Filtros

Procesamiento Digital de Imágenes

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

1er Cuatrimestre 2021

Filtros

¿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
- 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 $p_1, p_2, \ldots, p_8$.
- 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

Ejemplo: suavizado de una imagen

- De forma más compacta:

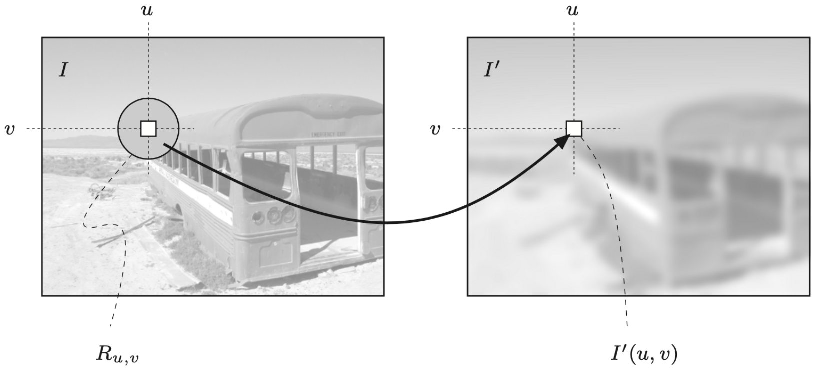
$$I'(u,v) \leftarrow \frac{1}{9} \cdot \sum_{i=-1}^{1} \sum_{j=-1}^{1} I(u+i, v+j)$$

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

Filtros

Ejemplo: suavizado de una imagen

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

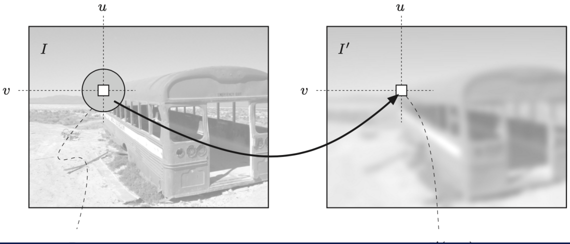


Filtros

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

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:

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

Filtros

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

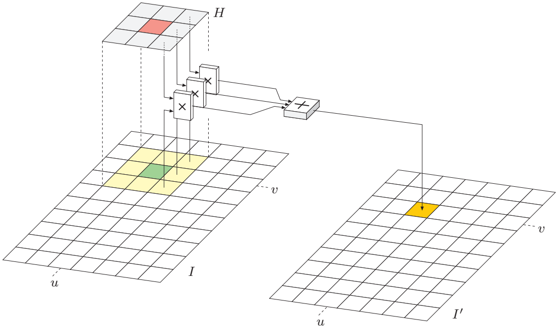


Figure 1: La matriz del filtro H se coloca con su origen en la posición (u, v) en la imagen I. Cada coeficiente de filtro $H(i, j)$ se multiplica por el píxel de imagen correspondiente $I(u + i, v + j)$, los resultados se suman y la suma final se inserta como el nuevo valor de píxel $I'(u, v)$.

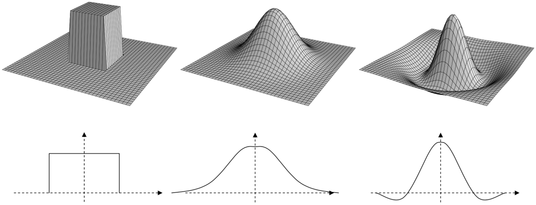
Filtros

Ejemplos de filtros

(a) '*Box filter*'

(b) Filtro Gaussiano de suavizado

(c) Filtro de diferencias '*Mexican hat*'



0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

(a)

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

(b)

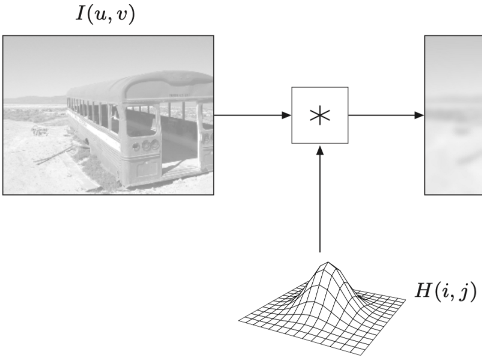
0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

(c)

Convolución

Operador "$*$" de convolución: $I' = I * H$

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



Convolución

Propiedades

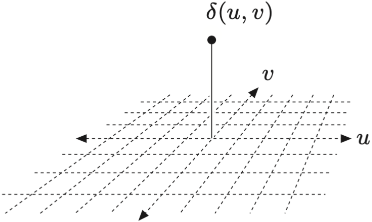
Conmutatividad	$H * I = I * H$
Linealidad	$(s \cdot I) * H = I * (s \cdot H) = s \cdot (I * H)$ $(I_1 + I_2) * H = (I_1 * H) + (I_2 * H)$ $(b + I) * H \neq b + (I * H)$
Asociatividad	$(A * B) * C = A * (B * C)$

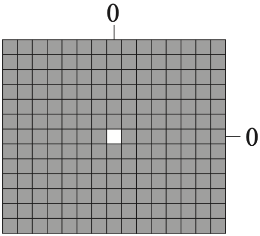
Respuesta al impulso

Función impulso

Consideremos la función discreta bidimensional:

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

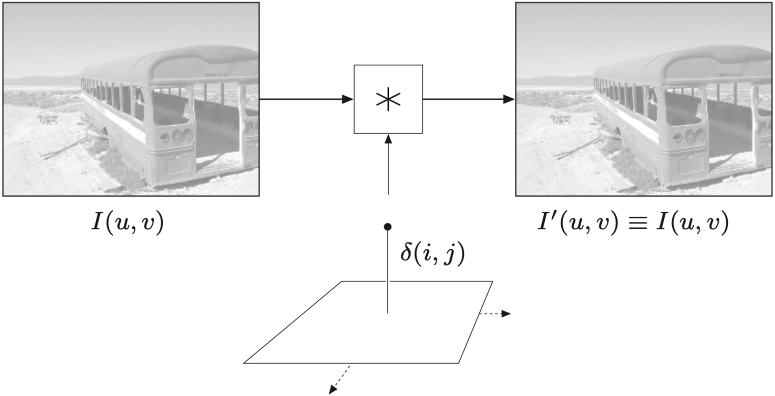




Respuesta al impulso

Función impulso

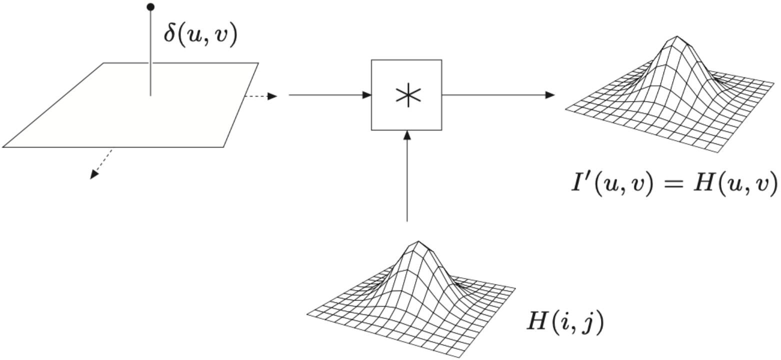
Elemento neutro de la operación de convolución: $\delta * I = I * \delta = I$



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

(a) Origin f w
0 0 0 **1** 0 0 0 0 **1 2 4 2 8**

(b)
0 0 0 **1** 0 0 0 0
1 2 4 2 8
Starting position alignment

(c) Zero padding
0 0 0 0 0 **1** 0 0 0 0 0 0 0
1 2 4 2 8
Starting position

(d)
0 0 0 0 0 **1** 0 0 0 0 0 0 0
1 2 4 2 8
Position after 1 shift

(e)
0 0 0 0 0 **1** 0 0 0 0 0 0 0
1 2 4 2 8
Position after 3 shifts

(f)
0 0 0 0 0 **1** 0 0 0 0 0 0 0
1 2 4 2 8
Final position

Correlation result

(g) 0 8 2 4 2 1 0 0

Extended (full) correlation result

(h) 0 0 0 8 2 4 2 1 0 0 0 0

Convolution

Origin f w rotated 180°
0 0 0 **1** 0 0 0 0 **8 2 4 2 1** (i)

0 0 0 **1** 0 0 0 0 (j)
8 2 4 2 1
Starting position alignment

Zero padding
0 0 0 0 0 **1** 0 0 0 0 0 0 0 (k)
8 2 4 2 1
Starting position

0 0 0 0 0 **1** 0 0 0 0 0 0 0 (l)
8 2 4 2 1
Position after 1 shift

0 0 0 0 0 **1** 0 0 0 0 0 0 0 (m)
8 2 4 2 1
Position after 3 shifts

0 0 0 0 0 **1** 0 0 0 0 0 0 0 (n)
8 2 4 2 1
Final position

Convolution result

0 1 2 4 2 8 0 0 (o)

Extended (full) convolution result

0 0 0 1 2 4 2 8 0 0 0 0 (p)

Filtros

Ejemplos convolución 2D - Fig 3.30 Gonzalez-Woods

Origin f

0	0	0	0	0
0	0	0	0	0
0	0	**1**	0	0
0	0	0	0	0
0	0	0	0	0

w

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

(a)

Padded 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	**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

(b)

Initial position for w

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

(c)

Correlation result

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

(d)

Full correlation result

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

(e)

Rotated w

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

(f)

Convolution result

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

(g)

Full convolution result

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

(h)

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

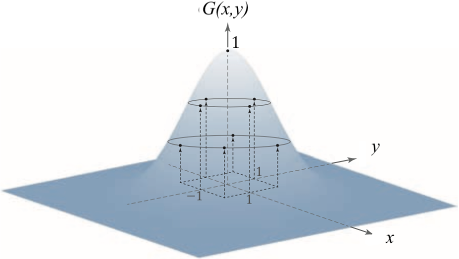
Filtro Gaussiano - *Smoothing*

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 $\longrightarrow$ 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)

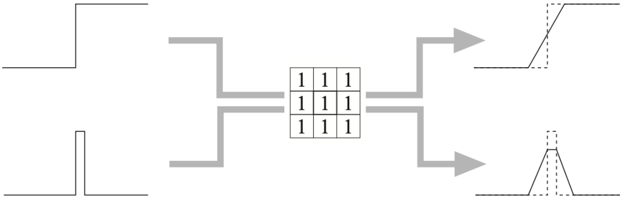
(c)

(d)

Filtros no lineales

Problema de los filtros lineales

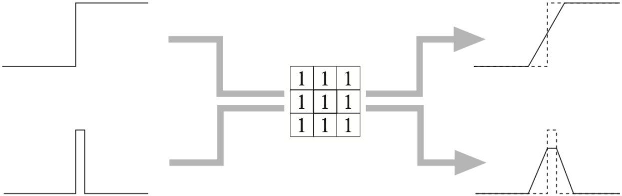
- Cuando se los utiliza, por ejemplo, para suavizar o eliminar ruido, todas las estructuras (puntos, líneas, bordes) también son suavizadas.



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"

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

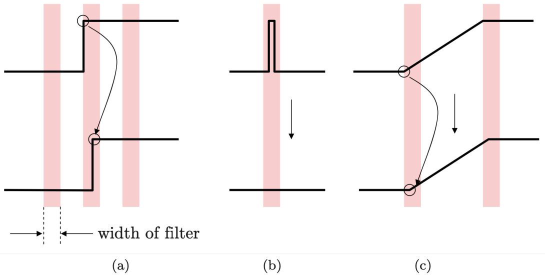
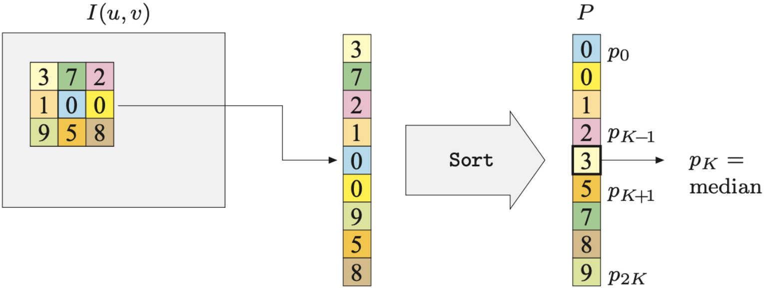


Figure 2: Efectos de un filtro mínimo unidimensional en varias estructuras de señales. Señal original (arriba) y resultado después del filtrado (abajo), donde las barras de colores indican la extensión del filtro. El borde del escalón (a) y la rampa lineal (c) se desplazan hacia la derecha la mitad del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

Filtros no lineales

Filtro de la mediana

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



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.

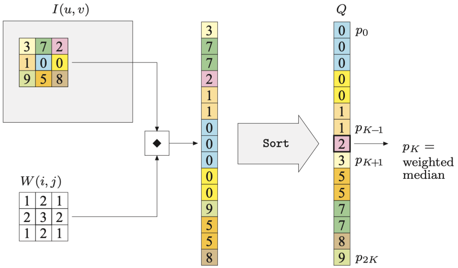


Figure 3: Cada valor de píxel se inserta en el vector de píxel extendido según lo especificado por la matriz de pesos W. Por ejemplo, el valor 0 del píxel central se inserta tres veces (ya que $W(0,0) = 3$) y el valor de píxel 7 dos veces. El vector de píxeles se ordena y el valor central (2) se toma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

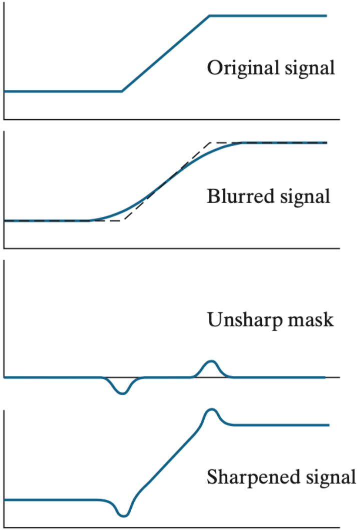
$$M \leftarrow I - (I * G) = I - \tilde{I}$$

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

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

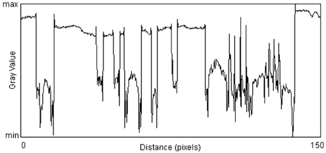
- Lo anterior es equivalente a: $\widehat{I} \leftarrow I + a(I - \tilde{I}) = (1 + a)I - a\tilde{I}$.

Ejemplo de aplicación: *Sharpening*



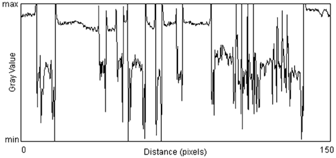
Ejemplo de aplicación: *Sharpening*

(a) Original



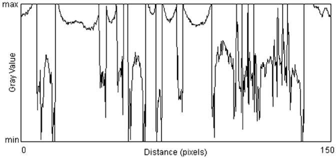
(b)

(c) $\sigma = 2.5$



(d)

(e) $\sigma = 10.0$



(f)

Ejemplo de aplicación: *Sharpening*

(g) Original

(h) $\sigma = 2.5$

(i) $\sigma = 10.0$