x, y) \\ f(x, y+1) - f(x, y) \end{pmatrix}$$

donde $f_x = \frac{\partial f}{\partial x}$ y $f_y = \frac{\partial f}{\partial y}$

- *Magnitud* o intensidad del borde
- *Dirección* del borde

Magnitud y Dirección de ∇f

- *Magnitud*

$$|\nabla f| = \sqrt{f_x^2 + f_y^2}$$

- *Dirección*

$$\alpha = \arctan\left(\frac{f_x}{f_y}\right)$$

La magnitud y la dirección de f son dos imágenes del mismo tamaño que f .

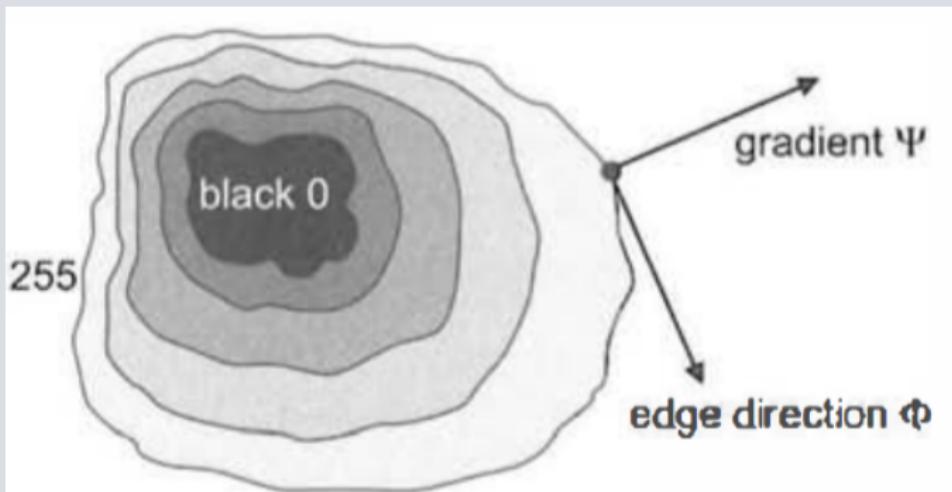
Detección del borde

Detectar un borde consiste en encontrar, para cada (x, y) :

- $|\nabla f(x, y)|$,
- $\alpha(x, y)$, **ángulo** que forma el vector gradiente con el eje x

Dirección del borde

El gradiente es perpendicular a la dirección del borde



La dirección del gradiente es la dirección de máximo crecimiento de la función, o sea, es la dirección de máxima variación.

Los pixels con más alto valor de magnitud son puntos de borde.

Si consideramos una ventana de 2×2 , alrededor de cada punto (i, j) de modo que:

$$\left(\begin{array}{c|c} f(i, j) & f(i, j + 1) \\ \hline f(i + 1, j) & f(i + 1, j + 1) \end{array} \right)$$

en este caso una manera de aproximar la derivada primera es mediante:

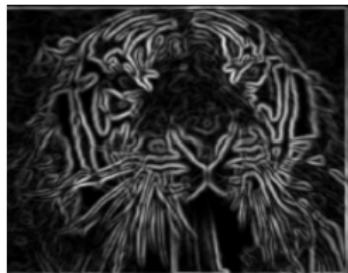
$$\begin{aligned}\frac{\partial f}{\partial x} &= f(i + 1, j) - f(i, j) \\ \frac{\partial f}{\partial y} &= f(i, j + 1) - f(i, j)\end{aligned}$$

otra forma es haciendo diferencias cruzadas:

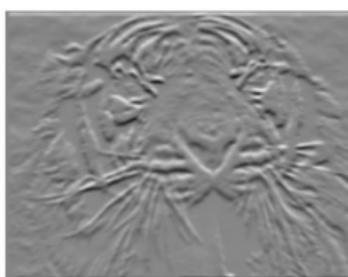
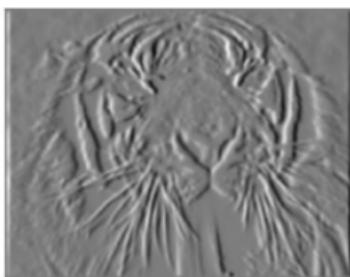
$$\begin{aligned}\frac{\partial f}{\partial x} &= f(i, j) - f(i + 1, j + 1) \\ \frac{\partial f}{\partial y} &= f(i, j + 1) - f(i + 1, j)\end{aligned}$$

Derivadas parciales de una imagen

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



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



-1	1
----	---

-1	?	or	1
1			-1

Cuál muestra cambios respecto del eje x?

Roberts



Original Image

$$h_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad h_2 = [1 \quad -1]$$

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Si en cambio son de 3×3 , alrededor del punto (i, j)

$$\left(\begin{array}{c|c|c} f(i-1, j-1) & f(i-1, j) & f(i-1, j+1) \\ \hline f(i, j-1) & f(i, j) & f(i, j+1) \\ \hline f(i+1, j-1) & f(i+1, j) & f(i+1, j+1) \end{array} \right)$$

que se puede aproximar:

$$f_x = f_{x_3} - f_{x_1} \quad f_y = f_{y_3} - f_{y_1}$$

donde:

$$f_{x_3} = f(i+1, j-1) + f(i+1, j) + f(i+1, j+1)$$

$$f_{x_1} = f(i-1, j-1) + f(i-1, j) + f(i-1, j+1)$$

$$f_{y_3} = f(i, j+1) + f(i, j+1) + f(i+1, j+1)$$

$$f_{y_1} = f(i-1, j-1) + f(i, j-1) + f(i+1, j-1)$$

f_{x_k} son las filas y f_{y_k} son las columnas, con $k = 1, 2, 3$.

Operadores gradientes

Roberts

Utiliza una vecindad del pixel de 2×2 , las máscaras de convolución son:

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

Es simple de calcular la magnitud del borde, pero tiene la desventaja de ser muy sensible al ruido, debido a usar muy pocos pixels para aproximar el gradiente.

La magnitud del gradiente se puede calcular como:

$$|f(i, j) - f(i + 1, j + 1)| + |f(i, j + 1) - f(i + 1, j)|$$

Operadores gradientes isotrópicos

Prewitt

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Sobel

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

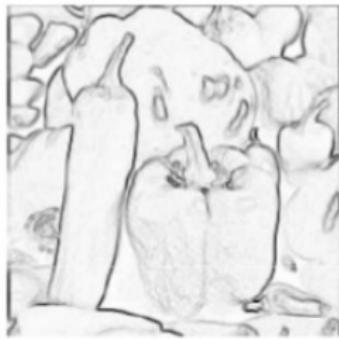
Isotrópico

$$\begin{pmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{pmatrix}$$

Propiedades

- Los operadores Prewitt, Sobel e Isotrópico calculan las diferencias horizontales y verticales de las sumas en un vecindario del pixel.
- Esto reduce el efecto del ruido en la imagen, y tienen la propiedad de ser cero en las regiones de intensidad uniforme.

Ejemplo Prewitt-Sobel



Prewitt

$$h_1 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$



Sobel

$$h_1 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

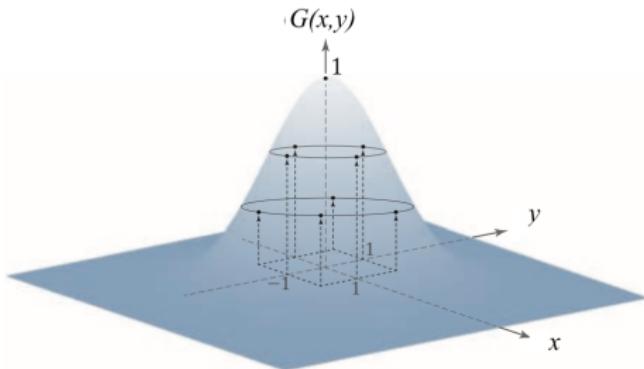
Filtro Gaussiano - *Smoothing*

Definición

- Siendo σ el ancho (desviación estándar) de la función en forma de campana y r es la distancia (radio) desde el centro:

$$G_\sigma(x, y) = e^{-\frac{r^2}{2\sigma^2}} = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Matriz H^σ del filtro gaussiano —> discretización de G .



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

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

Filtro Gaussiano

Separabilidad del filtro gaussiano

- La función que lo define es separable:

$$G_\sigma(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} = e^{-\frac{x^2}{2\sigma^2}} \cdot e^{-\frac{y^2}{2\sigma^2}} = g_\sigma(x) \cdot g_\sigma(y)$$

- La convolución con un filtro gaussiano se puede implementar con dos filtros:

$$I' \leftarrow I * H^\sigma = I * h_x^\sigma * h_y^\sigma$$

Padding



(a)



(b)



(c)



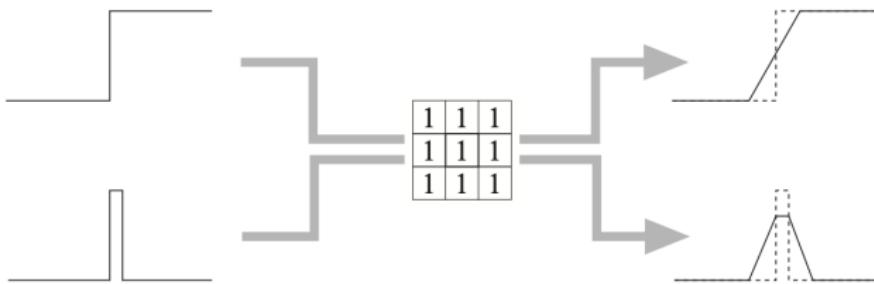
(d)

Filtros y Convolución

Filtros no lineales

Problema de los filtros lineales

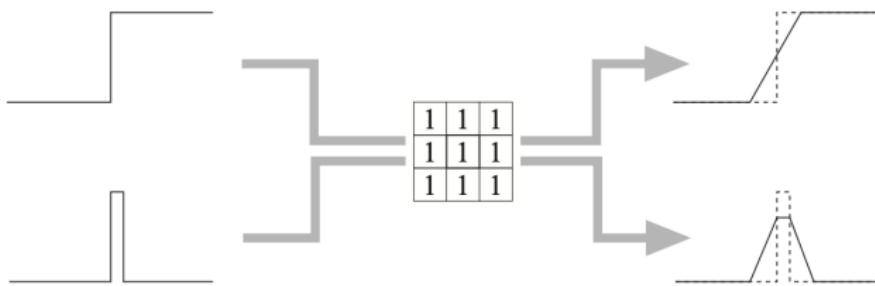
- Cuando se los utiliza, por ejemplo, para suavizar o eliminar ruido, todas las estructuras (puntos, líneas, bordes) también son suavizadas.



Filtros no lineales

Problema de los filtros lineales

- También se utiliza un esquema de *ventana deslizante*.
- La operación sobre los píxeles de la ventana es no lineal



Filtros no lineales

Filtro “mínimo”

- $I'(u, v) \leftarrow \min\{I(u + i, v + j) | (i, j) \in R\}$

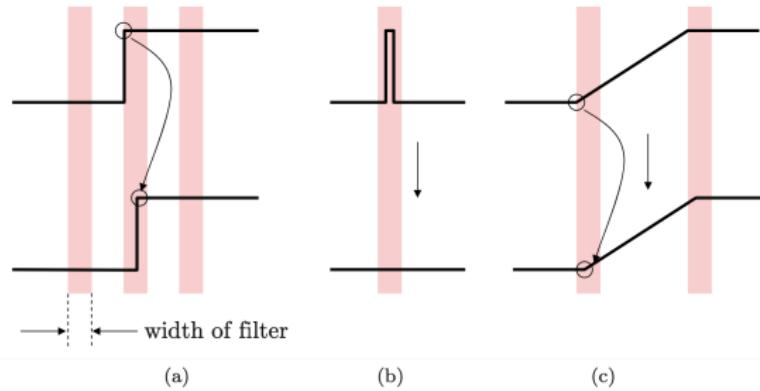
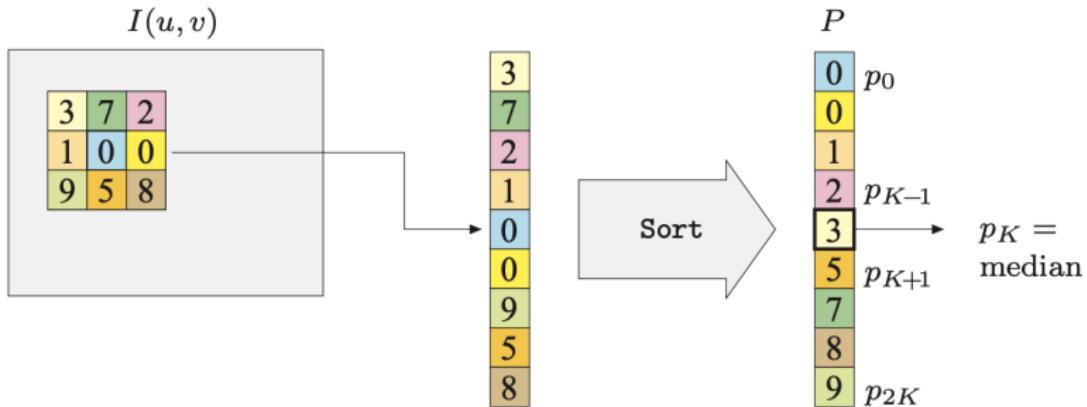


Figure 2: Efectos de un filtro mínimo unidimensional en varias estructuras de señales. Señal original (arriba) y resultado después del filtrado (abajo), donde las barras de colores indican la extensión del filtro. El borde del escalón (a) y la rampa lineal (c) se desplazan hacia la derecha la mitad del ancho del filtro, y el pulso estrecho (b) se elimina por completo.

Filtros no lineales

Filtro de la mediana

- $I'(u, v) \leftarrow \text{median}\{I(u + i, v + j) \mid (i, j) \in R\}$



Filtros no lineales

Filtro de la mediana ponderada

- Se agrega una matriz de pesos que indica la ‘cantidad de votos’ de cada pixel en la ventana.

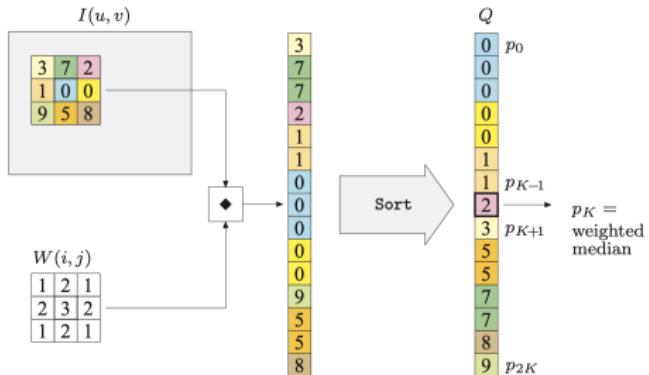


Figure 3: Cada valor de píxel se inserta en el vector de píxel extendido según lo especificado por la matriz de pesos W . Por ejemplo, el valor 0 del píxel central se inserta tres veces (ya que $W(0, 0) = 3$) y el valor de píxel 7 dos veces. El vector de píxeles se ordena y el valor central (2) se toma como mediana.

Ejemplo de aplicación: *Sharpening*

Unsharp Masking

- Proviene de la fotografía, donde la nitidez de una imagen era mejorada utilizando una copia suavizada (“unsharp”) de la misma.

Proceso de Unsharp Masking

- La máscara M se genera restando una versión suavizada de la imagen I del original (G es un filtro gaussiano),

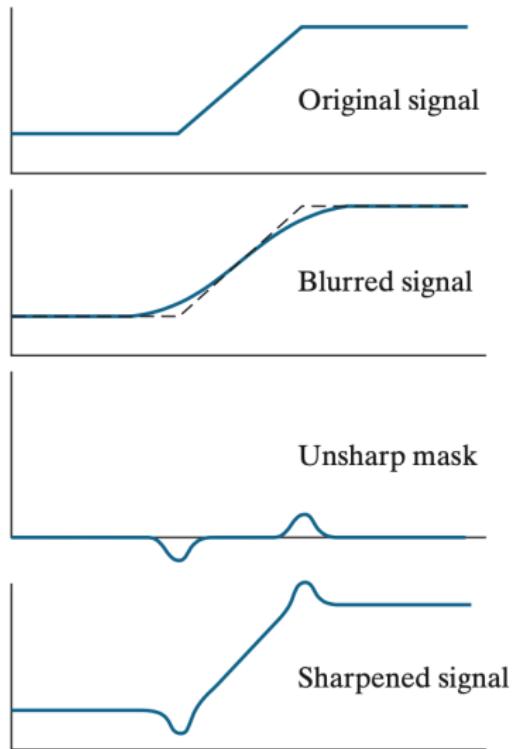
$$M \leftarrow I - (I * G) = I - \tilde{I}$$

- Para obtener la imagen más nítida \hat{I} , se suma la máscara M a la imagen original I , ponderada por el factor a , que controla la cantidad de nitidez,

$$\hat{I} \leftarrow I + a \cdot M$$

- Lo anterior es equivalente a: $\hat{I} \leftarrow I + a(I - \tilde{I}) = (1 + a)I - a\tilde{I}$.

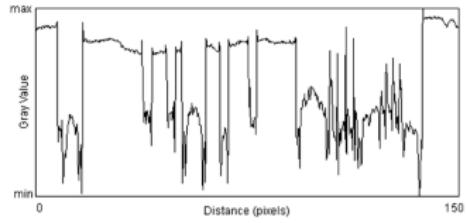
Ejemplo de aplicación: *Sharpening*



Ejemplo de aplicación: *Sharpening*



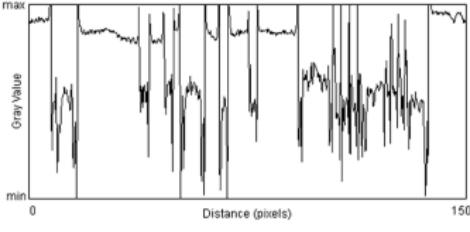
(a) Original



(b)



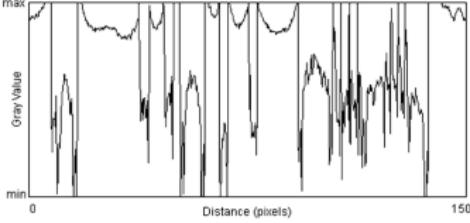
(c) $\sigma = 2.5$



(d)



(e) $\sigma = 10.0$



(f)

Filtros y Convolución

Ejemplo de aplicación: *Sharpening*



(g) Original



(h) $\sigma = 2.5$



(i) $\sigma = 10.0$