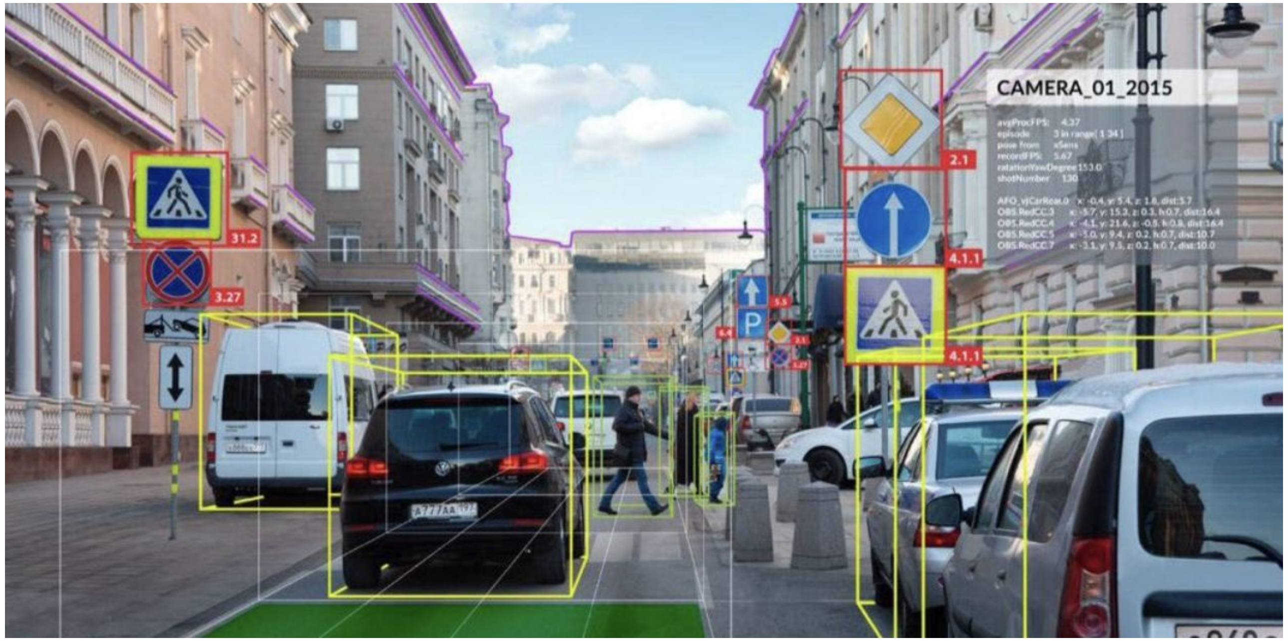
Visión Computacional
para imágenes y video

Módulo 1

Tema 2.2
Convolución y mejoramiento de imágenes



Gilberto Ochoa Ruiz, PhD
Associate Professor
Researcher in Computer Vision

Computer Science Dept.
Advanced AI Research Group
gilberto.ochoa@tec.mx

Tecnológico de Monterrey

Convolución y aplicaciones en procesamiento de imágenes

Representación de imágenes

Localización del pixel (m,n)

Intensidad I(m,n)

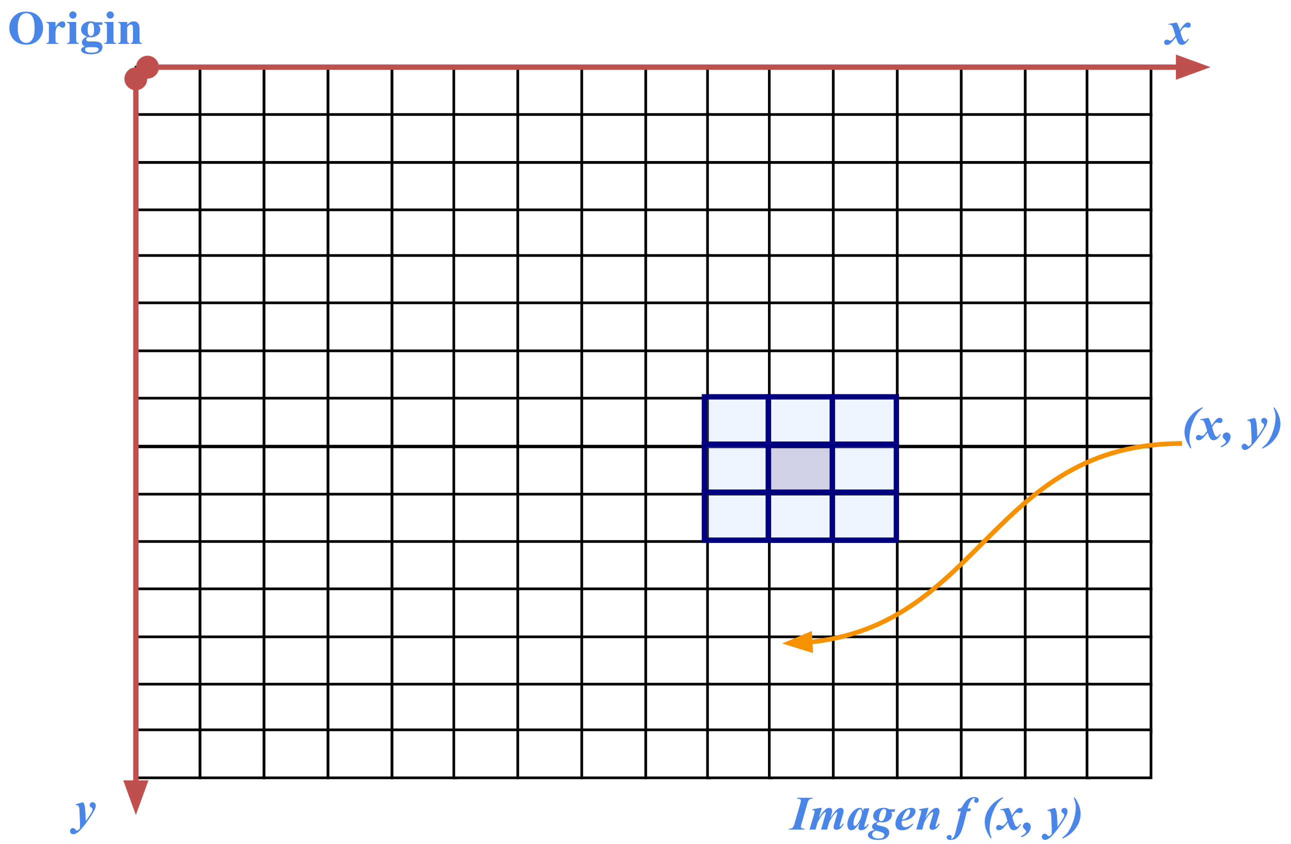
Convolución y aplicaciones en procesamiento de imágenes

Representación de imágenes

La mayoría de las operaciones de de mejoramiento de imágenes en el domiio especial se puede reducir a la forma

g (x, y) = T[f (x, y)] *donde*

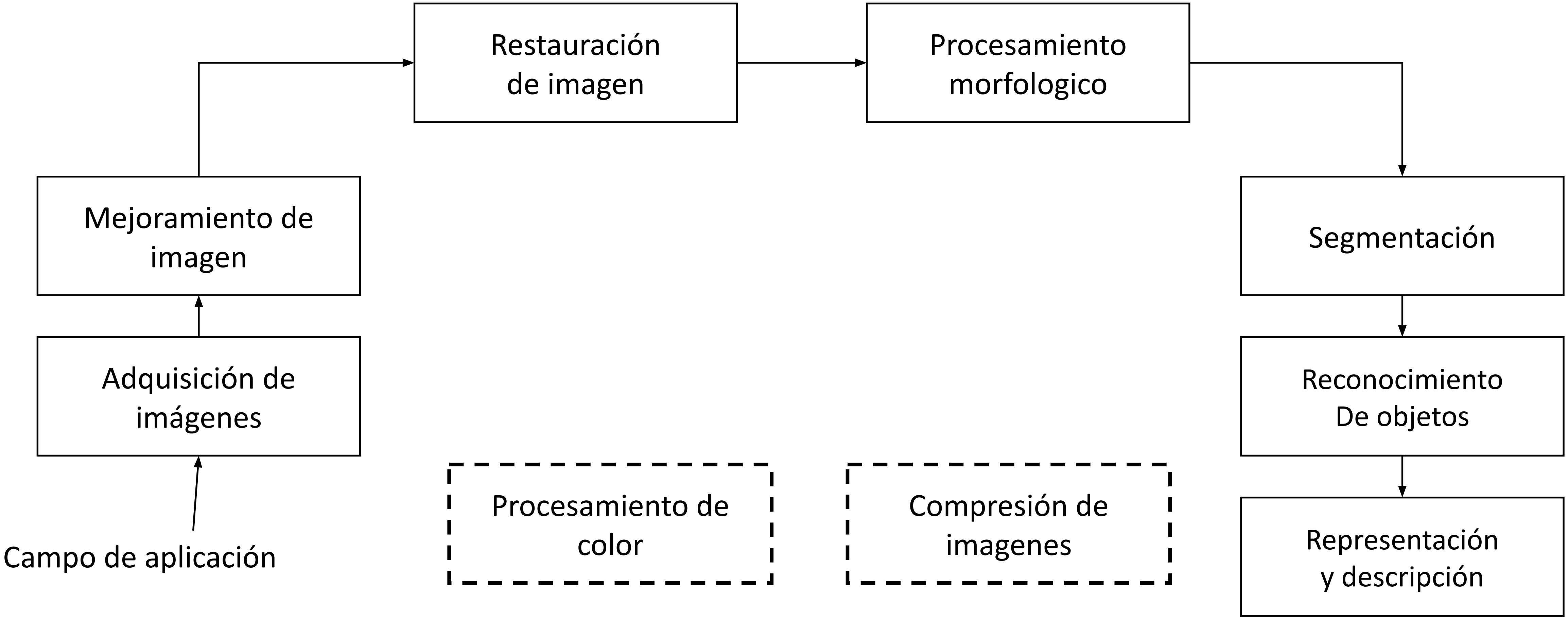
f (x, y) □ imagen de entrada
g (x, y) □ imagen de salida
T (x, y) □ operador matematico



Puede ser por punto o una convolución

Convolución y aplicaciones en procesamiento de imágenes

Representación de imágenes



Convolución y aplicaciones en **procesamiento de imágenes**

Convolución y sus propiedades

A diferencia de los métodos visto antes

Mejoramiento de la imagen **usando información de la vecindad de los pixelws**

Estos **pixeles vecinos** se conocen como **filtro** (o mascara, kernel, ventana)

111	115	113	111	112	111	112	111
135	138	137	139	145	146	149	147
163	168	188	196	206	202	206	207
180	184	206	219	202	200	195	193
189	193	214	216	104	79	83	77
191	201	217	220	103	59	60	68
195	205	216	222	113	68	69	83
199	203	223	228	108	68	71	77

f[m,n]

$\otimes$

-1	2	-1
-1	2	-1
-1	2	-1

h[m,n]

=

?	?	?	?	?	?	?	?
?	-5	9	-9	21	-12	10	?
?	-29	18	24	4	-7	5	?
?	-50	40	142	-88	-34	10	?
?	-41	41	264	-175	-71	0	?
?	-24	37	349	-224	-120	-10	?
?	-23	33	360	-217	-134	-23	?
?	?	?	?	?	?	?	?

g[m,n]

Tecnológico de Monterrey

Convolución y sus propiedades

Los valores del **kernel** son denominados **coeficientes**

Operación de convolución □

Modificar los pixeles en una imagen usando **una función de los pixeles** de su vecindad, usando estos **kernels de coeficientes**

La mas simple: **filtrado lineal** (remplazar cada pixel usando una combinación lineal de sus vecinos

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

-1	0	1
-1	0	1
-1	0	1

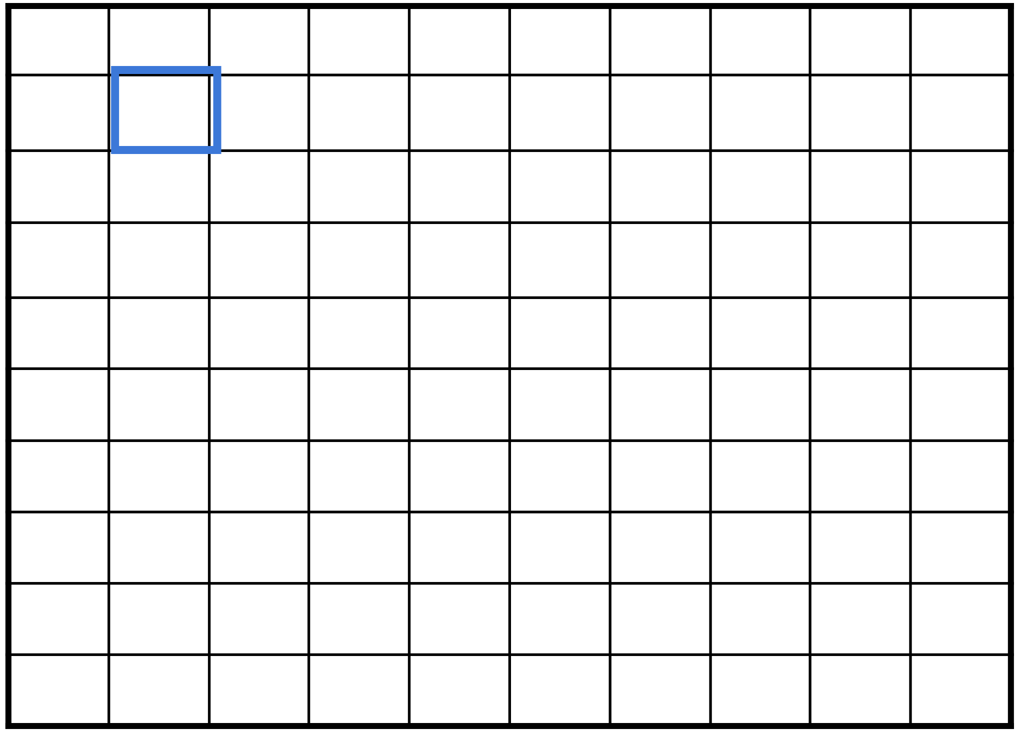
1	0	-1
2	0	-2
1	0	-1

Convolución y aplicaciones en procesamiento de imágenes

Convolución y sus propiedades

$$h[m,n] = \sum_{k,l} f[k,l]\, I[m+k, n+l]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



Credit: S. Seitz

Convolución y aplicaciones en procesamiento de imágenes

Convolución y sus propiedades

$$h[m,n] = \sum_{k,l} f[k,l]\, I[m+k, n+l]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10							

Credit: S. Seitz

Convolución y aplicaciones en procesamiento de imágenes

Convolución y sus propiedades

$$h[m,n]=\sum_{k,l} f[k,l]\, I[m+k,n+l]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10							

Convolución y aplicaciones en procesamiento de imágenes

Convolución y sus propiedades

$$h[m,n] = \sum_{k,l} f[k,l]\, I[m+k, n+l]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20						

Convolución y sus propiedades

$$h[m,n]=\sum_{k,l} f[k,l]\, I[m+k,n+l]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30					

Convolución y aplicaciones en procesamiento de imágenes

Convolución y sus propiedades

$$h[m,n] = \sum_{k,l} f[k,l]\, I[m+k, n+l]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30				

Convolución y aplicaciones en procesamiento de imágenes

Convolución y sus propiedades

$$h[m,n] = \sum_{k,l} f[k,l]\, I[m+k, n+l]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Convolución y aplicaciones en procesamiento de imágenes

Convolución y sus propiedades

Los “kernels” computan una función de los pixeles aledaños

$$h[m,n] = \sum_{k,l} f[k,l]\, I[m+k, n+l]$$

Muy importantes en pre-procesamiento de imagenes

- Mejorar o realzar propiedades
  - Ruido, cambios de tamaño, contraste, etc
- Extracción de información
  - Textura, líneas, puntos distintivos, etc.
- Detectar patrones
  - Encontrar correspondencias entre “templates”

“Gradient Filter”

-1	0	1
-1	0	1
-1	0	1

“Sobel Filter”

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

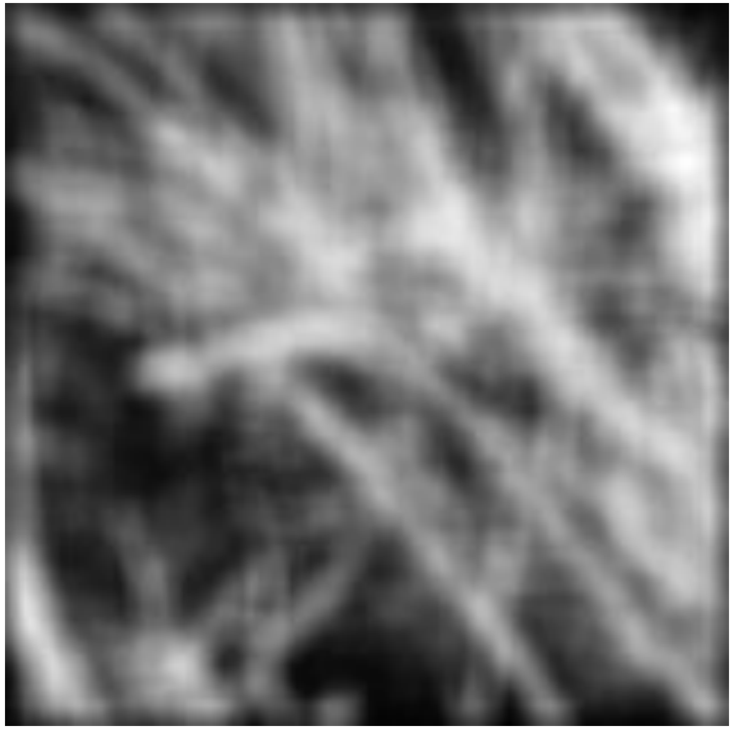
Convolución y aplicaciones en **procesamiento de imágenes**

Convolución y sus propiedades

- El “kernel” remplaza cada pixel con la media de sus vecinos
- Logra un efecto de suavizado (filtro pasabajos)
- Tipo de procesamiento ampliamente usado en visión computacional

$$f[\cdot\,,\cdot\,] \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

“Box filter”



Tecnológico de Monterrey

La **convolución** requiere de **pixeles que no existen en los bordes**

¿Que hacer? → hacer uso de técnicas de **“border padding”**

Existen **diferentes variantes**, el padding depende del tamaño del filtro

zero

fixed/clamp

periodic/wrap

reflected/mirror

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

filtros separables: ventajas

Convolución 2D
(localización central)

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix} \quad \begin{aligned} &= 2 + 6 + 3 = 11 \\ &= 6 + 20 + 10 = 36 \\ &= 4 + 8 + 6 = 18 \\ & \overline{\quad\quad\quad 65} \end{aligned}$$

El filtro puede ser **factorizado** como el producto de dos filtros 1D

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Convolución sobre las filas

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix} = \begin{bmatrix} & 11 & \\ & 18 & \\ & 18 & \end{bmatrix}$$

Seguido por Convolución
Sobre la columna restante:

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} & 11 & \\ & 18 & \\ & 18 & \end{bmatrix} = \begin{bmatrix} & & \\ & 65 & \\ & & \end{bmatrix}$$

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

filtros separables: ventajas

¿Como es la separabilidad útil en la practica?

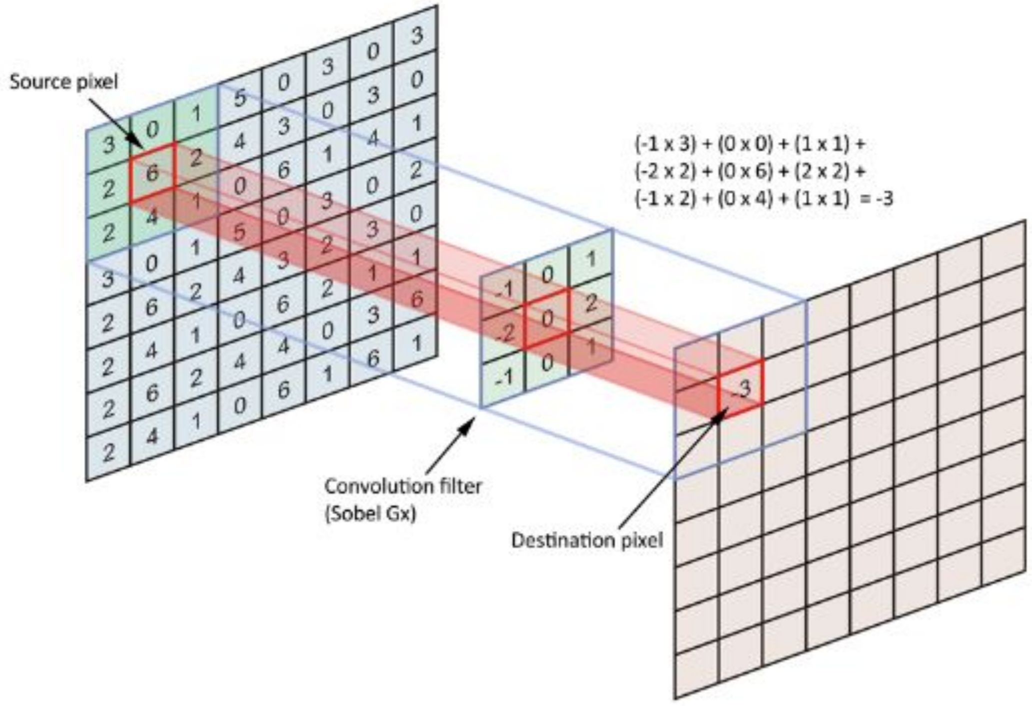
Para una imagen MxN y filtro de tamaño PxQ

- **Convolución en 2D :**

~MNPQ multiply-adds

- **Separable en 2D:**

~MN(P+Q) multiply-adds



Speed up = PQ/(P+Q)
Un filtro 9x9 = ~4.5x mas rapido

Convolución y aplicaciones en **procesamiento de imágenes**

Filtrado de imágenes

Comúnmente, una imagen está compuesta de

- Algún tipo de **estructura subyacente** ideal
- La cual deseamos **detectar y describir**
- Además de **ruido aleatorio** o artefactos, los cuales deseamos **remover**

Veremos algunos **métodos de filtrado especial** para la remoción de ruido, así como otros que nos ayudan a detectar estructuras como líneas

En primer lugar, los **filtros de suavizado** son usados para disminuir ciertos tipos de variaciones en la imagen, así como ruido

Los filtros de suavizado son conocidos también como **filtros de promediado**

Convolución y aplicaciones en procesamiento de imágenes

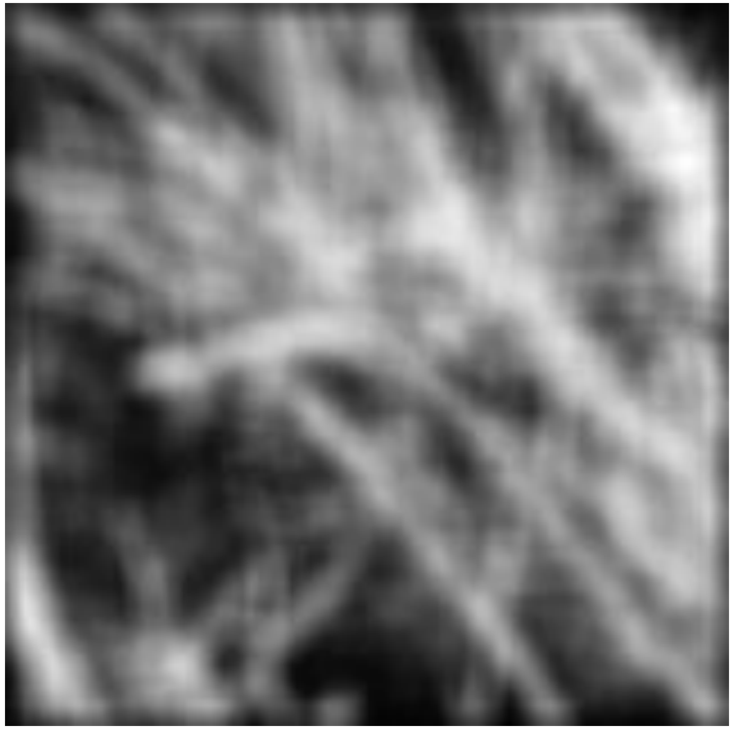
Filtrado de imágenes

Suavizado

- El “kernel” remplaza cada pixel con la media de sus vecinos
- Logra un efecto de suavizado (filtro pasabajos)
- Tipo de procesamiento ampliamente usado en visión computacional

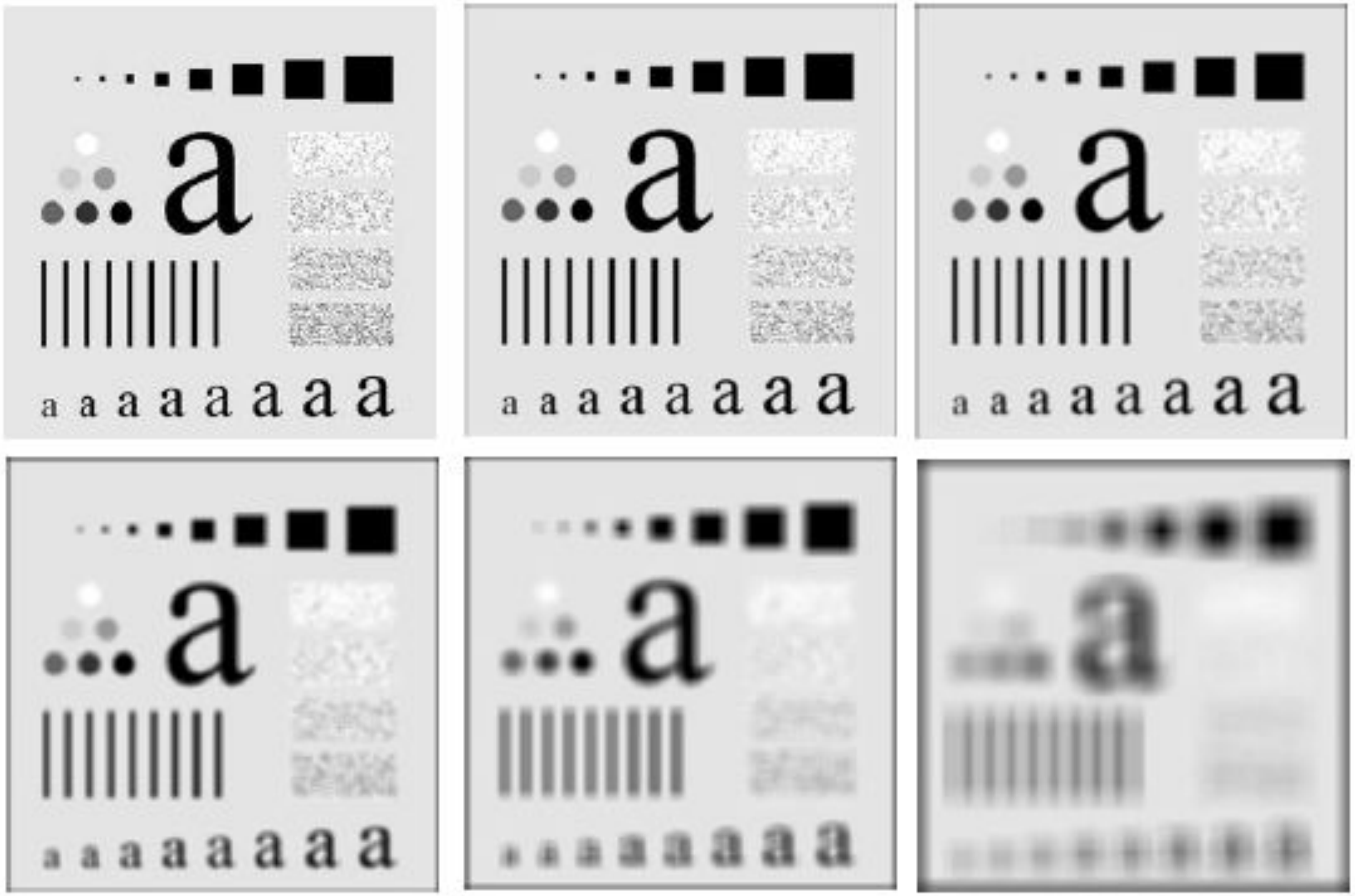
$$f[\cdot\,,\cdot\,] \quad \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

“Box filter”



Filtrado de imágenes

Suavizado



Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Suavizado

I

10	11	10	0	0	1
9	10	11	1	0	1
10	9	10	0	2	1
11	10	9	10	9	11
9	10	11	9	99	11
10	9	9	11	10	10

F

1/9

1	1	1
1	1	1
1	1	1

O

X	X	X	X	X	X
X	10				X
X					X
X					X
X					X
X	X	X	X	X	X

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Suavizado

I

10	11	10	0	0	1
9	10	11	1	0	1
10	9	10	0	2	1
11	10	9	10	9	11
9	10	11	9	99	11
10	9	9	11	10	10

F

1/9

1	1	1
1	1	1
1	1	1

O

X	X	X	X	X	X
X					X
X					X
X					X
X				20	X
X	X	X	X	X	X

Convolución y aplicaciones en procesamiento de imágenes

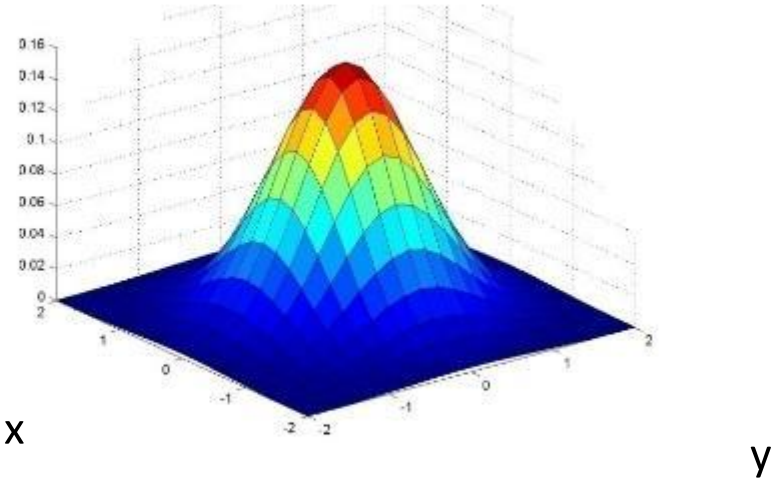
Filtrado de imágenes

Suavizado

El box filter introduce artefactos ("box")

Filtro Gaussiano pesa las contribuciones de los pixeles aledaños por cercanía

Filtro: Gaussian



$$
\begin{array}{c|ccccc}
 & & & x & & \\
\hline
 & 0.003 & 0.013 & 0.022 & 0.013 & 0.003 \\
 & 0.013 & 0.059 & 0.097 & 0.059 & 0.013 \\
y & 0.022 & 0.097 & 0.159 & 0.097 & 0.022 \\
 & 0.013 & 0.059 & 0.097 & 0.059 & 0.013 \\
 & 0.003 & 0.013 & 0.022 & 0.013 & 0.003
\end{array}
$$

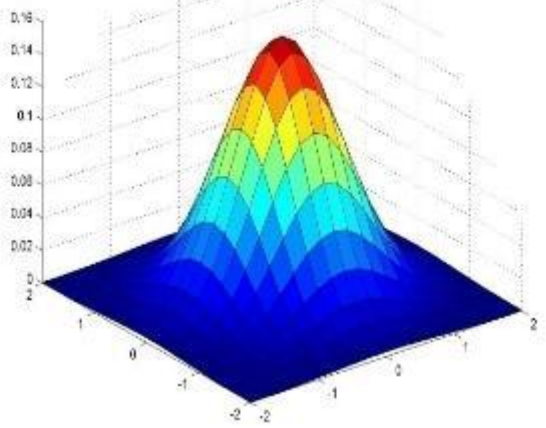
5 x 5, σ = 1

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

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Suavizado



- Remueve componentes de "alta-frecuencia" de la imagen (filtro pasa-bandas)

  Las imágenes se suavizan

- Gaussiana convolucionada con otra …es otra Gaussiana

  mejora procesamiento

- *Kernel separable*
  - La convolución puede separarse en dos componentes

    menos operaciones

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

Suavizado

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

$$= \left(\frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{x^2}{2\sigma^2}}\right) \left(\frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{y^2}{2\sigma^2}}\right)$$

- **Filtro Gaussiano** *Kernel separable*
  - La función puede expresarse como el producto de dos funciones independientes, una para x y otra para y
  - En este caso las dos gaussianas idénticas en 1D

Filtrado de imágenes

Suavizado

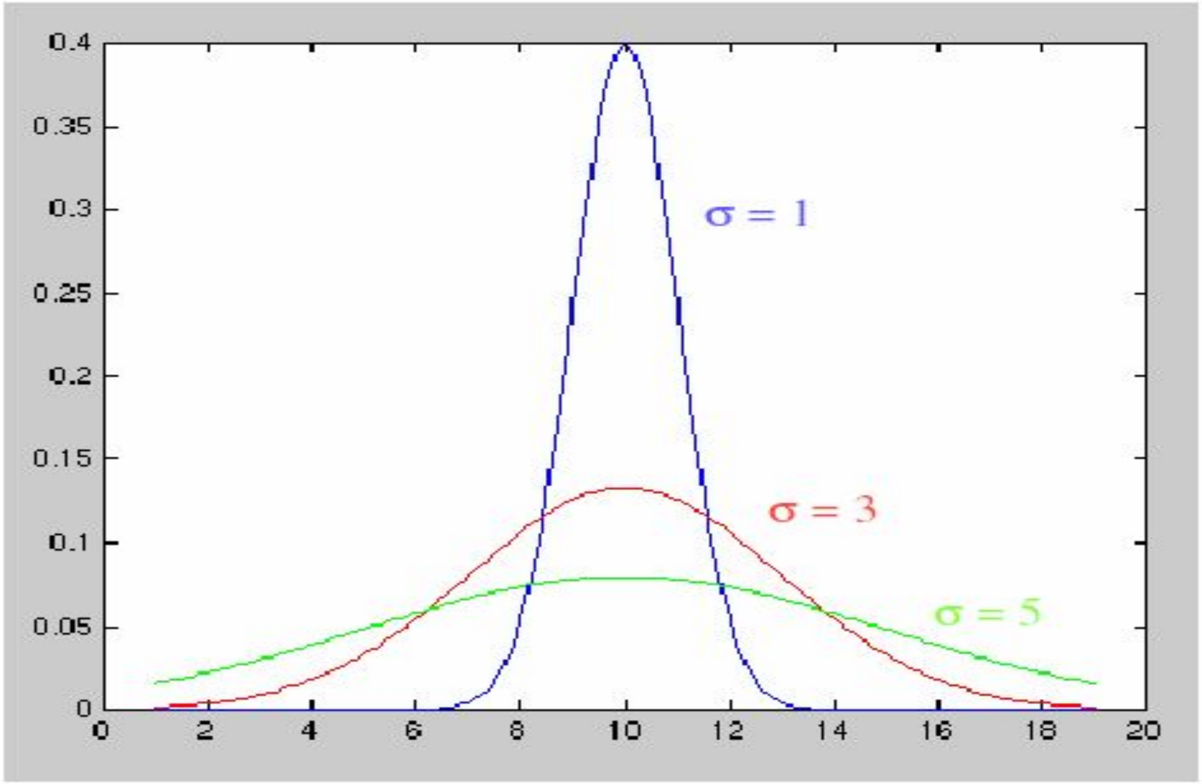
Efectos del tamaño de σ

Si σ es pequeña: el suavizado tendrá poco efecto sobre pixeles adyacentes

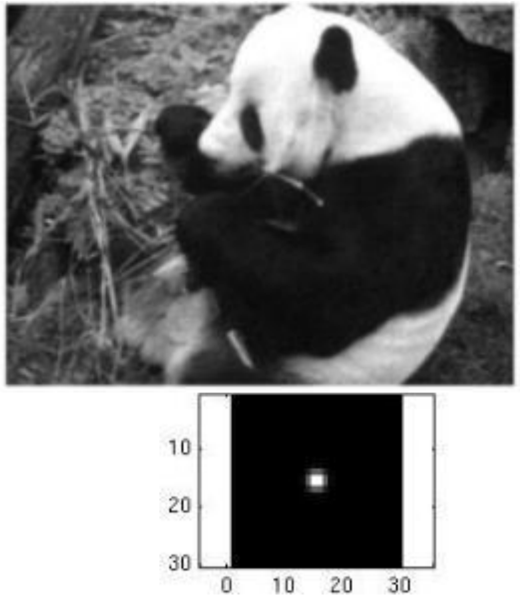
Si σ es grande: los pixeles adyacentes tendrán pesos mas grandes , resultando en un consenso con los vecinos en la ventana

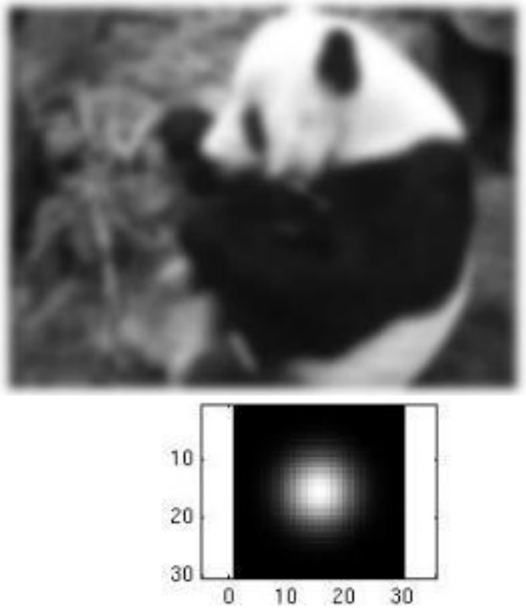
Si σ es muy grande: los detalles de a imagen desparecen junto al ruido

Effect of σ

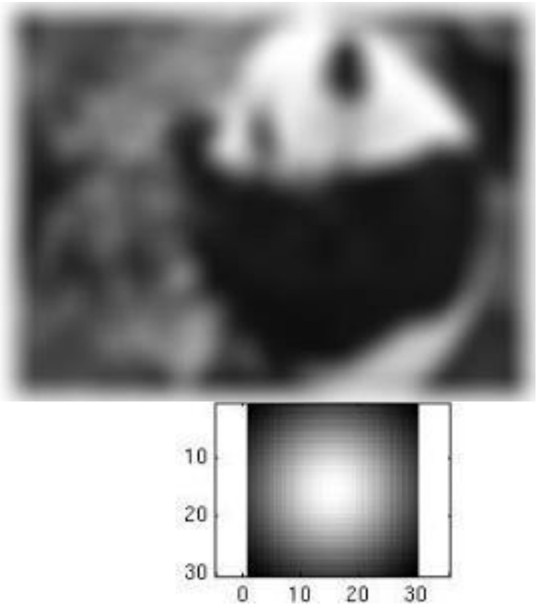


Suavizado





. . .

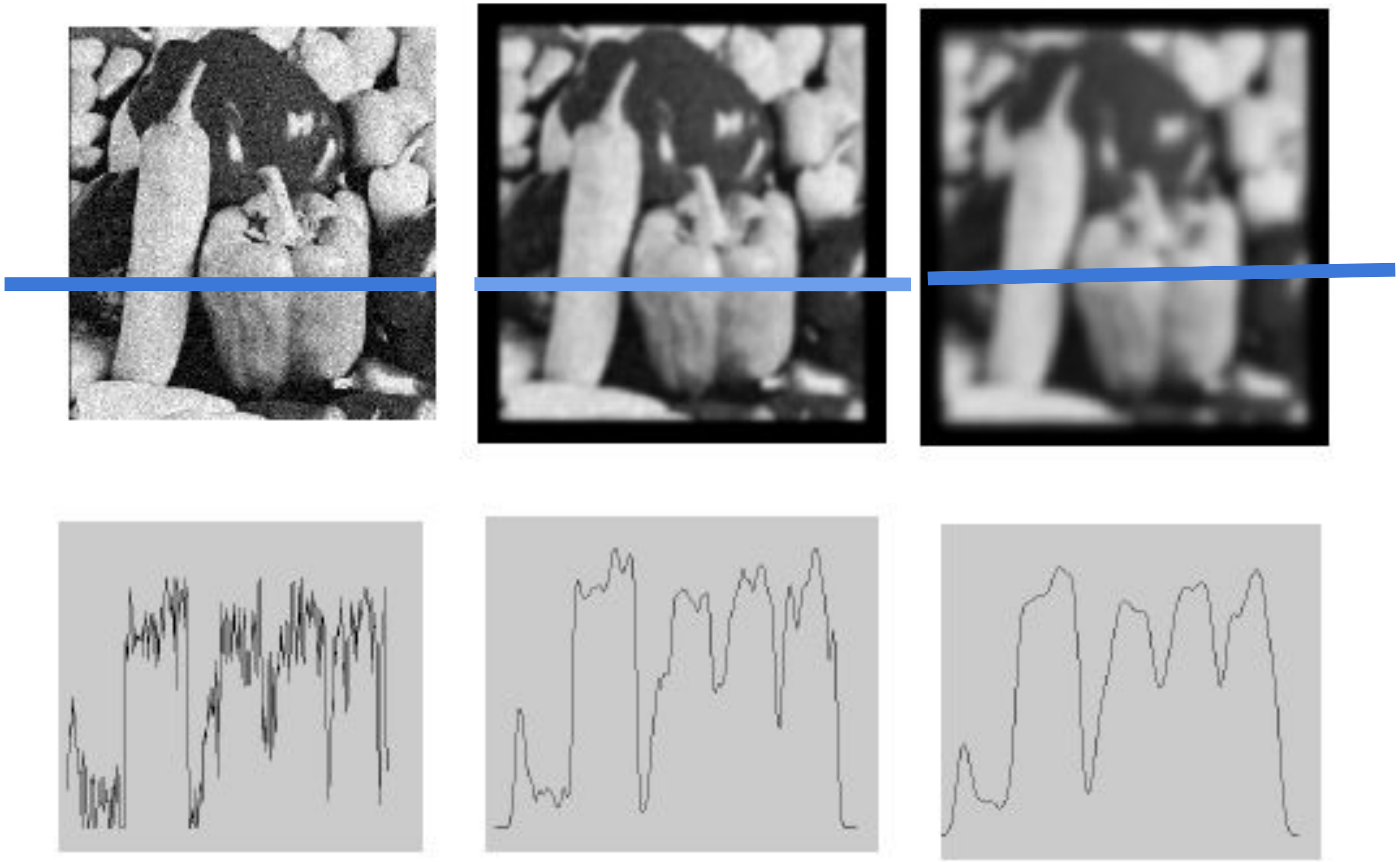


Ejemplo de como el **ancho** de la Gaussiana **controla el suavizado**

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

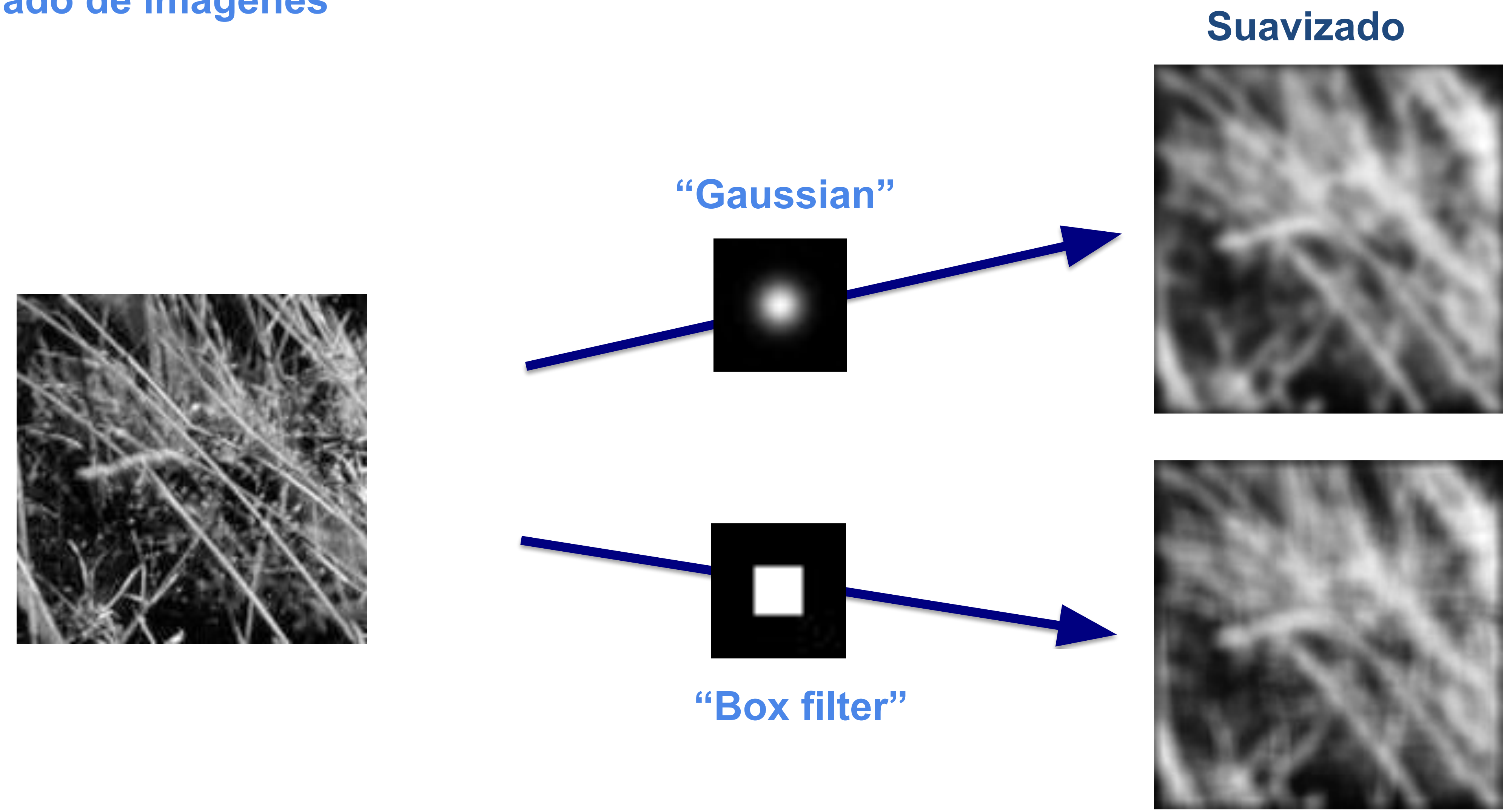
Suavizado



No smoothing $\sigma = 2$ $\sigma = 4$

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes



Convolución y aplicaciones en **procesamiento de imágenes**

Filtrado de imágenes

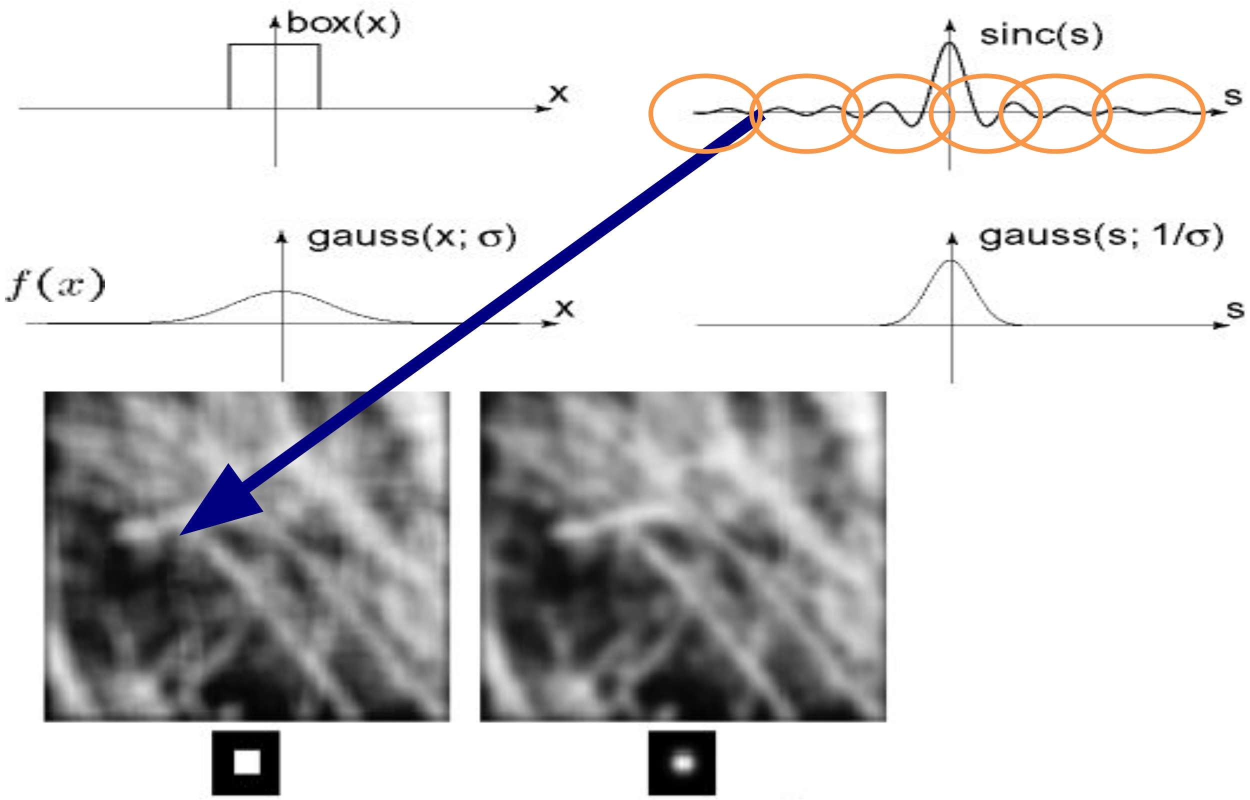
Suavizado

Dominio Espacial | Dominio Frecuencia

Resultado de usar
un **filtro tipo “box”**

Produce un set de **barras** H y V

Fenomento conocido como **“ringing effect”**



Resultado de suavizado usando un filtro **Gaussiano**

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Suavizado

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

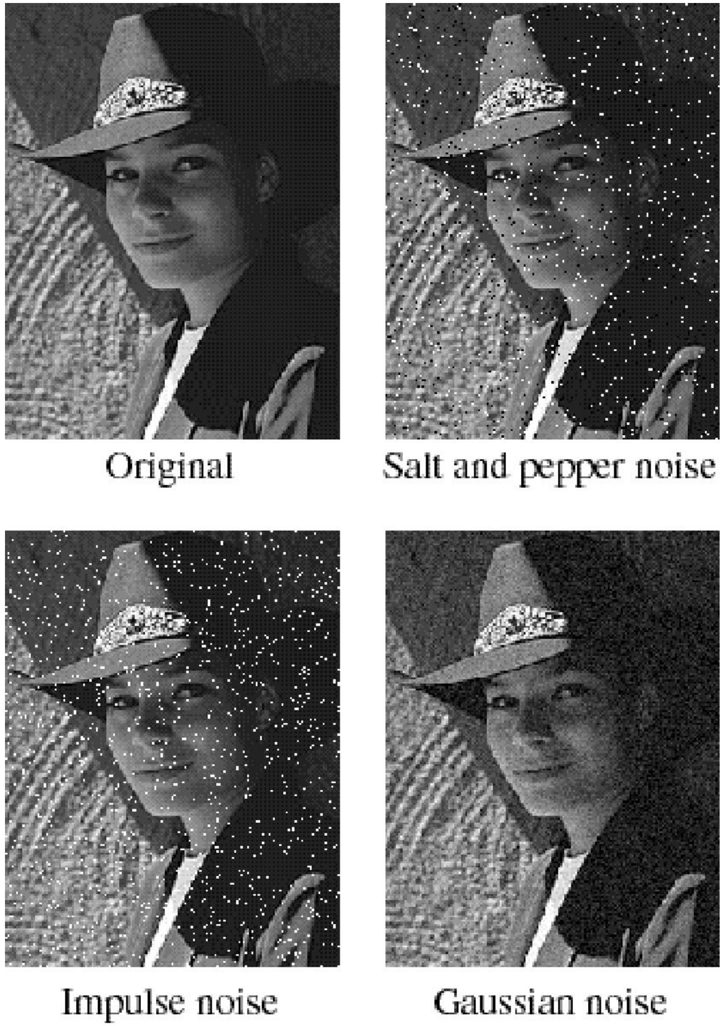
Remoción de ruido

Tipos comunes de ruido

Salt-and-pepper noise: contiene ocurrencias aleatorias de pixeles blancos y negros

Ruido impulsivo: contiene o esta caracterizado por ocurrencias aleatorias de pixeles blancos

Ruido Gaussiano: variaciones en intensidad debidas a una distribución Gaussiana normal (IID)



Remoción de ruido

Los **filtros de orden estadísticos** son filtros espaciales no lineales cuya respuesta esta dada por el **ordenamiento de los pixeles** contenidos en el kernel

El filtro aplica un **ordenamiento** de los pixeles y selecciona el **valor de la mediana**

Y es **reemplazado** en la imagen de salida en el la **posición central del kernel**

El **kernel** mas conocido de este tipo es el **median filter**

Este es particularmente efectivo en la **presencia de ruido** de tipo “impulse” o salt-and-pepper noise, **con menos efectos de suavizado de otros fitros**

Remoción de ruido

I

10	11	10	0	0	1
9	10	11	1	0	1
10	9	10	0	2	1
11	10	9	10	9	11
9	10	11	9	99	11
10	9	9	11	10	10

O

X	X	X	X	X	X
X	10				X
X					X
X					X
X					X
X	X	X	X	X	X

10,11,10,9,10,11,10,9,10

ordenar

9,9,10,10,10,10,10,11,11

mediana

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Remoción de ruido

I

10	11	10	0	0	1
9	10	11	1	0	1
10	9	10	0	2	1
11	10	9	10	9	11
9	10	11	9	99	11
10	9	9	11	10	10

O

X	X	X	X	X	X
X					X
X					X
X					X
X				10	X
X	X	X	X	X	X

10,9,11,9,99,11,11,10,10

ordenar

9,9,10,10,10,11,11,11,99

mediana

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Remoción de ruido

Filtro →

Mean **Gaussian** **Median**

Ventana 3 x 3

Ventana 7 x 7

Resultado 1

Resultado 2

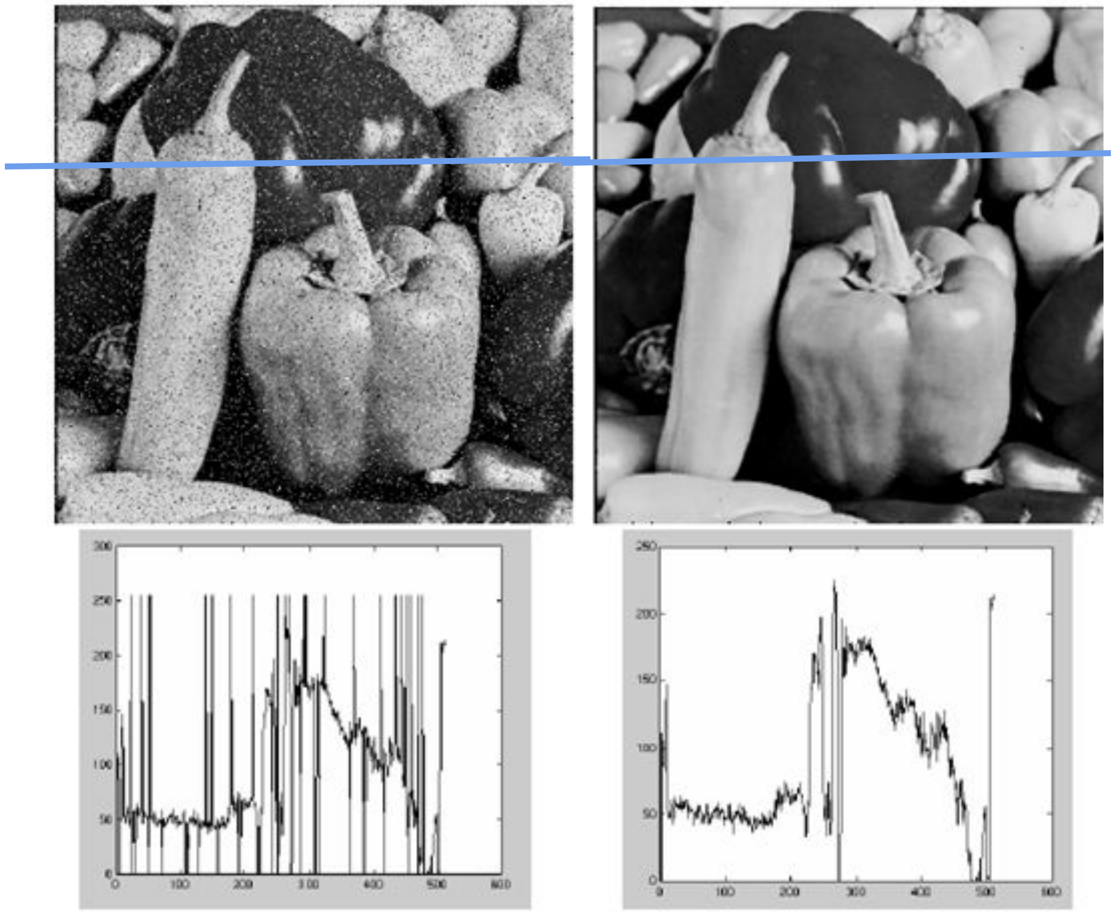
Tecnológico de Monterrey

Convolución y aplicaciones en **procesamiento de imágenes**

Filtrado de imágenes

Remoción de ruido

Efecto del filtro de mediana
en **ruido salt-and peppper**



Reduce fluctuaciones en la señal □ se usa como **filtro de pre-procesamiento**

Tecnológico de Monterrey

Mejoramiento de detalles

El objetivo es de este filtro es **resaltar o mejorar detalles finos** de una imagen

Dado que suavizado (averaging) es análoga a la integración, el **"sharpening"** puede ser realizado como una **diferenciación espacial**

Derivada de **primer orden** de una función f(x) 1D

$f(x+1) - f(x)$.

Derivada de **segundo orden** de una función f(x) 1D

$f(x+1) - 2f(x) + f(x-1)$.

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Detección de líneas

Discontinuidad de profundidad

Discontinuidad o cambios de orientación de superficies

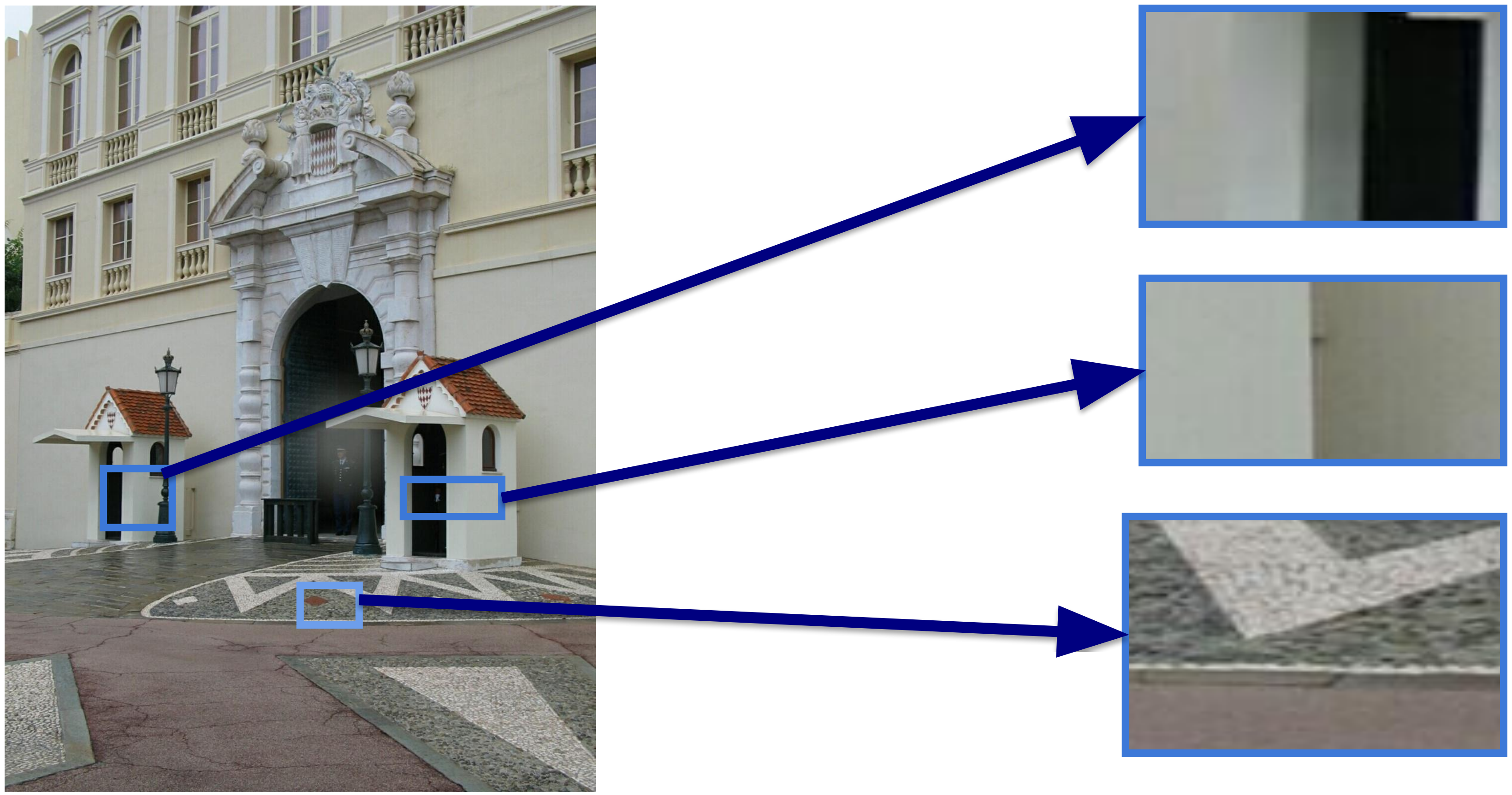
Discontinuidad en reflectancia (i.e., cambios las propiedad de los materiales de la superficie)

Discontinuidad en iluminación (e.g., brillos, sombras)

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Detección de líneas



Discontinuidad
en la normal
de la superficie

Discontinuidad
en profundidad

Discontinuidad
En color o textura

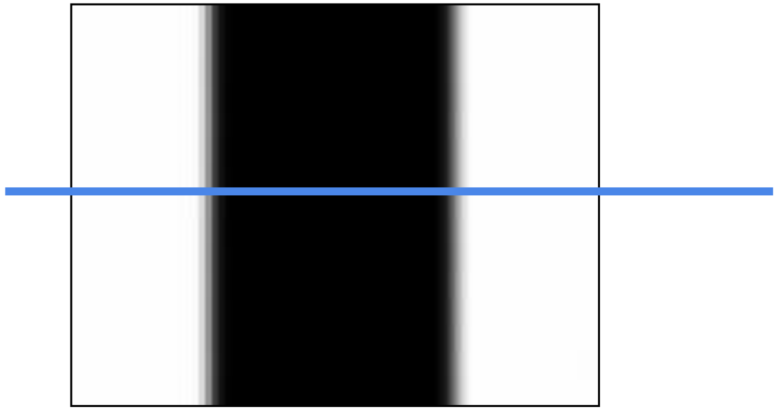
Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

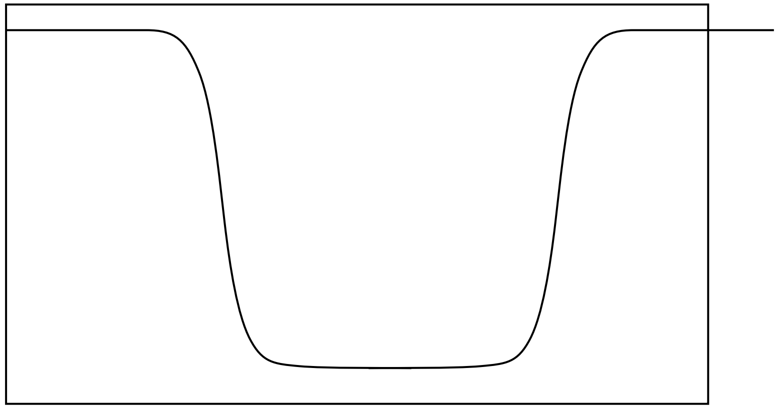
Detección de líneas

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x) = f_x$$

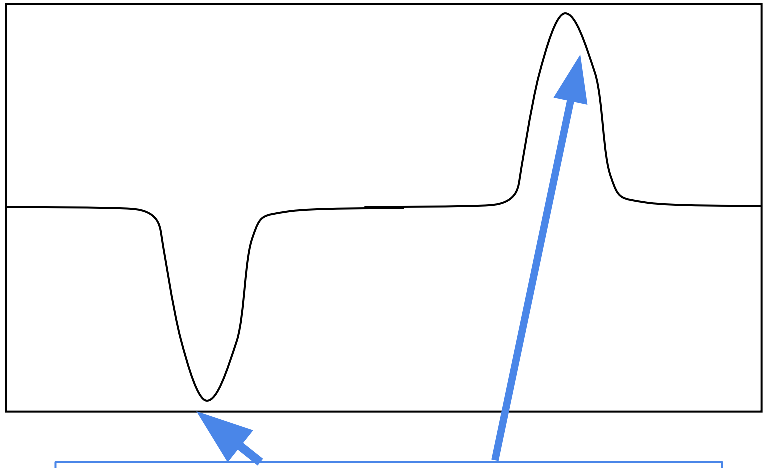
imagen



intensidad



Primera derivada



Edge □ extrema de f’

Tecnológico de Monterrey

Filtrado de imágenes

Detección de líneas

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x)$$

$$\frac{df}{dx} = \frac{f(x) - f(x-1)}{1} = f'(x)$$

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Derivada discreta en 1D

Detección de líneas

Backward

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Forward

$$\frac{df}{dx} = f(x) - f(x+1) = f'(x)$$

Central

$$\frac{df}{dx} = f(x+1) - f(x-1) = f'(x)$$

Tipos de derivadas discretas en 1D

Filtrado de imágenes

Detección de líneas

Backward filter: $f(x) - f(x-1) = f'(x)$ [0 1 -1]

Forward: $f(x) - f(x+1) = f'(x)$ [-1 1 0]

Central: $f(x+1) - f(x-1) = f'(x)$ [1 0 -1]

Tipos de derivadas discretas en 1D

Filtrado de imágenes

Detección de líneas

Derivada discreta en 2D

Función dada → $f(x,y)$

Vector de gradientes → $\nabla f(x,y) = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$

Magnitud del gradiente → $|\nabla f(x,y)| = \sqrt{f_x^2 + f_y^2}$

Dirección del gradiente → $\theta = \tan^{-1} \dfrac{f_x}{f_y}$

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Detección de líneas

Robert's cross-gradient operators

-1	0	
0	1	

0	-1	
1	0	

Prewitt Gradient Operators

+1	0	-1
+1	0	-1
+1	0	-1

+1	+1	+1
0	0	0
-1	-1	-1

Sobel Gradient Operators

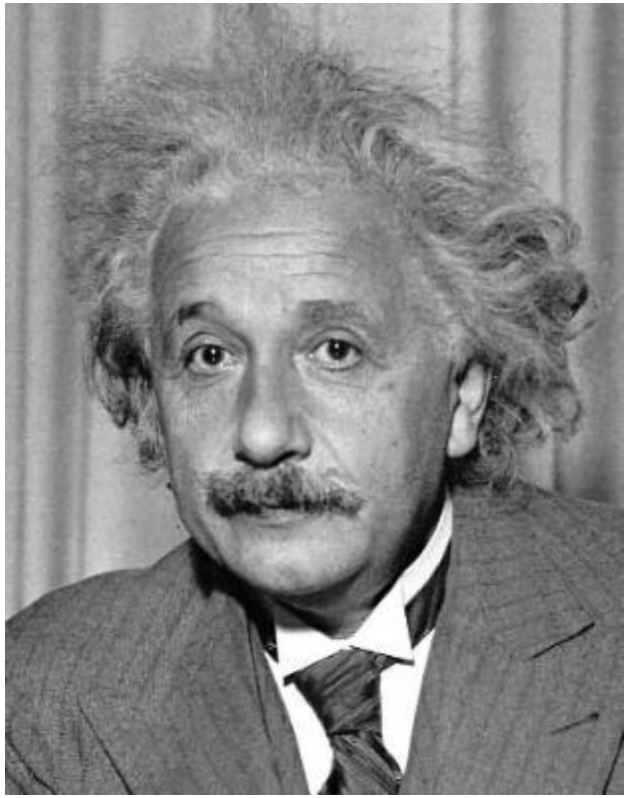
-1	0	1
-2	0	2
-1	0	1

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

Convolución y aplicaciones en procesamiento de imágenes

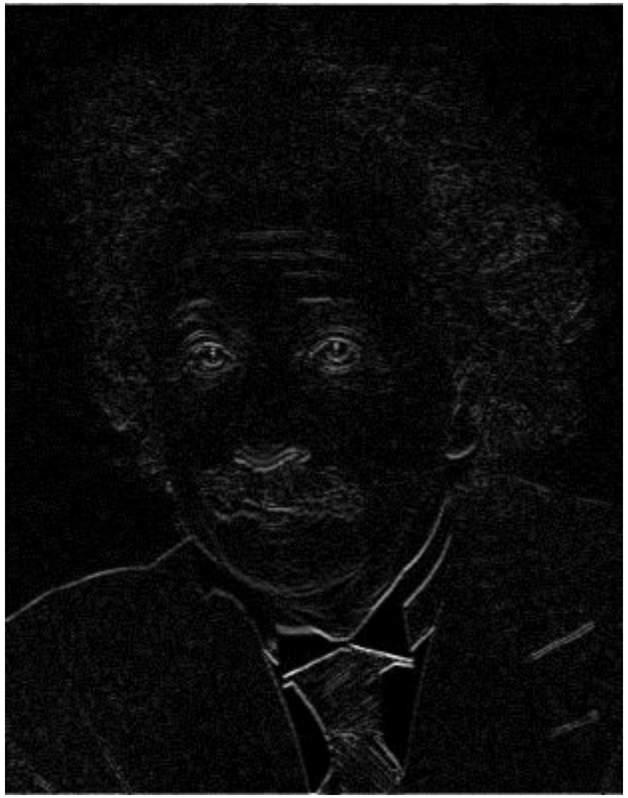
Filtrado de imágenes

Detección de líneas



Sobel

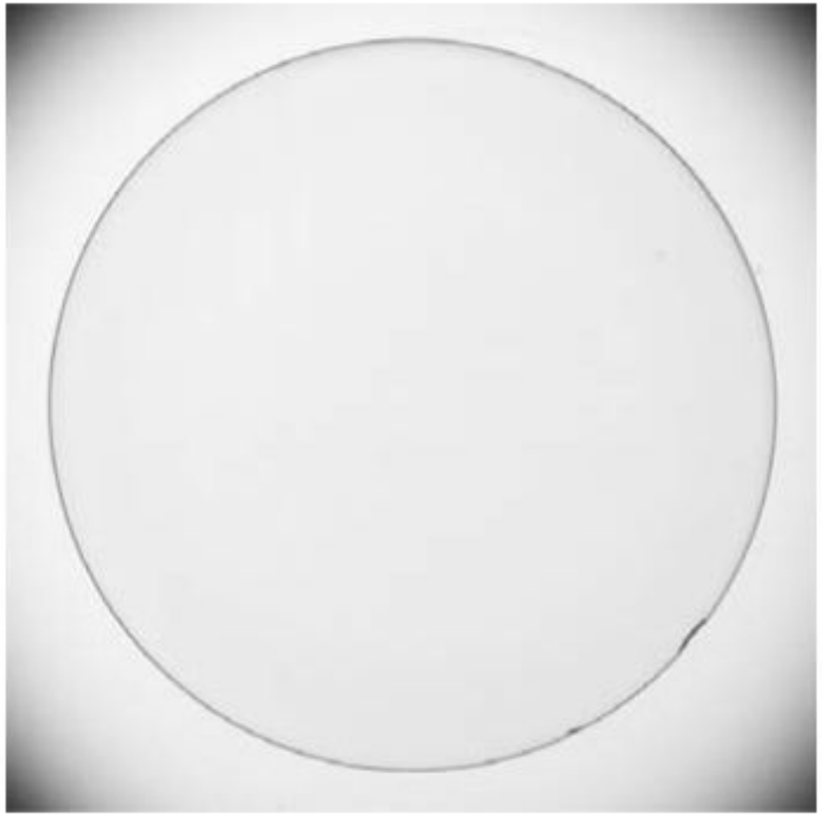
1	2	1
0	0	0
-1	-2	-1

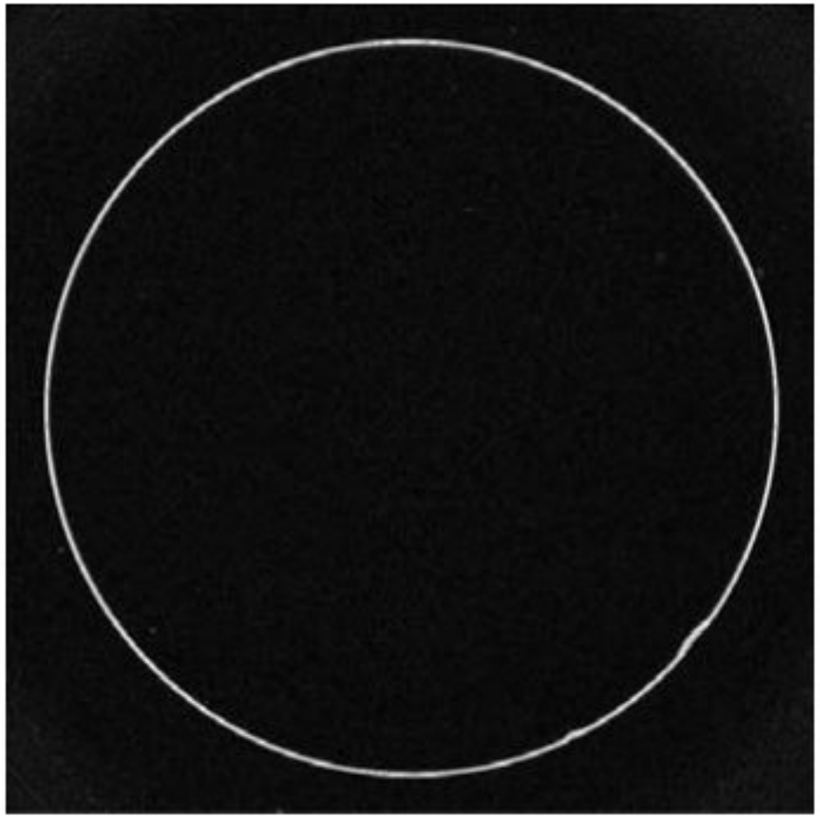


Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Detección de líneas





Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Detección de líneas

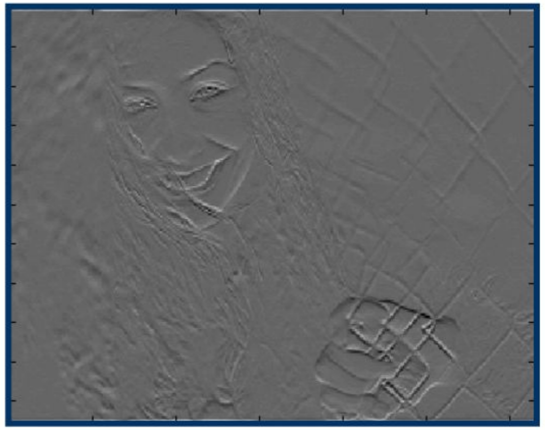
Derivada discreta en 2D

Lineas verticales

$$\frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Lineas horizontales

$$\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$



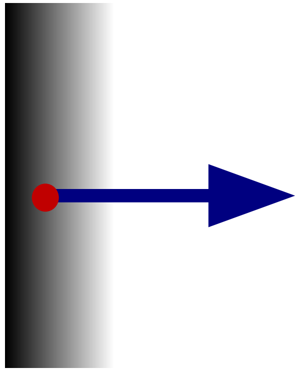
Convolución y aplicaciones en **procesamiento de imágenes**

Filtrado de imágenes

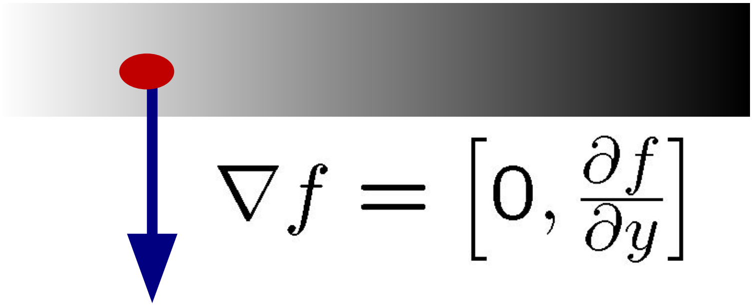
Detección de líneas

El **gradiente** de una imagen

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$



$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$



$$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$$

El **vector de gradiente** apunta en la dirección de **mayor incremento** en la intensidad de la función de entrada

La **"fuerza de la línea"** esta dada por la **magnitud del gradiente**

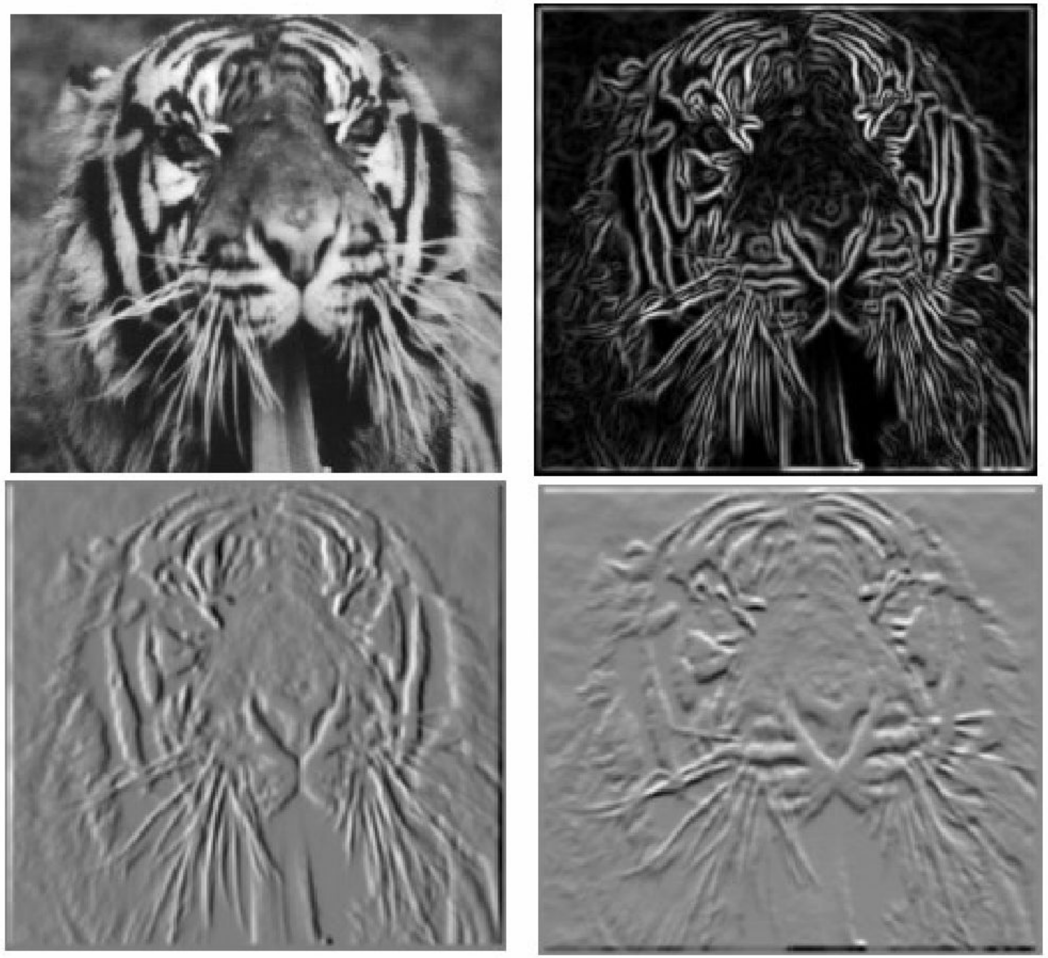
Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Detección de líneas

Imagen original

Magnitud gradiente



Gradiente dirección x

Gradiente dirección y

Detección de líneas

El **Laplaciano** de una imagen → $$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

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

2da derivada w.r.t. x

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

2da derivada w.r.t. y

Por lo que el **Laplaciano Discreto** para 2D es

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

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Sharpening

0	1	0
1	-4	1
0	1	0

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

Laplaciano en direcciones X -Y

Esta ecuación se implementa en los kernels de la izquierda, el cual es isotrópico en 90°

1	1	1
1	-8	1
1	1	1

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

Laplaciano en direcciones X -Y

El mismo laplaciano puedo usarse para considera direcciones diagonales, lo que hace que el centro sea 8 (4 x (-2 f(x,y))

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Sharpening

Filtro de sharpening (afinamiento o enfocar)

Acentúa las diferencias con la media local

Original

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

Filtrada

Filtrado de imágenes

Sharpening

1. Suavizar la imagen original

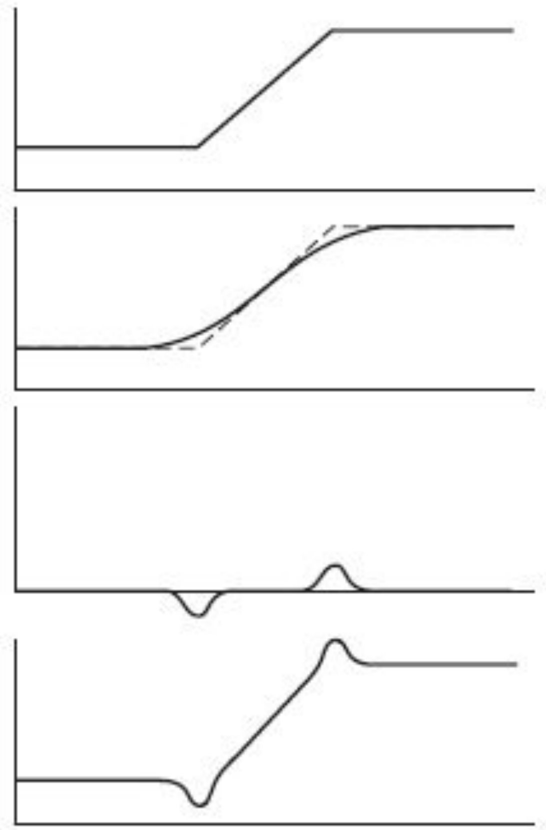
2. Substraer esta de la imagen original

$$g_{\text{mask}}(x, y) = f(x, y) - \overline{f}(x, y)$$

La imagen resultada se denomina **“mask”** □

3. Agregar la imagen original con la mascara

$$g(x, y) = f(x, y) + k * g_{\text{mask}}(x, y)$$



Señal Original

Señal Suavizada

Mascara “unsharp”

Imagen mejorada

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Sharpening

−

=

+

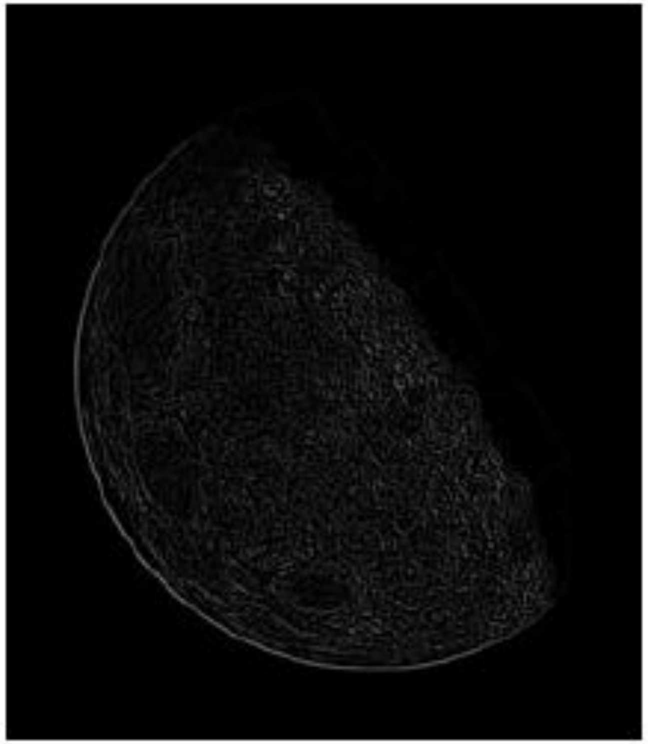
=

Convolución y aplicaciones en procesamiento de imágenes

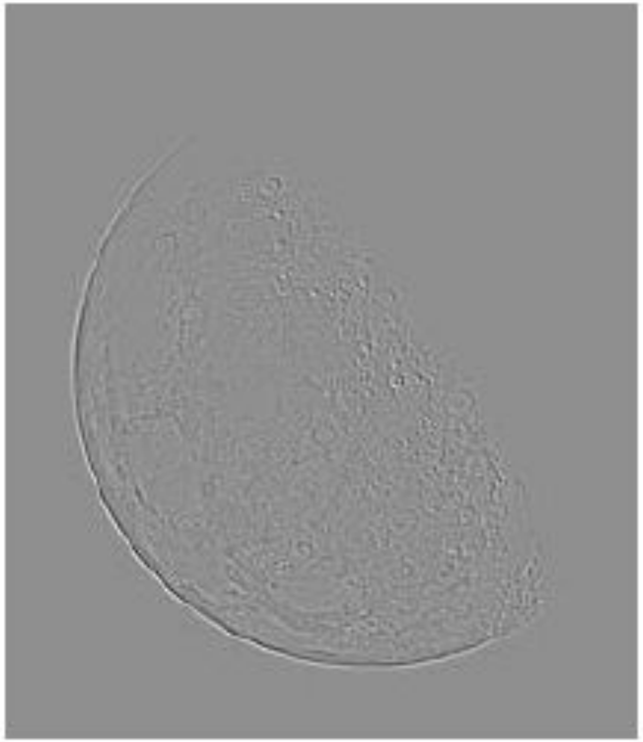
Filtrado de imágenes

Sharpening

Imagen Original



Laplaciano sin escala



Laplaciano con escala

Convolución y aplicaciones en procesamiento de imágenes

Filtrado de imágenes

Sharpening

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

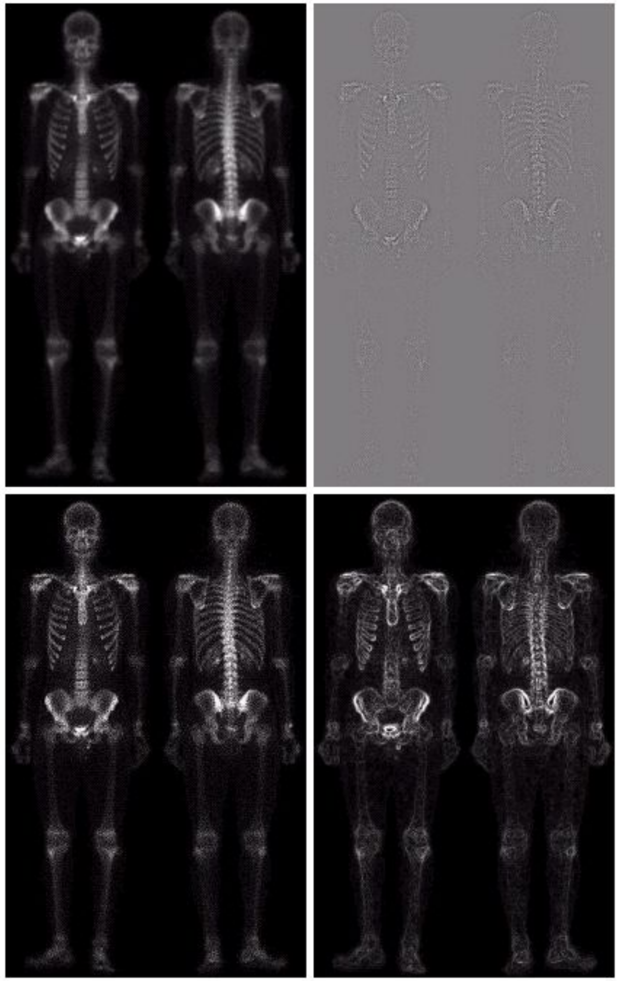
Convolución y aplicaciones en procesamiento de imágenes

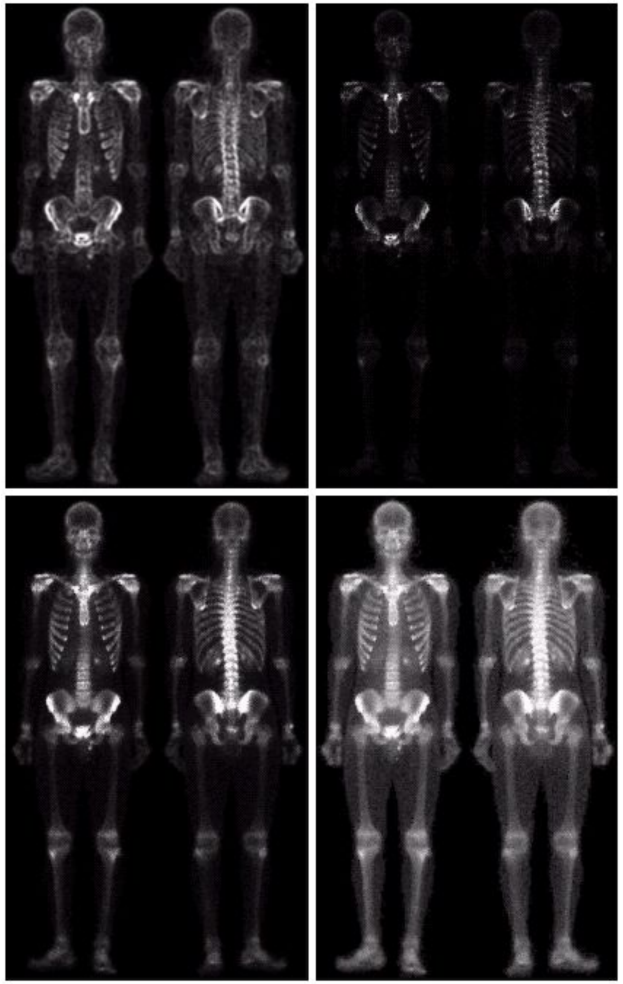
Filtrado de imágenes

Sharpening

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).





e f
g h

FIGURE 3.46
(Continued)
(e) Sobel image smoothed with a 5 × 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.)