



Extracción de Rasgos

Detección de Fronteras

Detección de Puntos de Interés. Esquinas



ugr

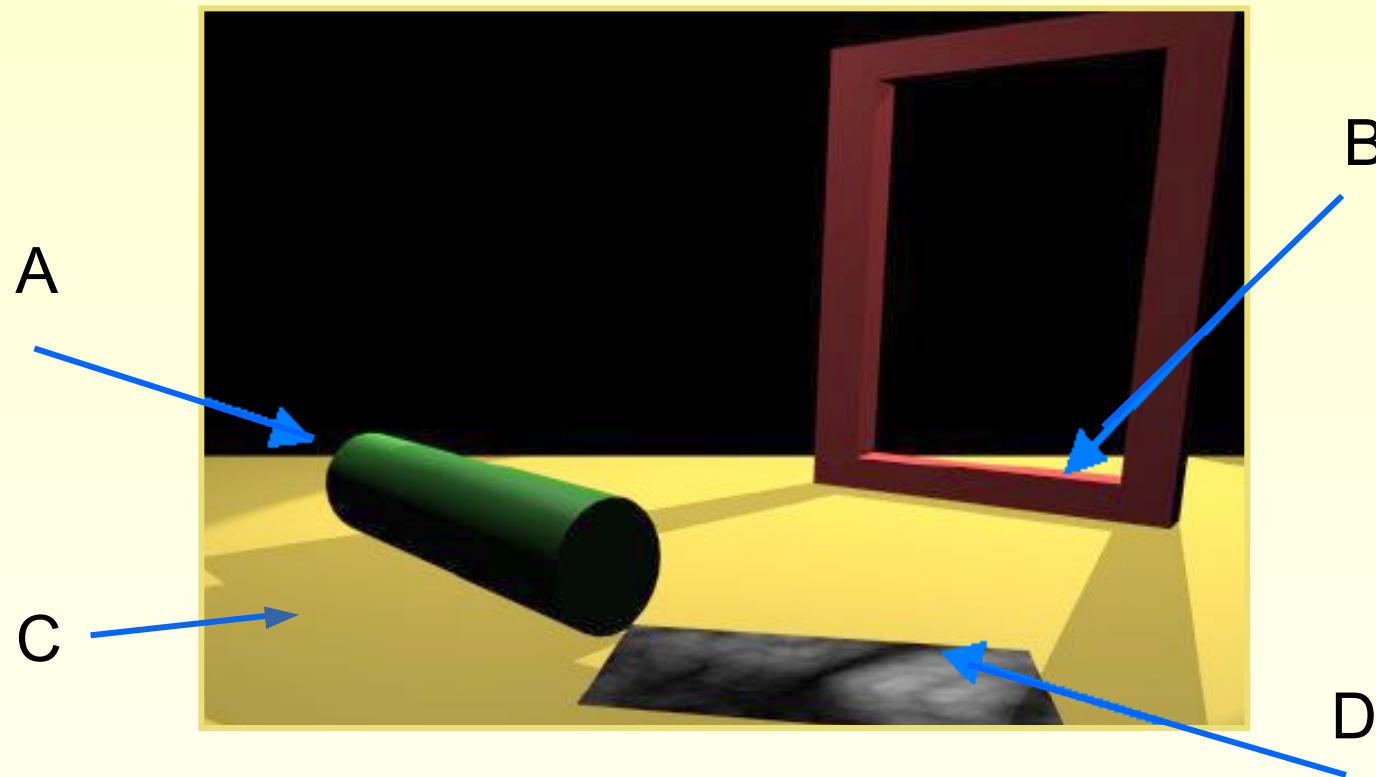
Universidad
de Granada





Extracción de Fronteras

Discontinuidad



- A: Discontinuidad de profundidad: cambio abrupto de la profundidad en el mundo
- B: Discontinuidad de superficie: cambio en la orientación de una superficie.
- C: Discontinuidad de iluminación: sombras, cambios de luz
- D: Discontinuidad de reflectancia: propiedades de la superficie, marcas.

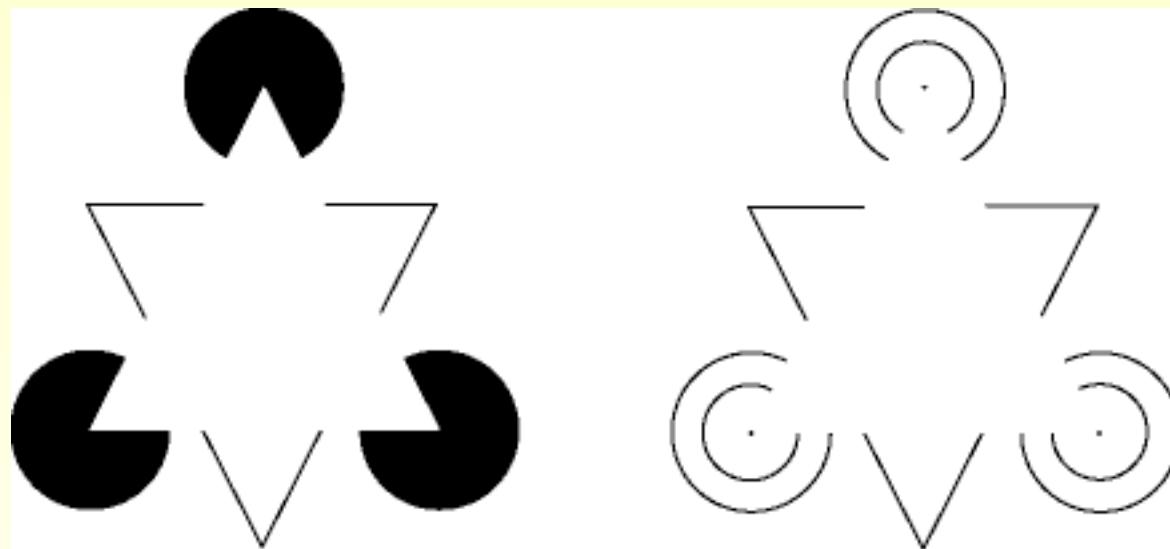


Extracción de Fronteras

Contornos Ilusorios



Triángulos
de Kanisza



- Los contornos ilusorios no serán detectables por los algoritmos que discutiremos
- Ningún cambio en la irradiancia de estas imágenes- ningún algoritmo de procesamiento de imágenes- puede directamente tratar con este tipo de situaciones.
- La Visión por Computador puede tratar con este tipo de cosas basándose en información externa a la imagen (técnicas de agrupamiento perceptual, p.ej.)



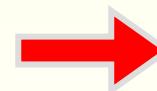
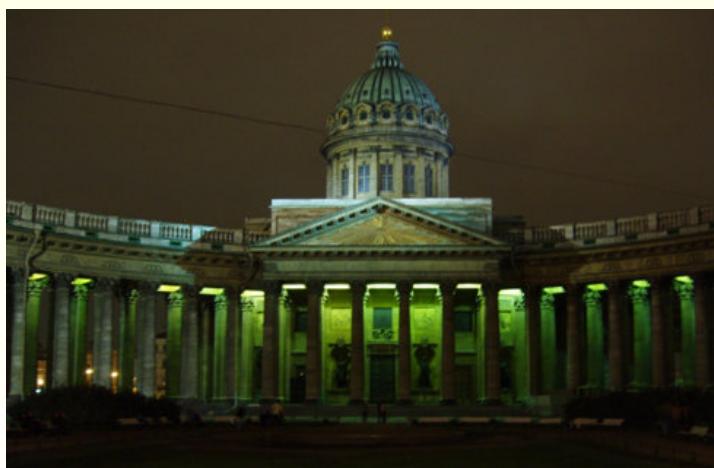
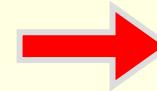
- Definimos un borde local como un cambio rápido en la función de la imagen (niveles de gris/color) sobre un área pequeña.
 - Implica que los bordes locales deberían ser detectables en un entorno local
- Los bordes locales NO son contornos, fronteras o líneas, pero son la base para la detección de las mismas, puesto que estas estructuras suelen estar formadas por bordes locales.
- Los bordes locales tienen propiedades de:
 - Orientación
 - Magnitud
 - Posición



Extracción de Fronteras Bordes Locales



- Los bordes de una escena son **invariantes a cambios** de luminosidad, color de la fuente de luz, etc. En **análisis de imágenes se** usan los bordes en muchas aplicaciones (en lugar de las originales).



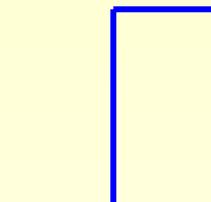


- Detectores de bordes de primer orden
 - Matemáticas
 - Roberts, Sobel, Prewitt...
- Detectores de bordes de segundo orden
 - Laplaciano, LOG / DOG
- Detector de bordes de Canny

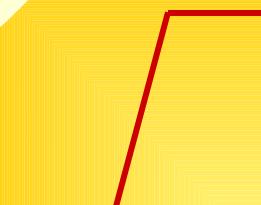


Extracción de Fronteras

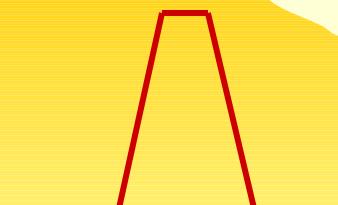
Modelos de fronteras



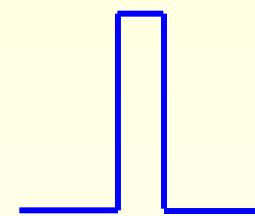
Escalón



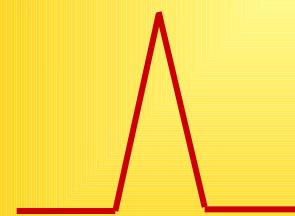
Rampa



Doble rampa

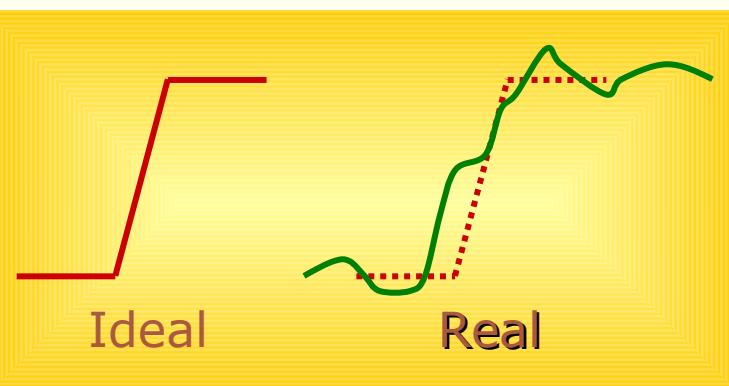


Línea



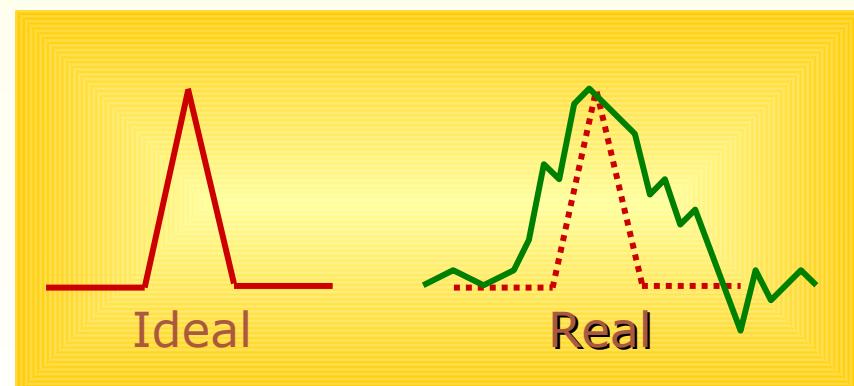
Tejado

Lo normal es que el salto no sea tan abrupto. Estas serían versiones suavizadas del escalón o la línea.



Ideal

Real



Ideal

Real



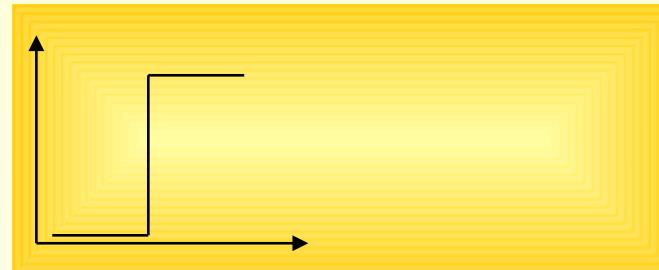
Extracción de Fronteras

Localizando bordes locales

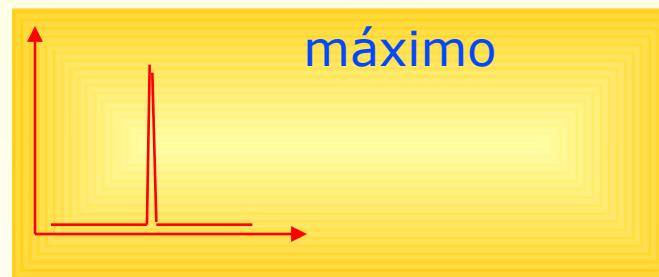


Cambios rápidos en la imagen => alto gradiente local => diferenciación

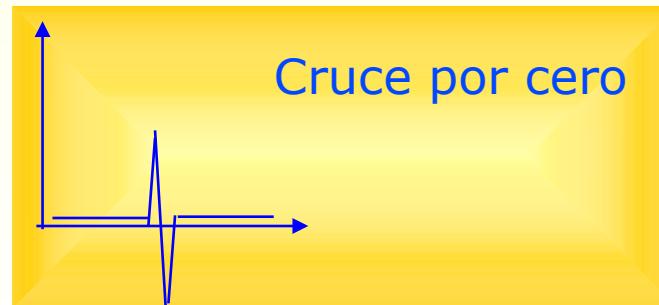
$f(x)$ = borde tipo escalón



1st Derivada $f'(x)$

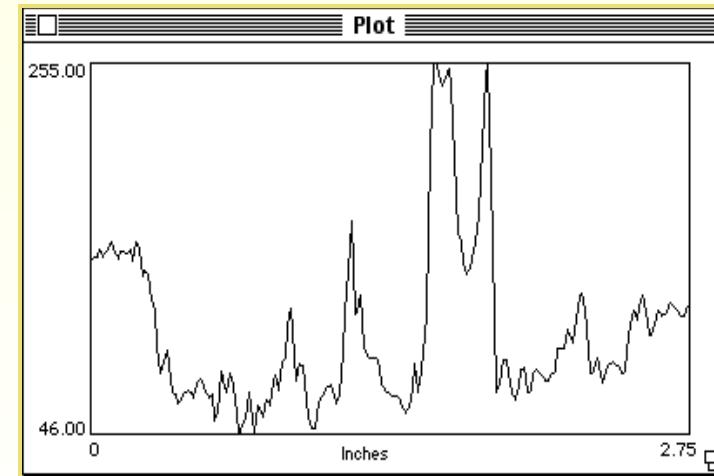
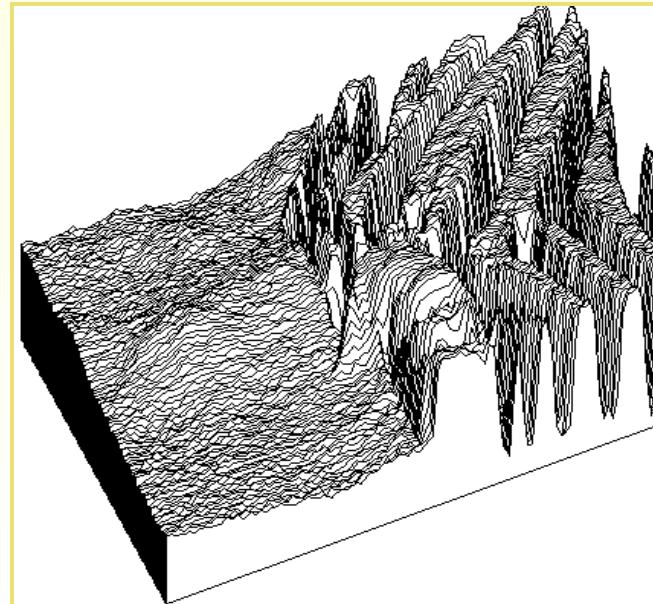
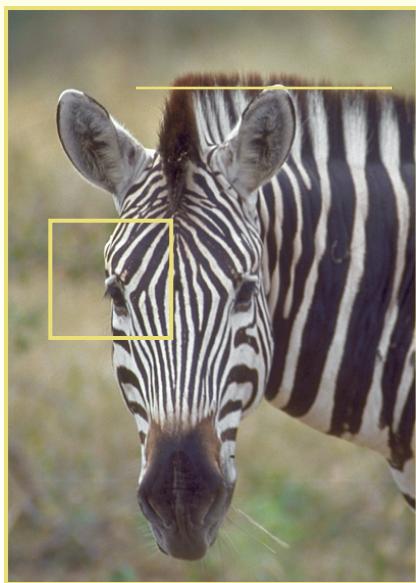
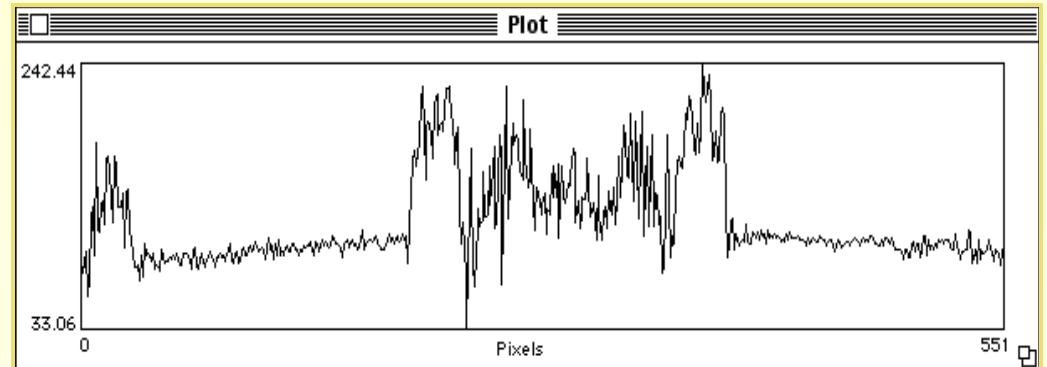
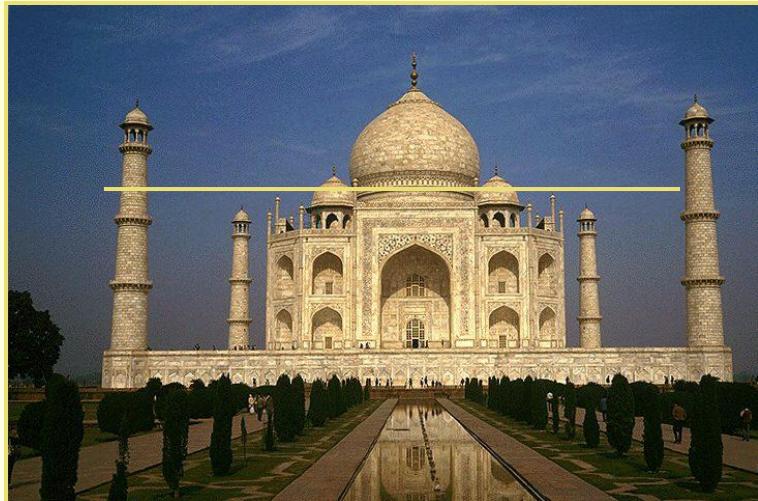


2nd Derivada $f''(x)$





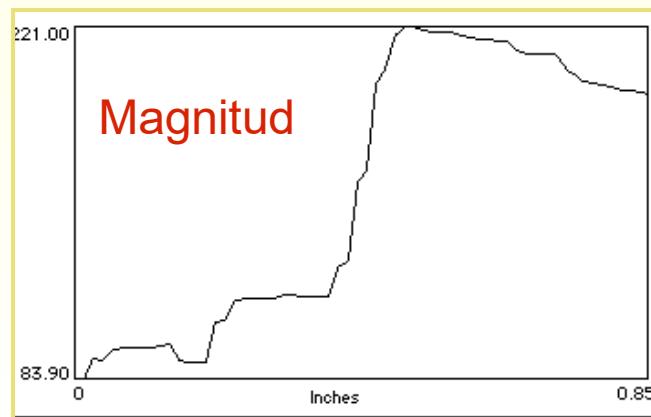
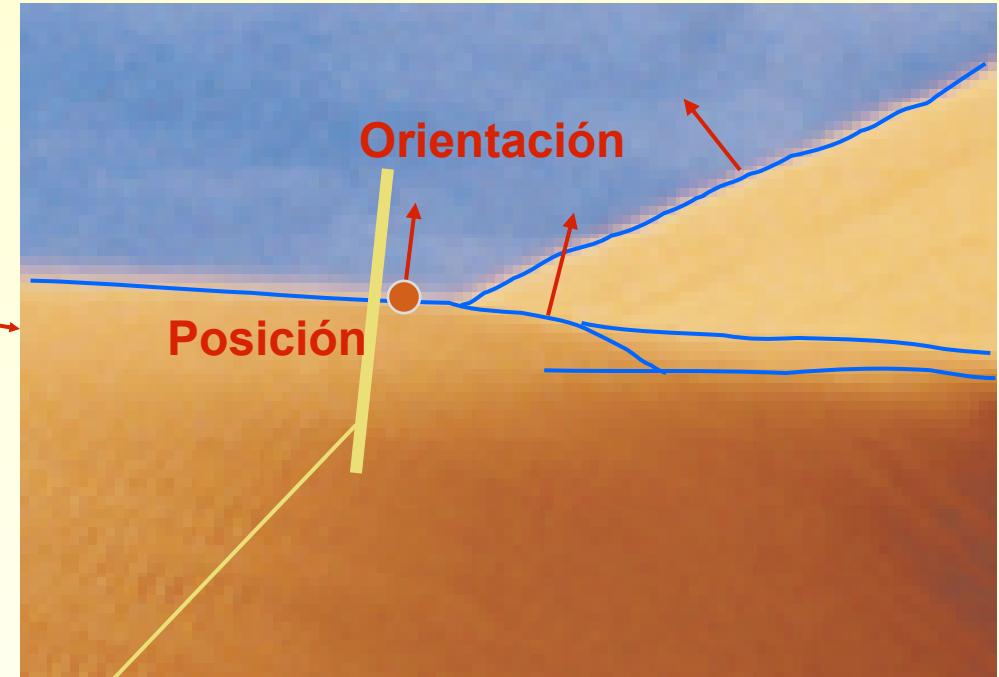
Extracción de Fronteras Realidad





Extracción de Fronteras

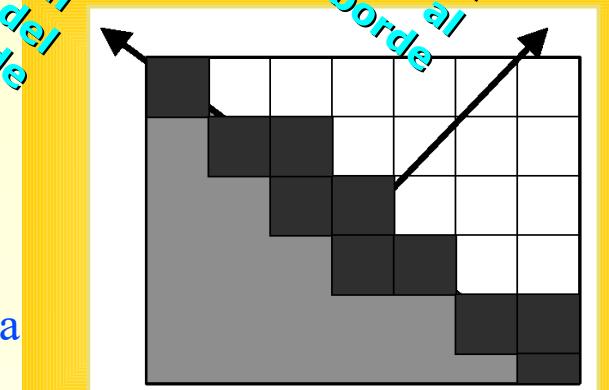
Propiedades de un borde





Orientación del borde

- Normal al borde - vector unitario en la dirección del cambio de intensidad máxima en el nivel de gris/color (máximo gradiente)
- Dirección del borde – vector unitario perpendicular a la normal del borde



Posición del borde o Centro

- Posición de la imagen en la que se localiza el borde (normalmente se almacena como un dato binario, por lo que la imagen de bordes suele ser una imagen binaria)

Fuerza del borde / Magnitud

- Relacionada con el contraste local o gradiente – cómo de rápida es la variación de intensidad en el borde a lo largo de la normal.

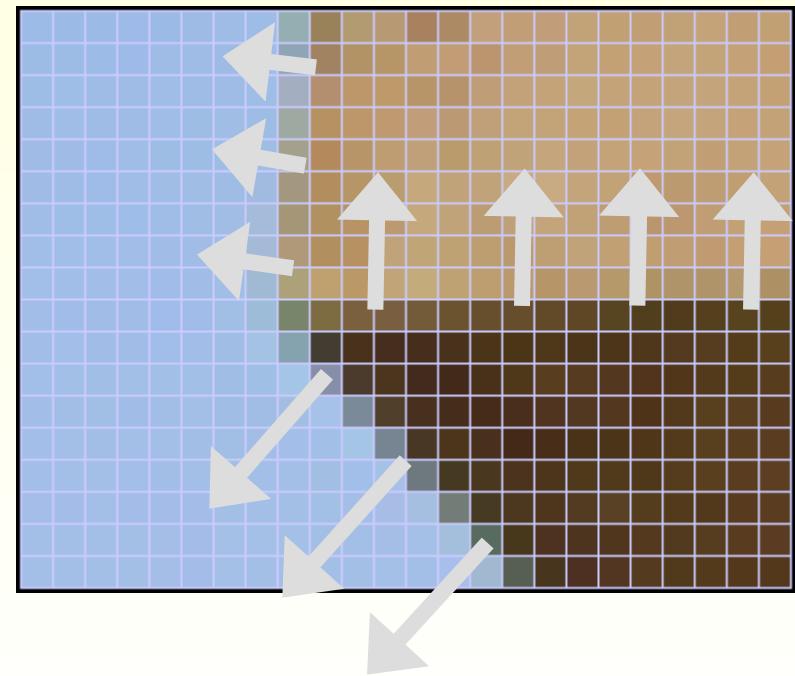
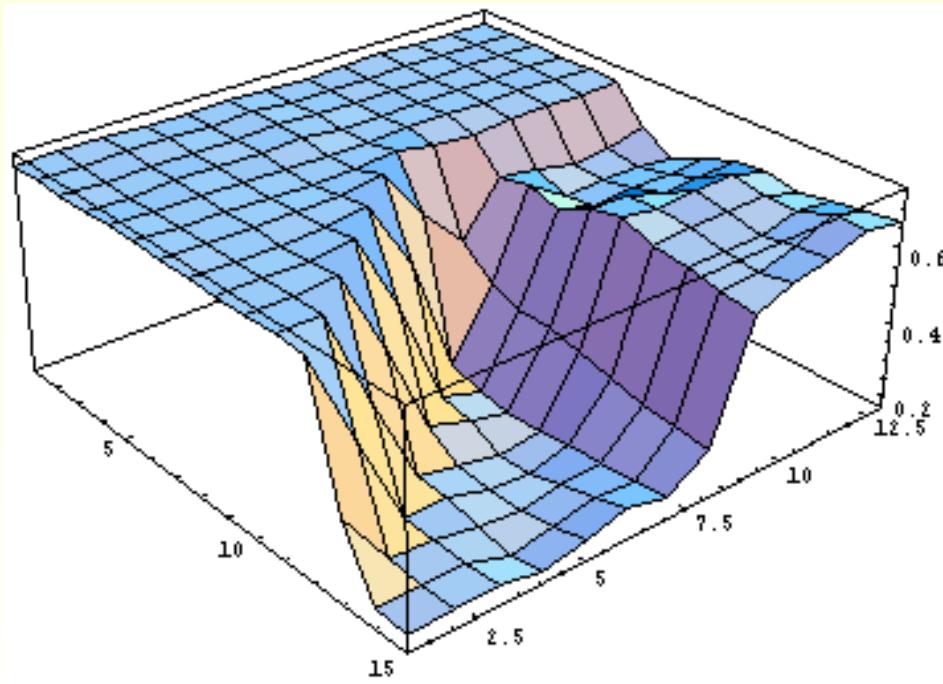


Extracción de Fronteras

Descriptores cuantitativos de bordes



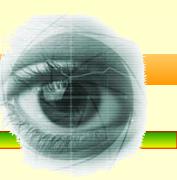
- **El gradiente** indica la dirección de máxima variación de una función (en 2D, la máxima pendiente).





Extracción de Fronteras

Descriptores cuantitativos de bordes



- El gradiente da lugar al concepto de **borde**.
- Un **borde** en una imagen es una curva a lo largo de la cual el gradiente es máximo.

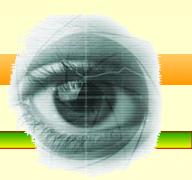


El borde es perpendicular a la dirección del gradiente.



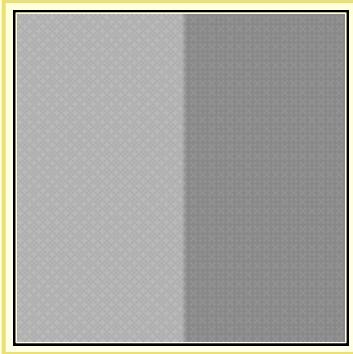
Extracción de Fronteras

Degradación de bordes por ruido

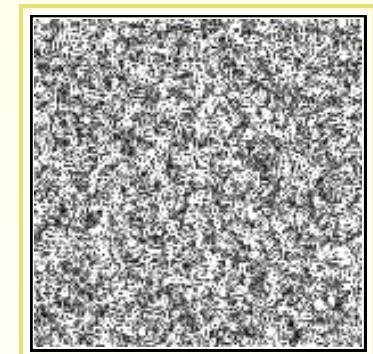
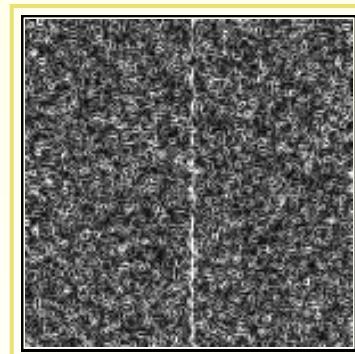
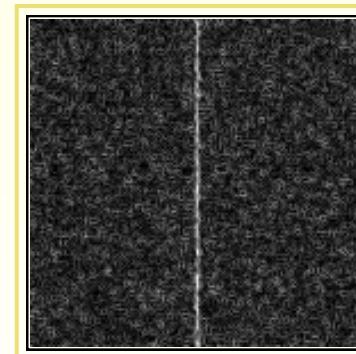
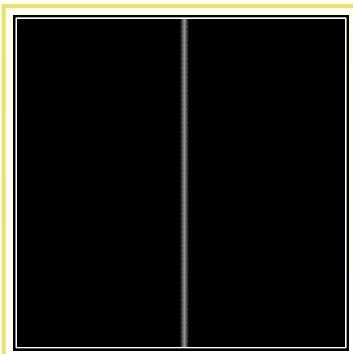
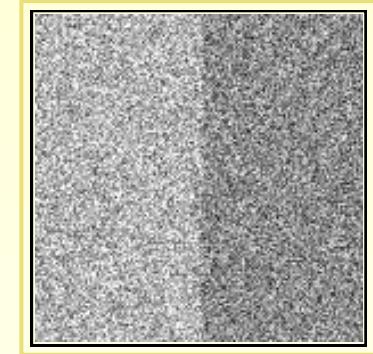
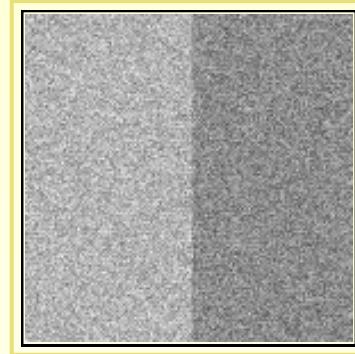


Ruido creciente →

Salto Ideal



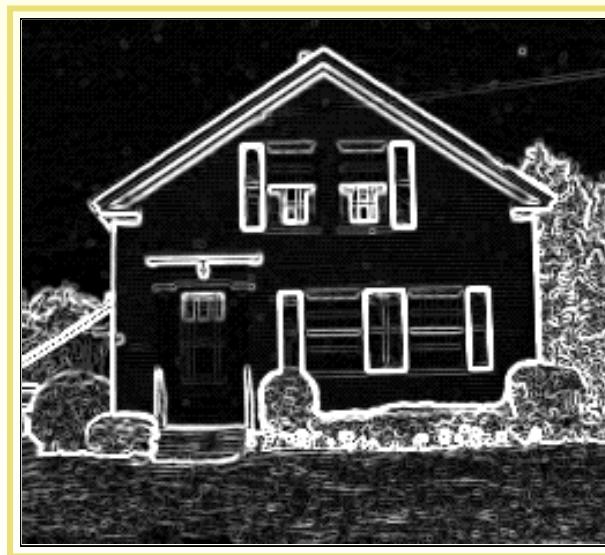
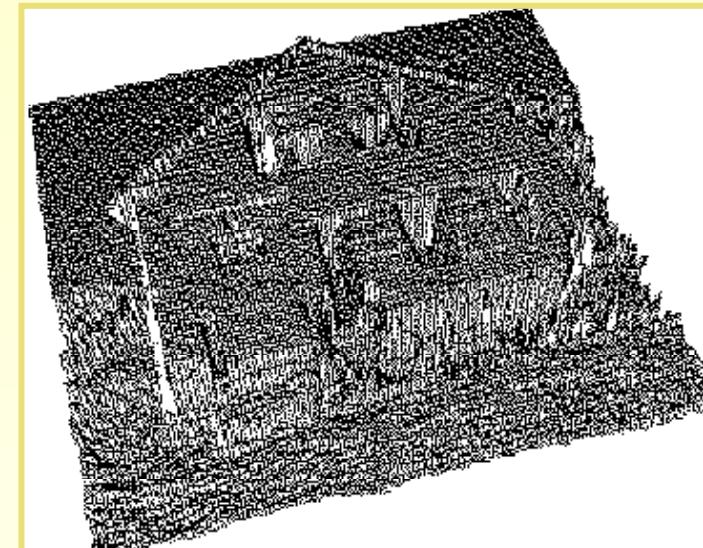
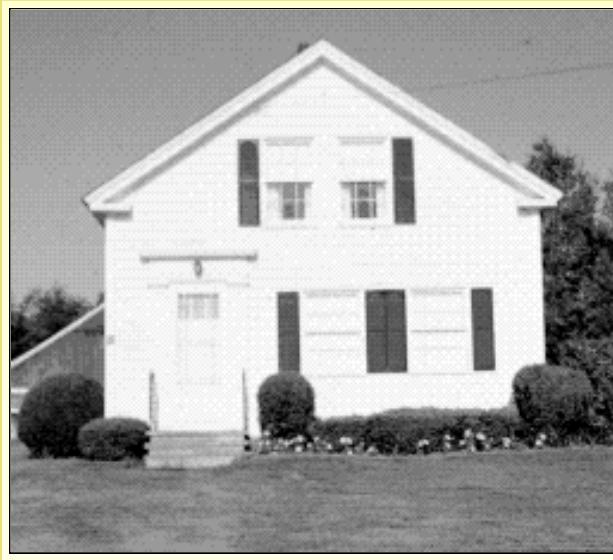
Salto ideal + ruido

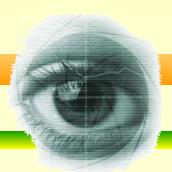




Extracción de Fronteras

Ejemplo de Imagen Real





- Alisamiento de ruido
 - Eliminar tanto ruido como sea posible, conservando las fronteras “verdaderas”
 - En ausencia de información adicional, asumir ruido “blanco” con una distribución Gaussiana
- Realce de fronteras
 - Diseñar un filtro que responda a fronteras; un valor alto en la salida del filtro corresponde a píxeles frontera y un valor bajo corresponde al resto.
- Localización de fronteras
 - Determinar qué píxeles frontera deberían descartarse por corresponder a ruido y cuales deberían mantenerse.
 - Adelgazar fronteras gruesas a 1 pixel de anchura (supresión de no-máximos)
 - Establecer el mínimo valor para declarar un máximo local a partir de un filtro de detección de bordes como una frontera (umbralización).



Extracción de Fronteras

Metodos de Detección

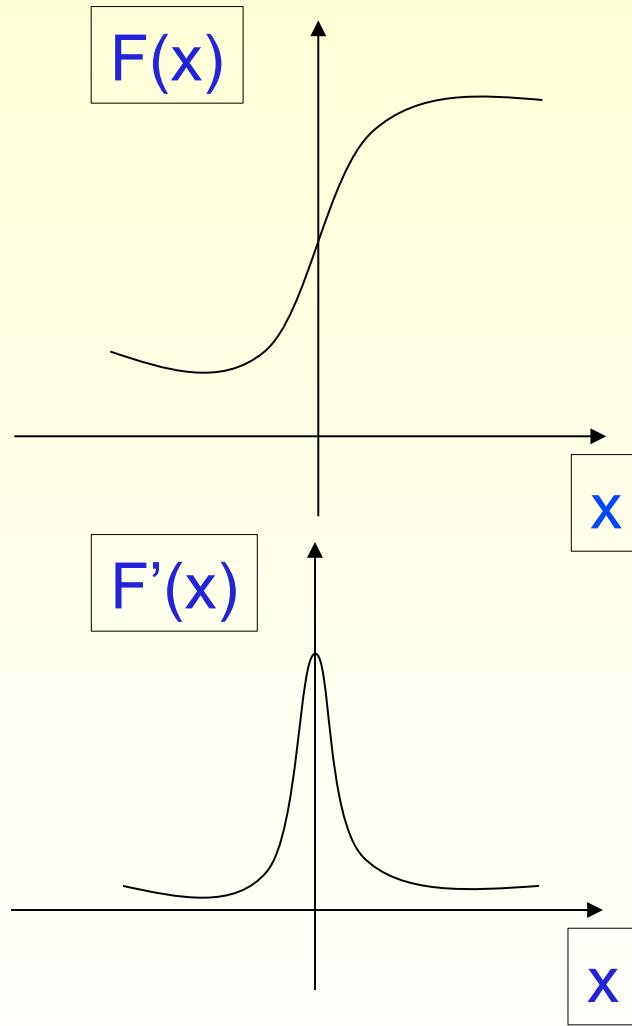


- Estimadores 1^a Derivada
 - Detectores de Gradiente
 - Detección combinada de bordes ("compass masks ó máscaras de brújula")
 - Detector de Canny (*)
- Estimadores 2^a Derivada
 - Laplaciana
 - Diferencia de Gaussianas
- Modelos Paramétricos (*)

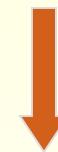


Extracción de Fronteras

Metodos Basados en el Gradiente



Borde= variación brusca

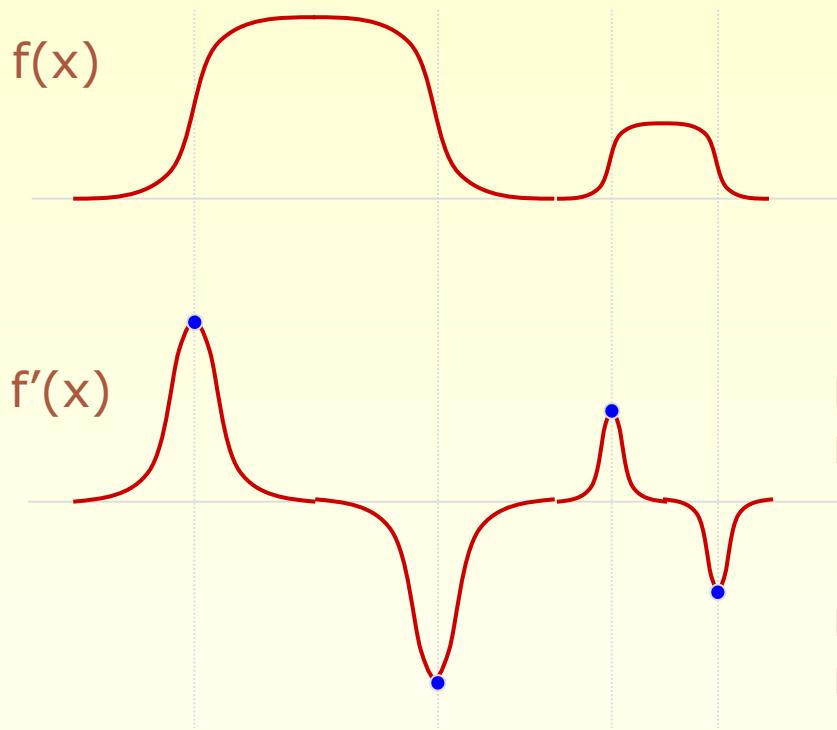


Valor alto en la 1^a derivada



Extracción de Fronteras

La derivada de una imagen



Los máximos coinciden con el punto medio de las fronteras que varían de menos a más.

Los mínimos coinciden con el punto medio de las fronteras que varían de más a menos.

En el caso de una imagen 2D, la derivada se puede expresar como un vector de dos componentes:

$$f'(x, y) = \frac{\partial f}{\partial x} \vec{x} + \frac{\partial f}{\partial y} \vec{y}$$



Extracción de Fronteras

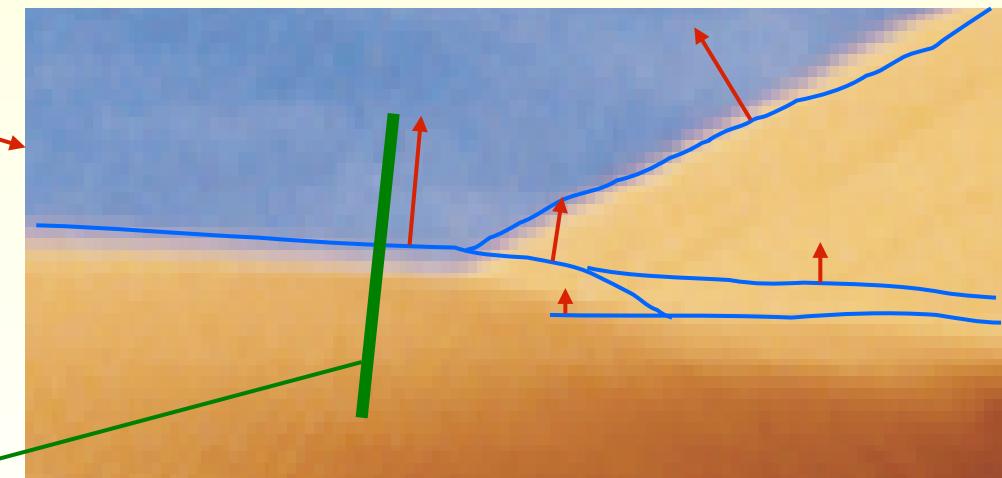
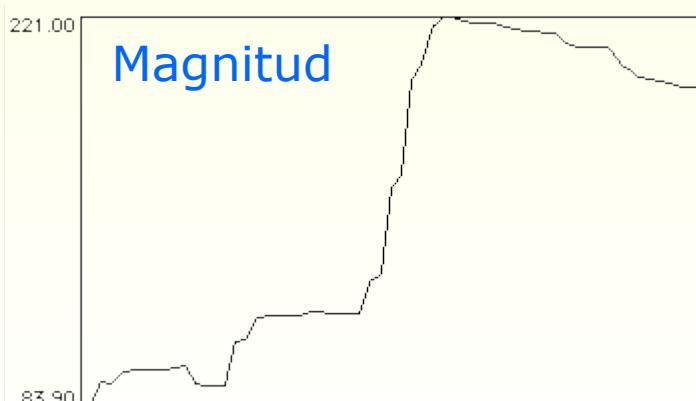
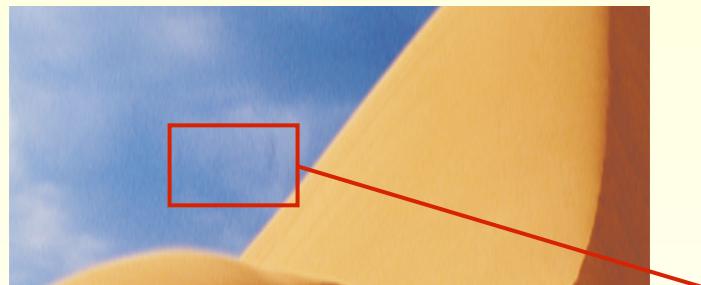
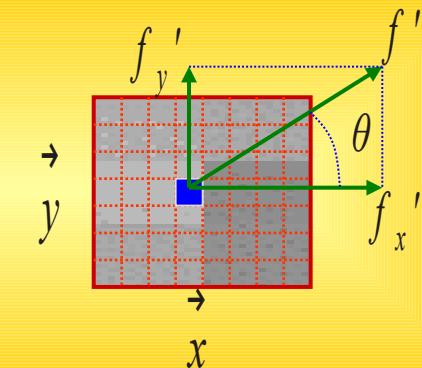
La derivada de una imagen



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

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

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



Las fronteras tienen orientación y magnitud



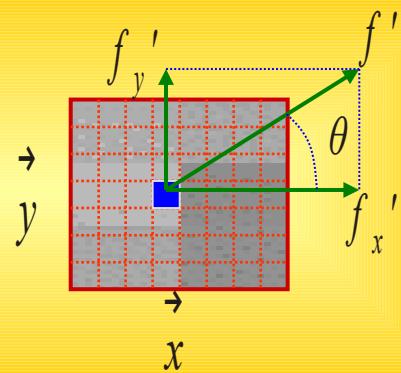
Extracción de Fronteras

La derivada de una imagen

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

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

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



Magnitud

$$m(x, y) = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2}$$

Dirección

$$\theta(x, y) = \tan^{-1} \left(\frac{\frac{\partial f(x, y)}{\partial y}}{\frac{\partial f(x, y)}{\partial x}} \right)$$



- En resumen: Si f es una función continua en (x,y) . Entonces

$$\Delta_x = \frac{\partial f}{\partial x}, \quad \Delta_y = \frac{\partial f}{\partial y}$$

- Son las tasas de cambio de la función f en las direcciones x e y resp.
- El vector (Δ_x, Δ_y) se denomina gradiente de f .
- Este vector tiene una magnitud:

$$s = \sqrt{\Delta_x^2 + \Delta_y^2}$$

y una orientación:

$$\theta = \tan^{-1} \left(\frac{\Delta_y}{\Delta_x} \right)$$

- θ es la dirección del máximo cambio en f .
- s es el tamaño de ese cambio.

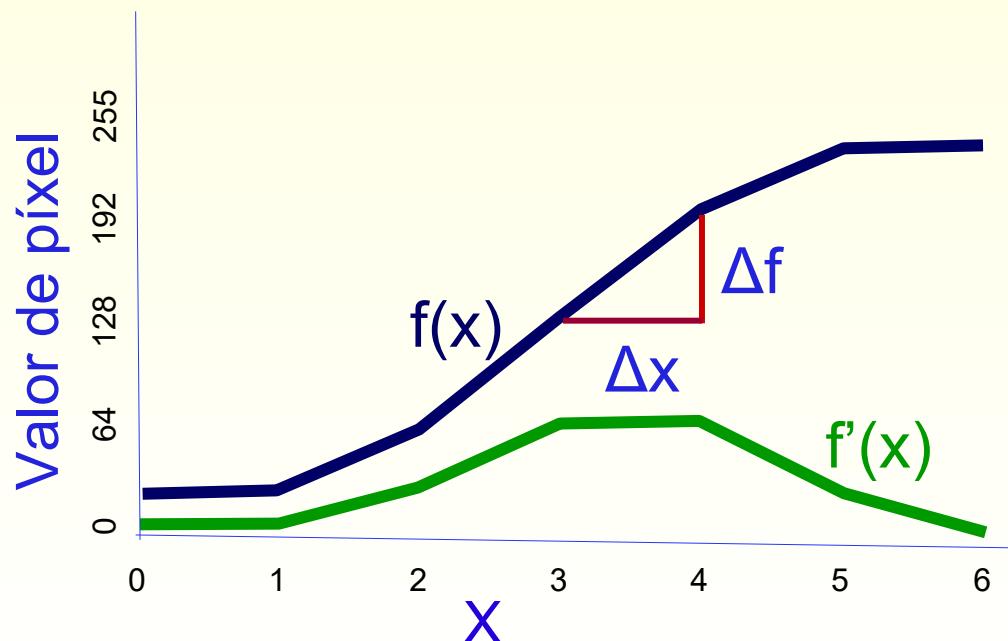


Extracción de Fronteras

Aproximaciones discretas



- Matemáticamente, la variación de una función $f(x)$ cualquiera viene dada por la **derivada** de esa función:
 - $f'(x) > 0$: función creciente en X
 - $f'(x) < 0$: función decreciente en X
 - $f'(x) = 0$: función uniforme en X
- En nuestro caso, tenemos **funciones discretas**. La “**derivada discreta**” se obtiene calculando diferencias.



$$f'(x) = \Delta f / \Delta x$$
$$\Delta f = f(x) - f(x-1) \quad \Delta x = 1$$

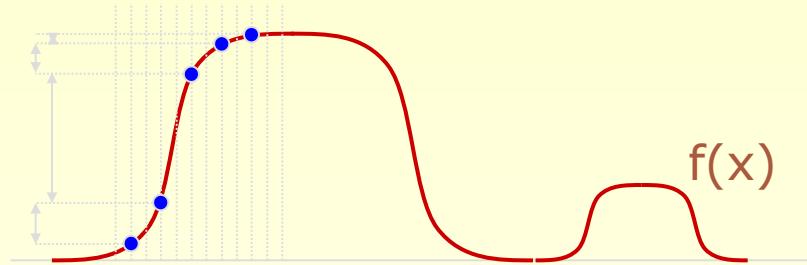
$$f'(x) = f(x) - f(x-1)$$

Conclusión: la derivada se calculará con máscaras del tipo:

-1	1
----	---



¿Cómo calculamos la derivada en una dirección?

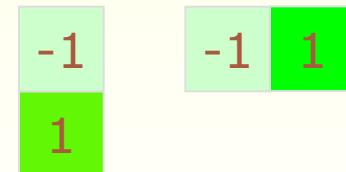


La derivada es una diferencia entre puntos adyacentes. La forma más simple de calcularla sería:

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

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

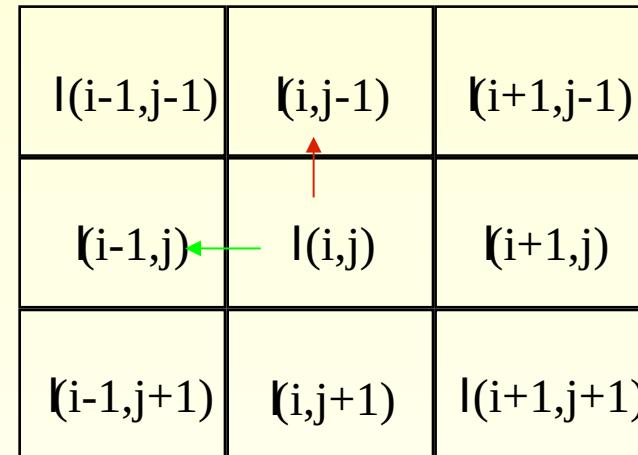
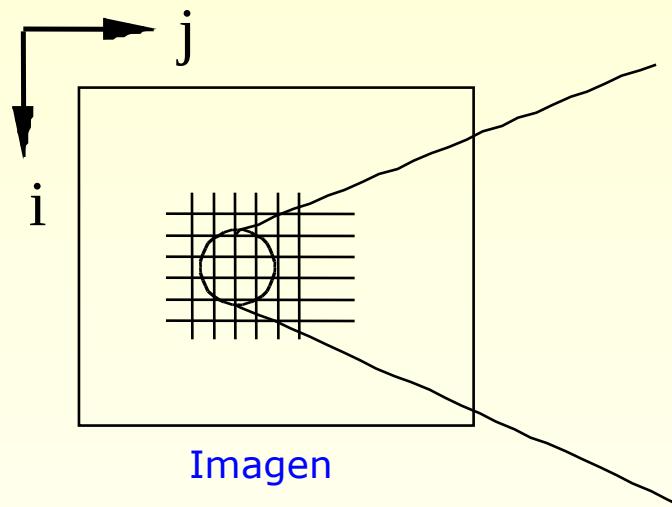
Que es lo mismo que convolucionar la imagen con los filtros:



Muy sensibles al ruido



■ Función de Imagen Discreta I



■ Derivadas → Diferencias

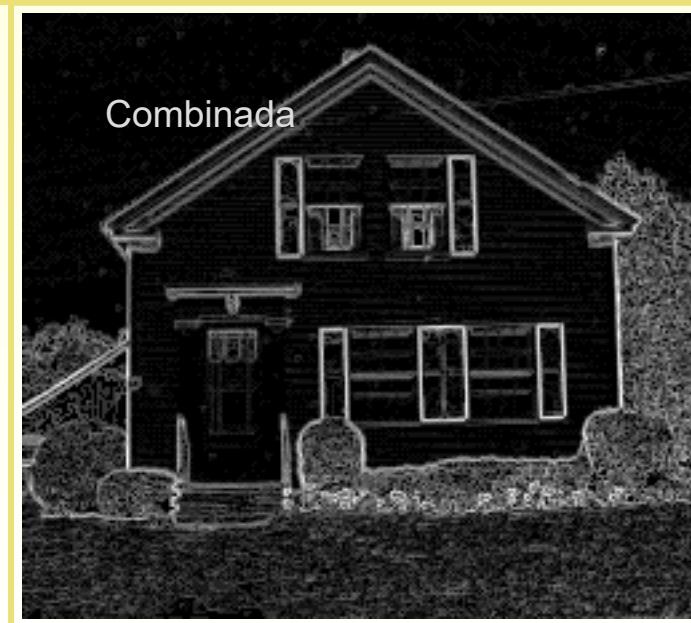
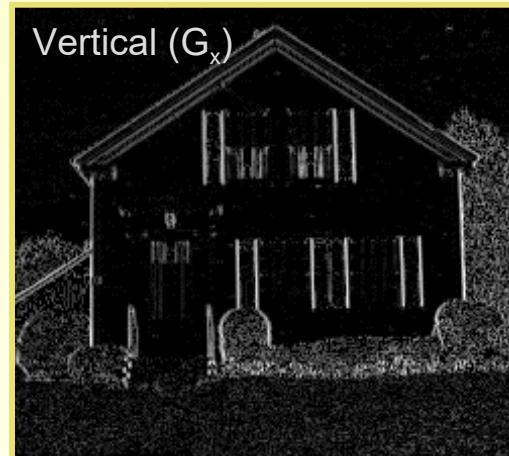
$$\Delta_i I = \begin{matrix} -1 & 1 \\ \bullet & \end{matrix}$$

$$\Delta_j I = \begin{matrix} -1 \\ \bullet \\ 1 \end{matrix}$$



Extracción de Fronteras

Ejemplo 1x2





Extracción de Fronteras

Diferentes máscaras de derivación



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

Diferencia hacia atrás

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

Diferencia hacia delante

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

Diferencia central

$$f(x) = \begin{matrix} 10 & 15 & 10 & 10 & 25 & 20 & 20 & 20 \end{matrix}$$

Máscaras
de
Derivación

$$f'(x) = \begin{matrix} 0 & 5 & -5 & 0 & 15 & -5 & 0 & 0 \end{matrix}$$

Diferencia
hacia atrás $[-1 \ 1]$

$$f''(x) = \begin{matrix} 0 & 5 & -10 & 5 & 15 & 20 & 5 & 0 \end{matrix}$$

Diferencia
hacia delante $[1 \ -1]$

Diferencia
central $[-1 \ 0 \ 1]$



Extracción de Fronteras

Ejemplo



Imagen I



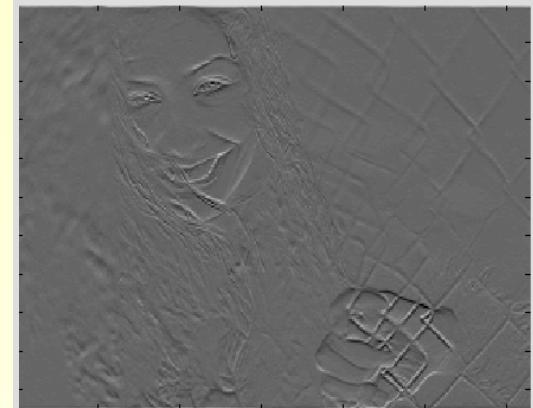
$$I_x = I * \begin{bmatrix} 1 & -1 \end{bmatrix}$$

Derivadas y ruido

Fuertemente afectadas por el ruido

- Razón obvia: el ruido en la imagen da lugar a píxeles que son muy diferentes de sus vecinos

Cuanto mayor es el ruido más fuerte es la respuesta



$$I_y = I * \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

¿Qué se puede hacer?

- Los píxeles vecinos parecen similares
- Los píxeles a lo largo de una frontera parecen similares
- El alisamiento de la imagen podría ayudar
- Forzar a píxeles diferentes a sus vecinos (posiblemente ruido) a parecerse a sus vecinos



- Es posible (y adecuado) combinar los operadores de **bordes** con **suavizados**.

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





Extracción de Fronteras

Alisamiento y Detección de bordes



- Las derivadas son operaciones 'ruidosas'
 - Las fronteras son fenómenos de las altas frecuencias espaciales
 - Los detectores de fronteras son sensibles a un ruido acentuado.
- Promediar reduce el ruido
 - Las medias espaciales pueden calcularse usando máscaras.

1	1	1
1	1	1
1	1	1

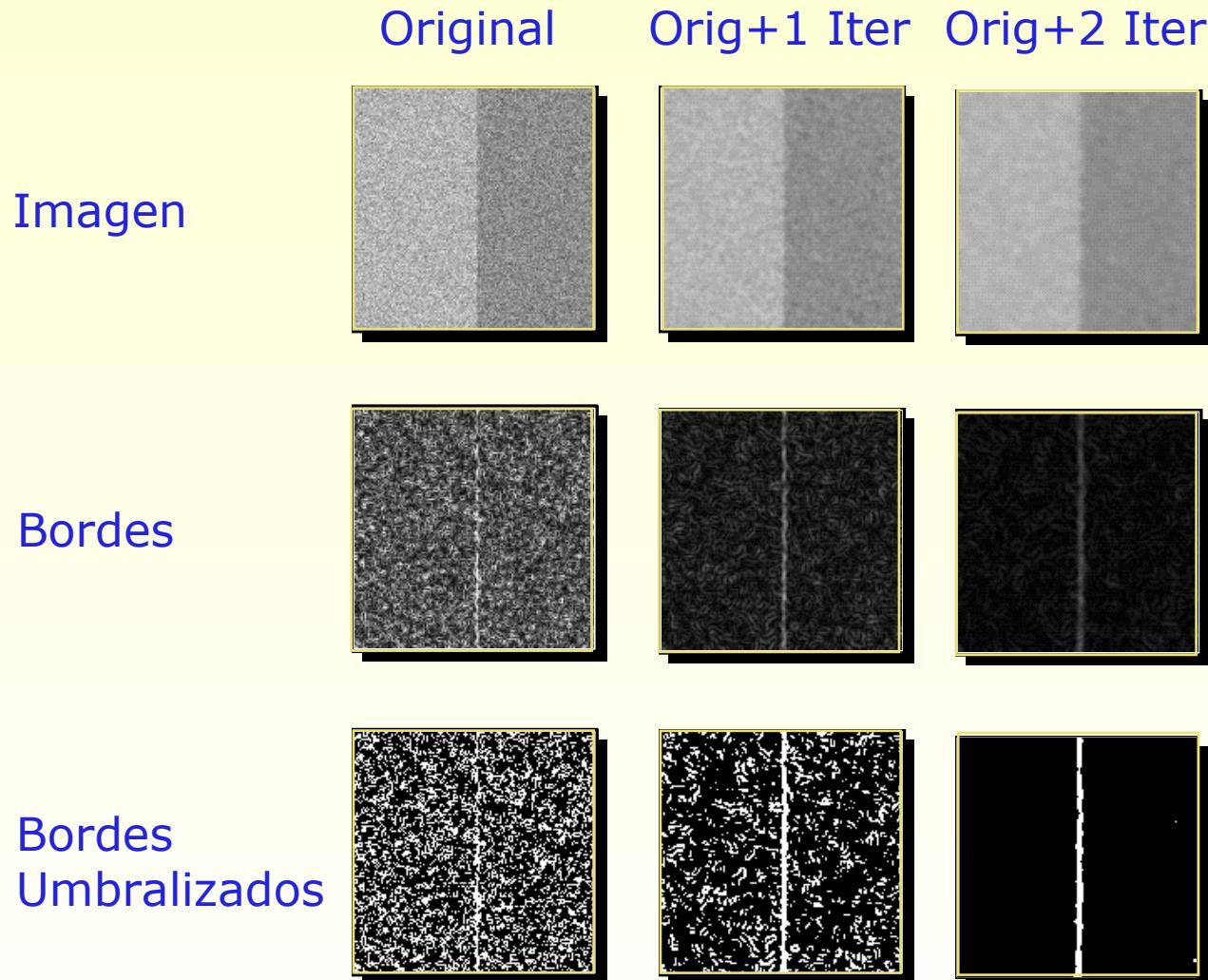
1	1	1
1	0	1
1	1	1

- Combinar alisamiento con detección de bordes.



Extracción de Fronteras

Efecto del emborronamiento

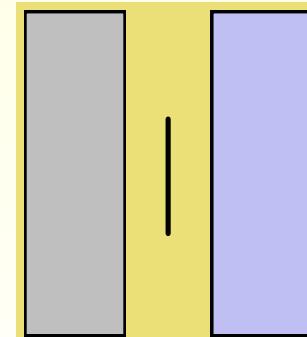




Extracción de Fronteras Combinando las dos

Aplicar esta máscara es equivalente a tomar la diferencia de medias a cada lado del pixel central.

-1	-1	-1
0	●	0
1	1	1



Variables
Tamaño del núcleo
Patrón de pesos



Extracción de Fronteras

Ejemplo



El rango de los pixeles de la imagen de aristas puede estar en [-255,255] pero el rango válido es [0,255]. Como solucionarlo:

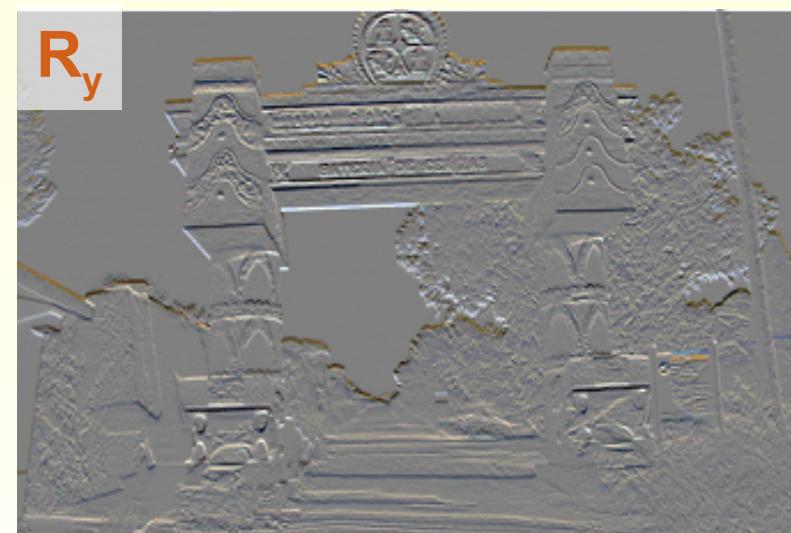
Los bordes decrecientes se saturan a 0...

Podemos sumar 128 para apreciar mejor el resultado:

- Gris (128): diferencia 0
- Negro: decreciente
- Blanco: creciente



Derivada X (+128)

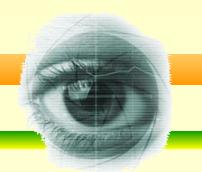


Derivada Y (+128)

- Se produce una especie de “**bajorrelieve**”, que puede usarse como “efectos especiales”...



Extracción de Fronteras Operador cruzado de Roberts



$$S = \sqrt{[I(x, y) - I(x+1, y+1)]^2 + [I(x, y+1) - I(x+1, y)]^2}$$

ó

$$S = | I(x, y) - I(x+1, y+1) | + | I(x, y+1) - I(x+1, y) |$$

$$\begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix} + \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix}$$

No devuelve ninguna información sobre la orientación del borde.



Extracción de Fronteras

Operador de Prewitt



$$P_1 = \begin{matrix} I_y \\ \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array} \end{matrix}$$

$$P_2 = \begin{matrix} I_x \\ \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \end{matrix}$$

$$\text{Magnitud del borde} = \sqrt{P_1^2 + P_2^2}$$

$$\text{Dirección del borde} = \tan^{-1} \left(\frac{P_1}{P_2} \right)$$

$$I_x$$



Extracción de Fronteras Máscaras de Prewitt



Los kernels tipo Prewitt son separables

$$\begin{array}{ccc} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{array}$$

$$= \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \times \begin{array}{ccc} -1 & 0 & 1 \end{array}$$

$$\begin{array}{ccc} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{array}$$

$$= \begin{array}{c} -1 \\ 0 \\ 1 \end{array} \times \begin{array}{ccc} 1 & 1 & 1 \end{array}$$

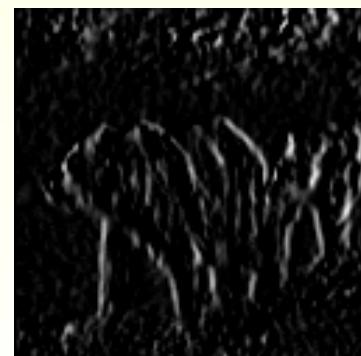
Incluye una suavización por lo que es menos sensible al ruido.

*Se usan más puntos del entorno:
la derivada es más precisa.*

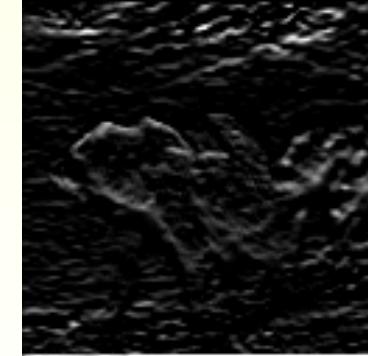
$$\begin{array}{ccc} -1 & & 1 \\ -1 & 1 & 1 \\ -1 & & 1 \end{array}$$



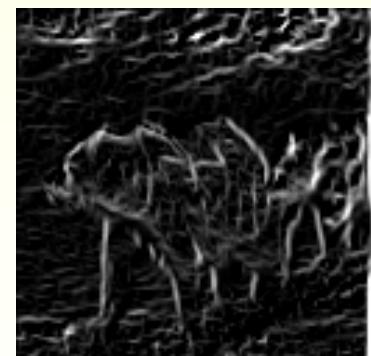
$f(x,y)$



$f'_x(x,y)$



$f'_y(x,y)$

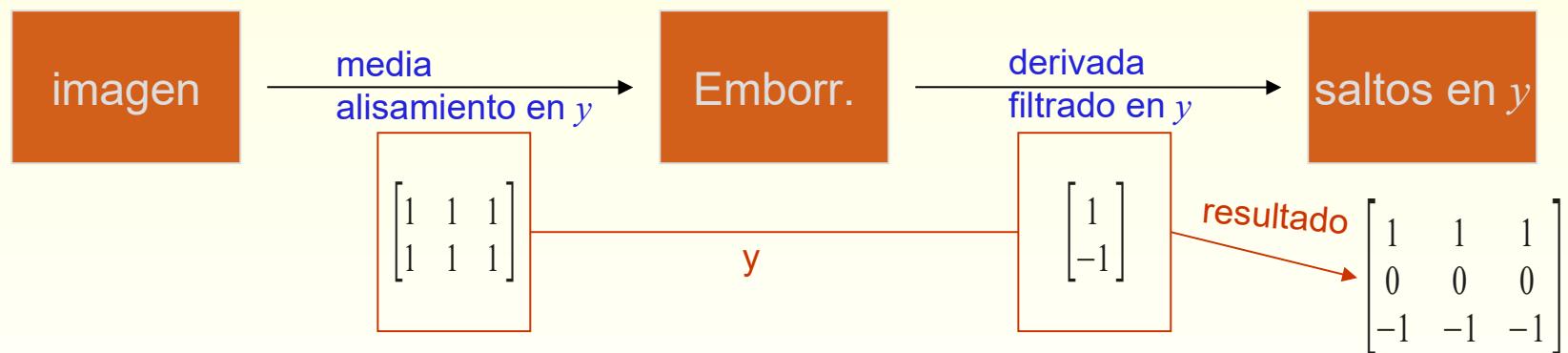
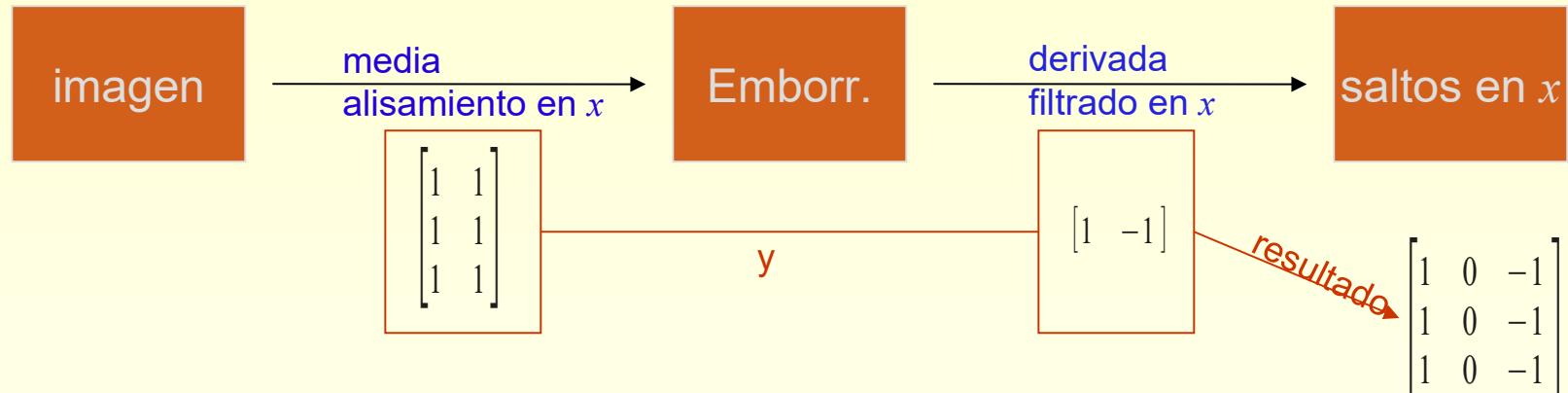


$m(x,y)$



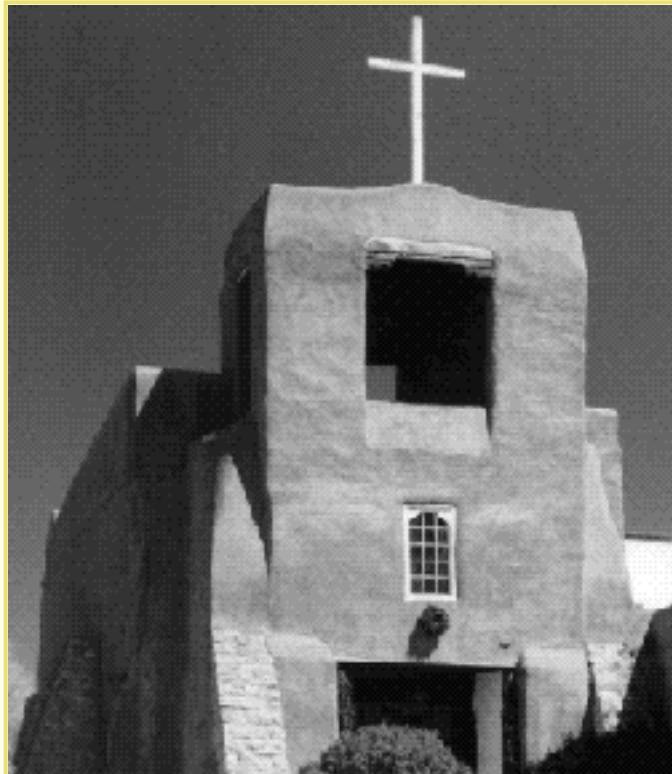
Extracción de Fronteras

Operadores de Prewitt

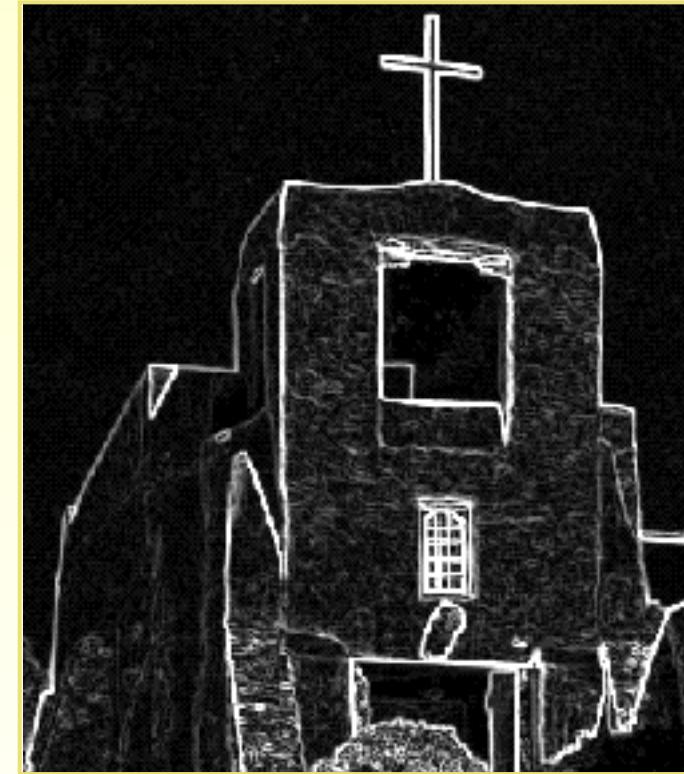




Extracción de Fronteras Ejemplo de Prewitt



Ermita



Salida de Prewitt
combinando bordes
horizontales y verticales.



Extracción de Fronteras Operador de Sobel



$$S_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Magnitud del borde = $\sqrt{S_1^2 + S_2^2}$

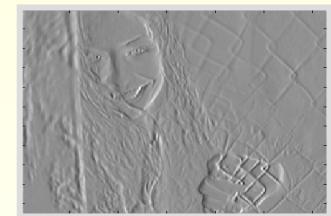
$$S_2 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Dirección del borde = $\tan^{-1} \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}$



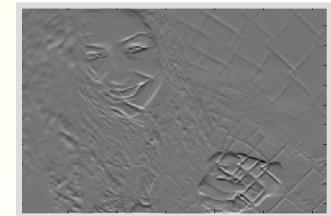
Fronteras de Sobel en la dirección x

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \rightarrow I_x$$



Fronteras de Sobel en la dirección y

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \rightarrow I_y$$



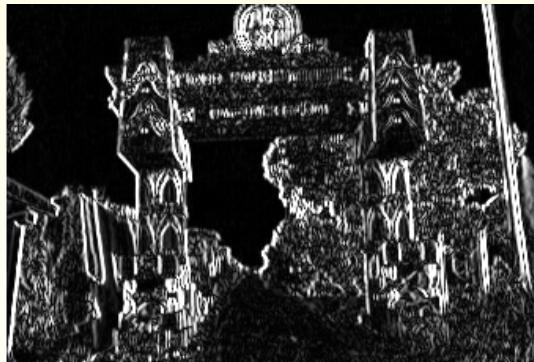
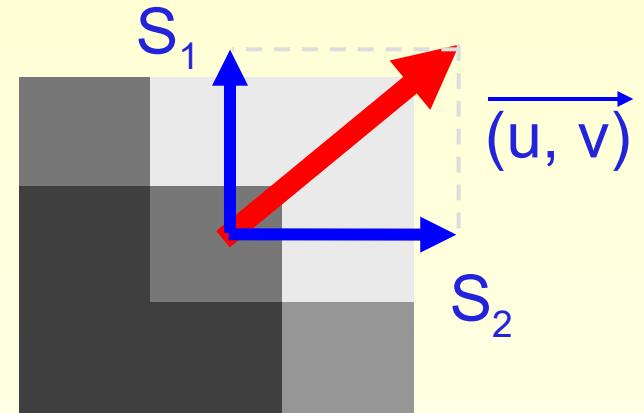


Extracción de Fronteras Operador de Sobel

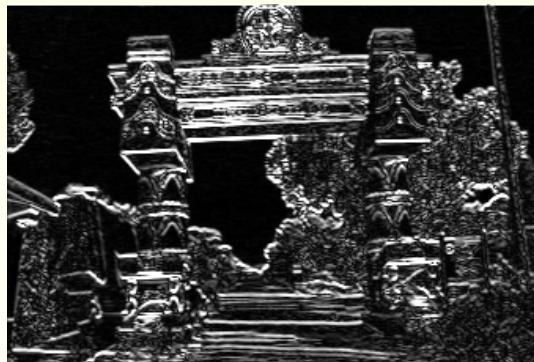


Cálculo del gradiente:

- Calcular **derivada en Y: S_1**
- Calcular **derivada en X: S_2**
- **Magnitud** del gradiente: $\sqrt{S_1^2 + S_2^2}$
- **Ángulo** del gradiente: $\arctan(S_1, S_2)$



Valor absoluto de
derivada en X
(Sobel de 3x3)



Valor absoluto de
derivada en Y
(Sobel de 3x3)



Magnitud del gradiente



Extracción de Fronteras

Máscaras de Sobel

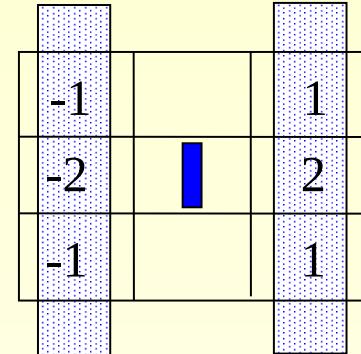


Los kernels tipo Sobel son separables

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

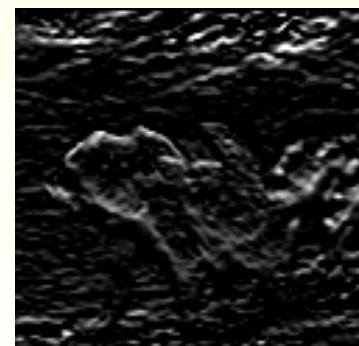
Se da más importancia a los píxeles cercanos al punto central.



Se promedia paralelamente al borde



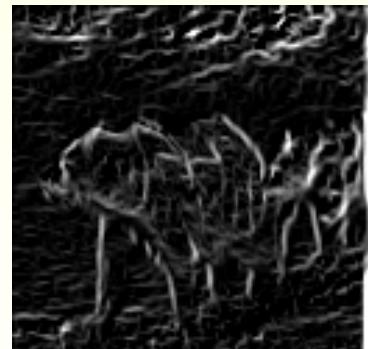
$f(x,y)$



$f'_y(x,y)$



$f'_x(x,y)$

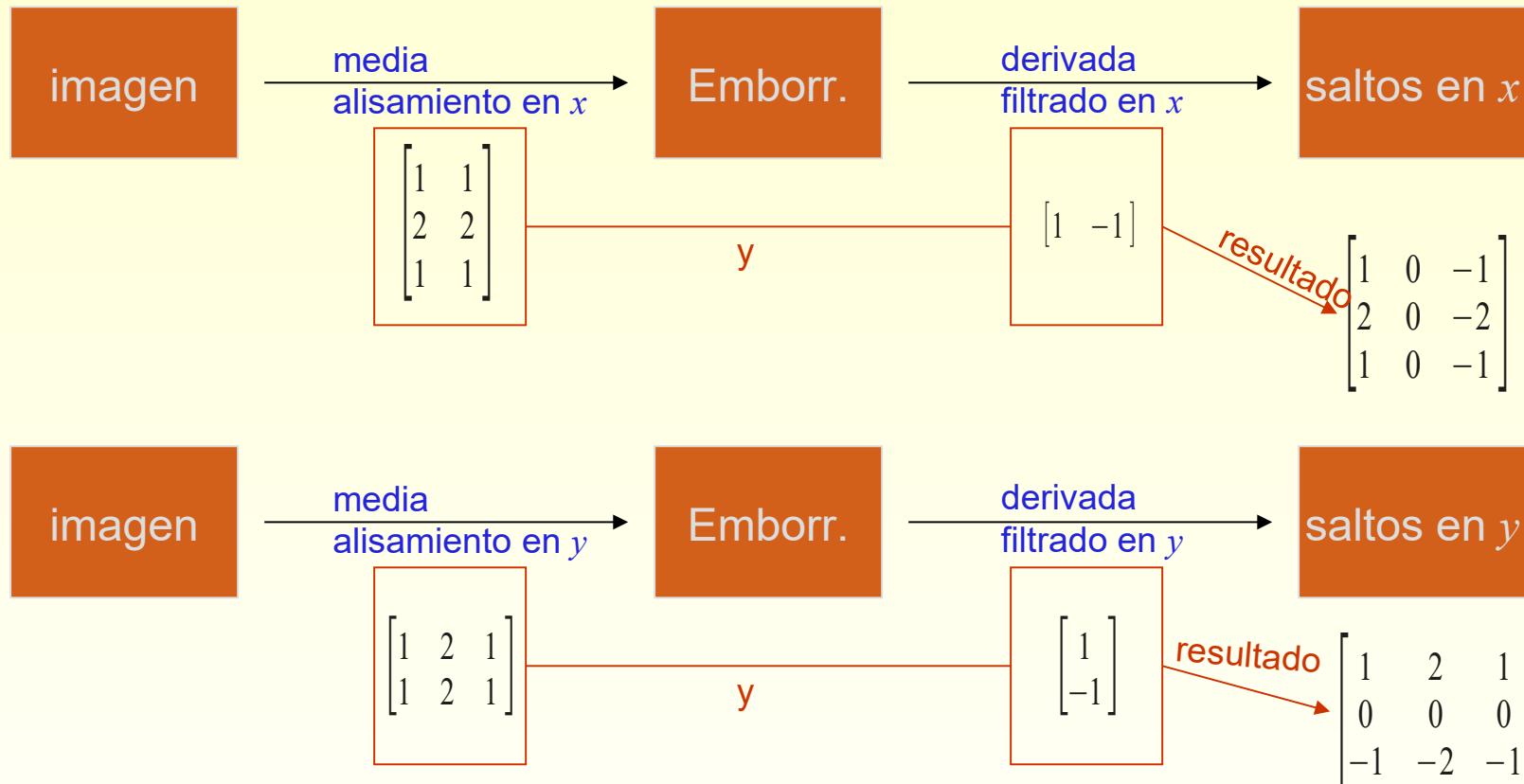


$m(x,y)$



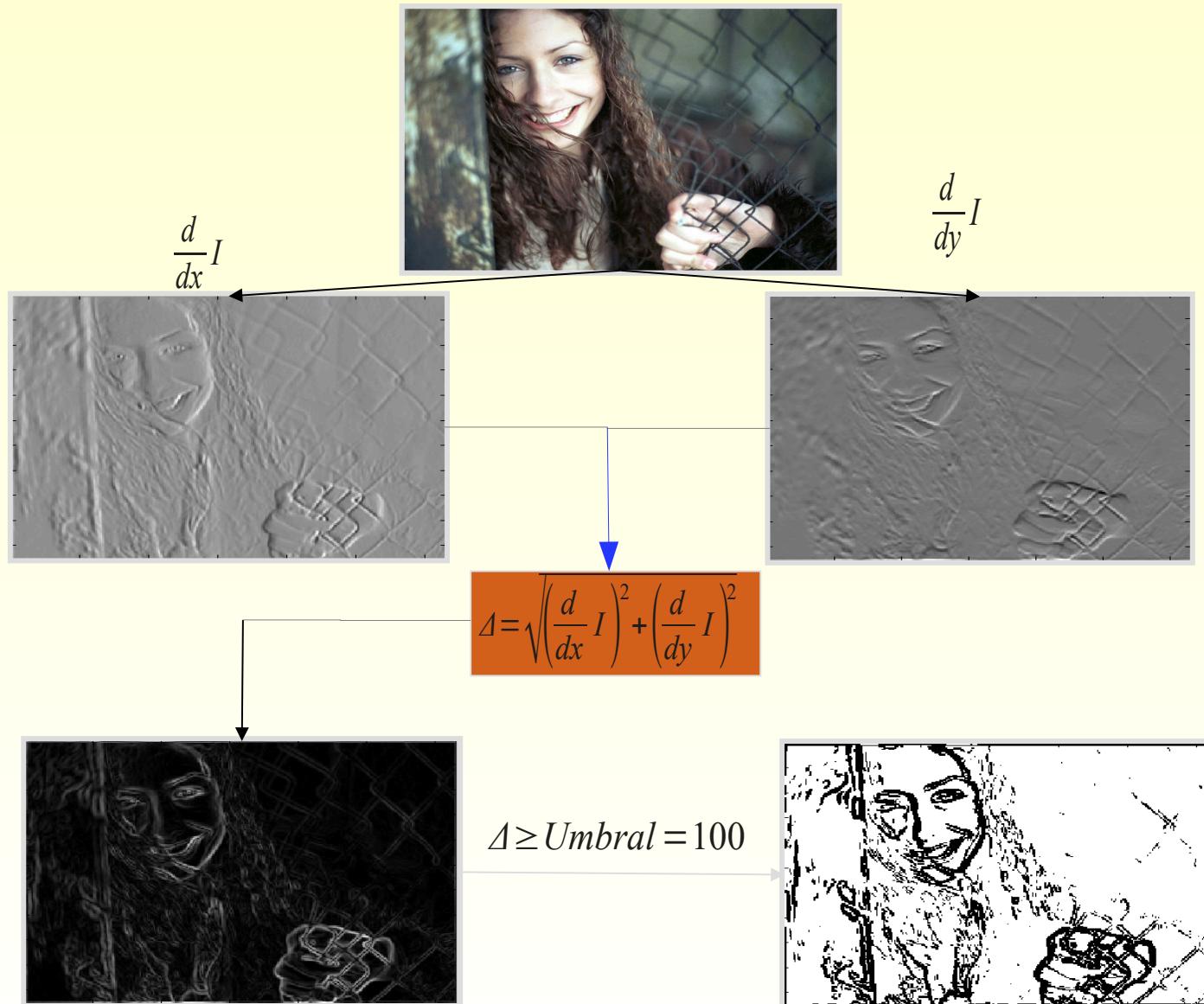
Extracción de Fronteras

Operador de Sobel





Extracción de Fronteras Operador de Sobel



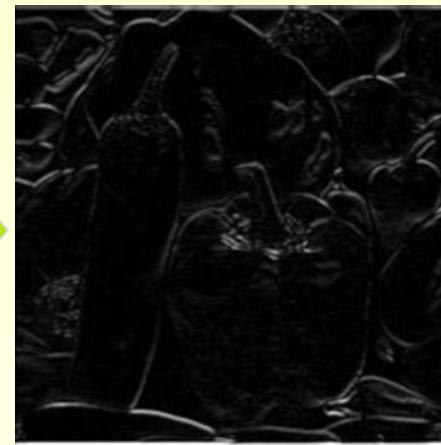


Extracción de Fronteras

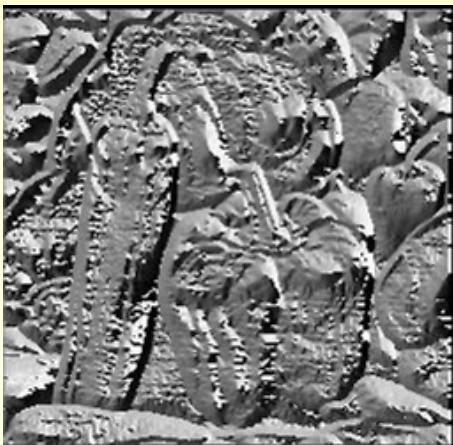
Operador de Sobel



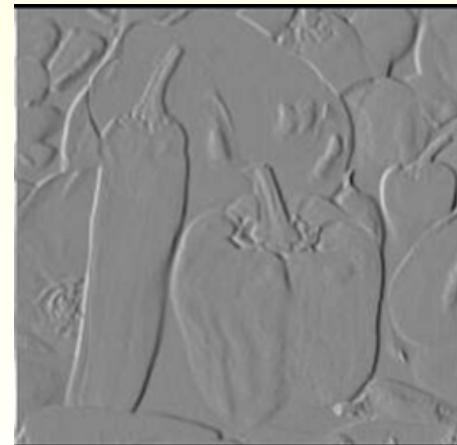
Original



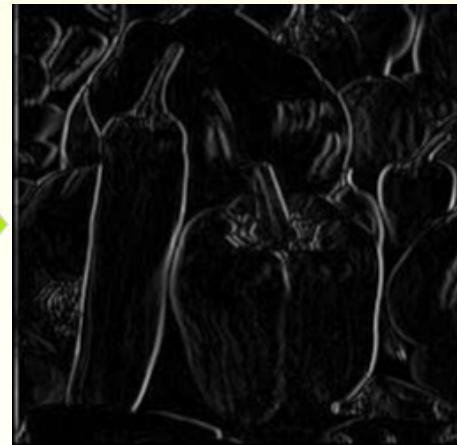
$$|f_y'(x, y)|$$



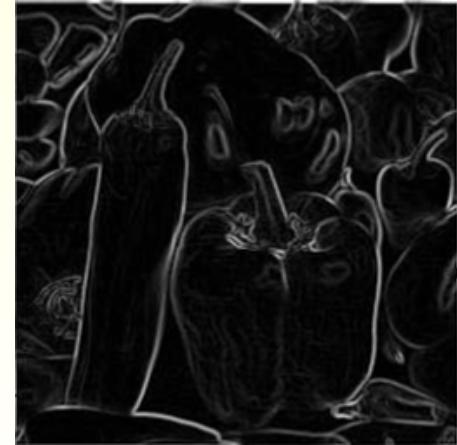
$$\theta(x, y)$$



$$|f_x'(x, y)|$$



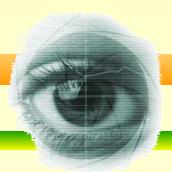
$$|f_x'(x, y)|$$



$$m(x, y)$$



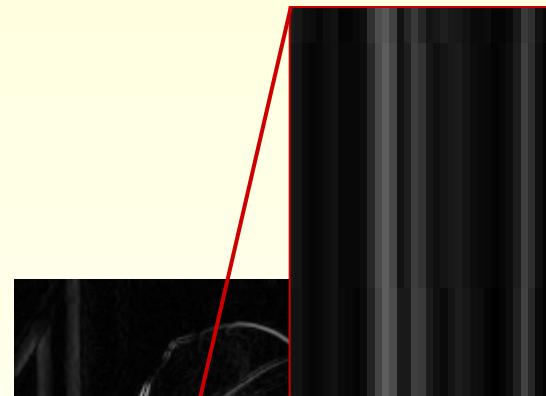
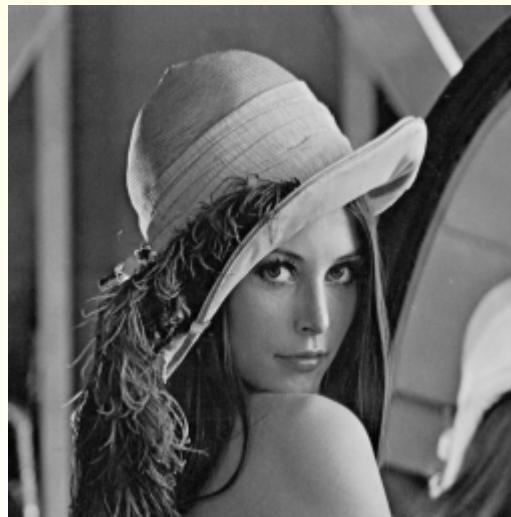
Otros filtros para calcular el gradiente



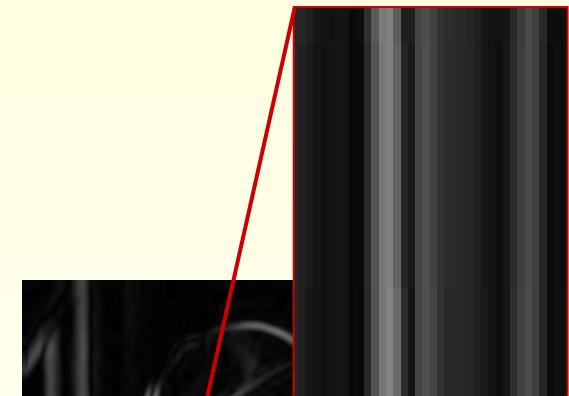
-1	-2	0	2	1
-1	-2	0	2	1
-1	-2	0	2	1
-1	-2	0	2	1
-1	-2	0	2	1

Podemos aumentar el tamaño de estos filtros para considerar más puntos del entorno:

- Evitamos que el ruido influya demasiado.
- Perdemos la localización exacta de la frontera.



Sobel 5x5



Sobel 15x15



Otras formas de calcular los bordes:

- 1.-Calcular la derivada en diferentes direcciones: D1, D2, D3, D4.
- 2.- Para cada punto, la magnitud del gradiente es la derivada de máximo valor absoluto:

$$G(x,y) := \max \{ |D1(x,y)|, |D2(x,y)|, |D3(x,y)|, |D4(x,y)| \}$$

La dirección del gradiente viene dada por el ángulo que ha producido el máximo:

$$A(x,y) := \operatorname{argmax} \{ |D1(x,y)|, |D2(x,y)|, |D3(x,y)|, |D4(x,y)| \}$$

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

D₁: N-S

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

D₂: NE-SO

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

D₃: E-O

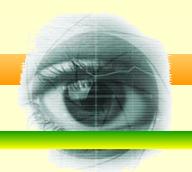
0	1	1
-1	0	1
-1	-1	0

D₄: SE-NO

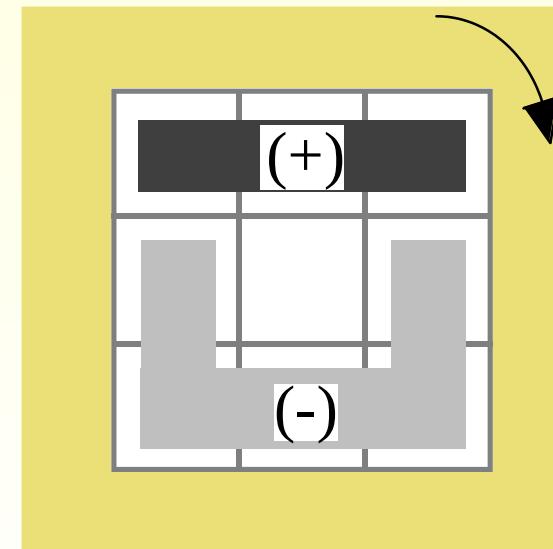
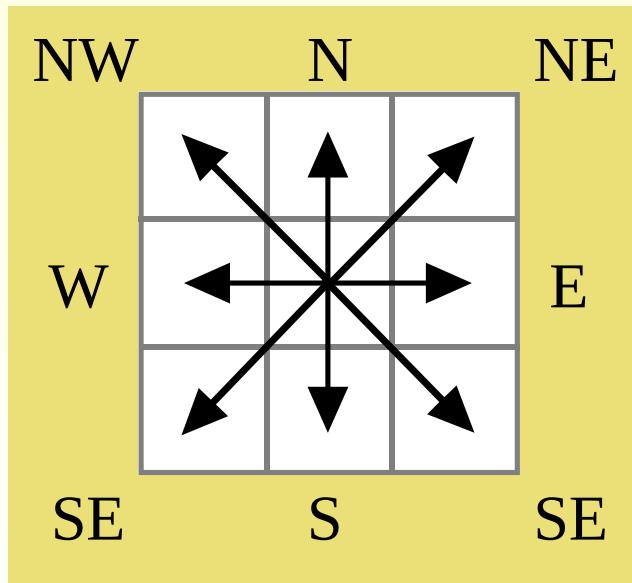


Extracción de Fronteras

Máscaras de Brújula



Usar 8 máscaras alineadas con las direcciones usuales de una brújula
Seleccionar la mayor respuesta (magnitud)
La orientación es la dirección asociada a la mayor respuesta.





Extracción de Fronteras

Muchos Kernels diferentes

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

Prewitt 1

5	5	5
-3	0	-3
-3	-3	-3

Kirsch

-1	$-\sqrt{2}$	-1
0	0	0
1	$\sqrt{2}$	1

Frei & Chen

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

Prewitt 2

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

Sobel



Extracción de Fronteras

Máscaras de Brújula de Sobel

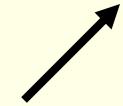


-1	0	1
-2	0	2
-1	0	1

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

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

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

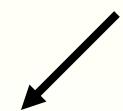


1	0	-1
2	0	-2
1	0	-1

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

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

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





Extracción de Fronteras

Máscaras de brújula de Kirsch

Kirsch

En cada punto se toma el valor máximo de los 8 filtros.

-3	-3	5
-3	0	5
-3	-3	5

0°

-3	5	5
-3	0	5
-3	-3	-3

45°

5	5	5
-3	0	-3
-3	-3	-3

90°

5	5	-3
5	0	-3
-3	-3	-3

135°

5	-3	-3
5	0	-3
5	-3	-3

180°

-3	-3	-3
5	0	-3
5	5	-3

225°

-3	-3	-3
-3	0	-3
5	5	5

270°

-3	-3	-3
-3	0	5
-3	5	5

315°

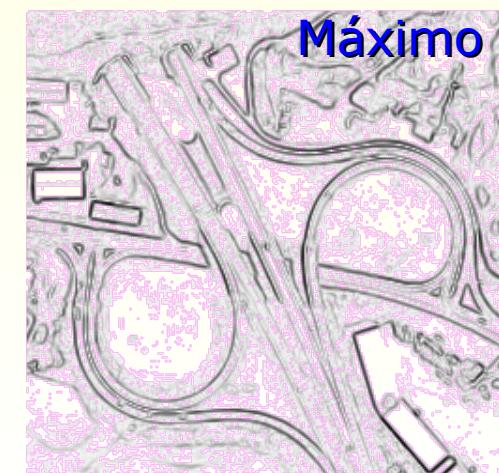
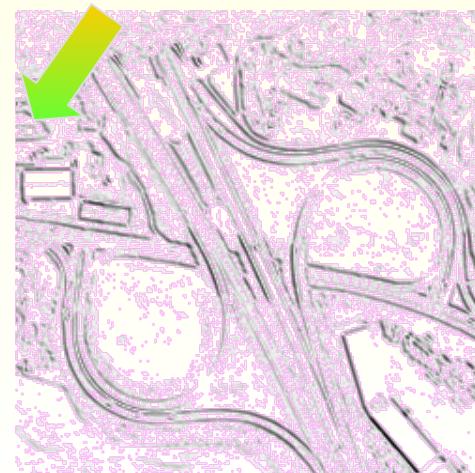
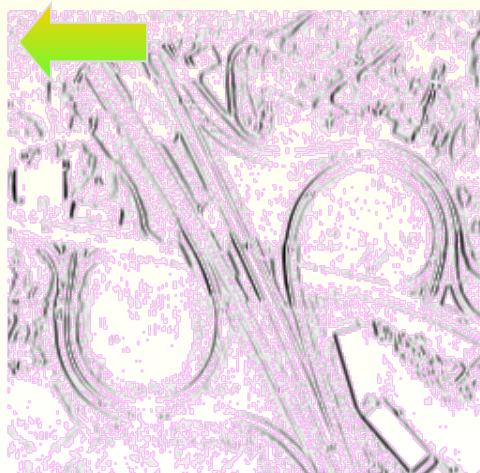
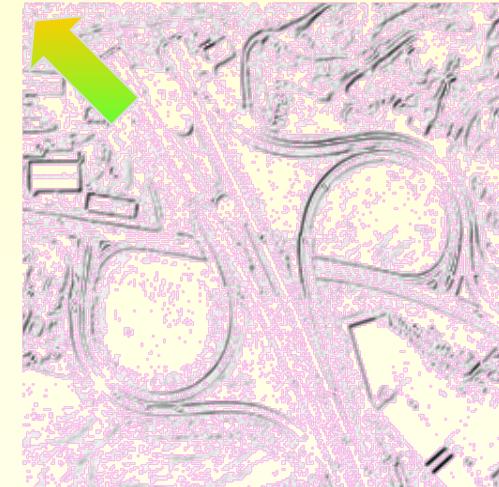
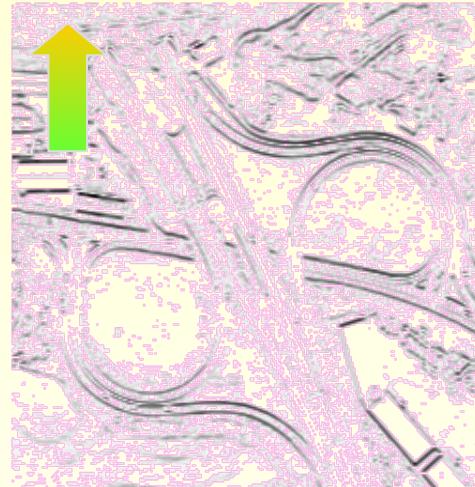


Extracción de Fronteras

Otros filtros para calcular el gradiente



Kirsch (usando sólo 4 orientaciones)





Resumen:

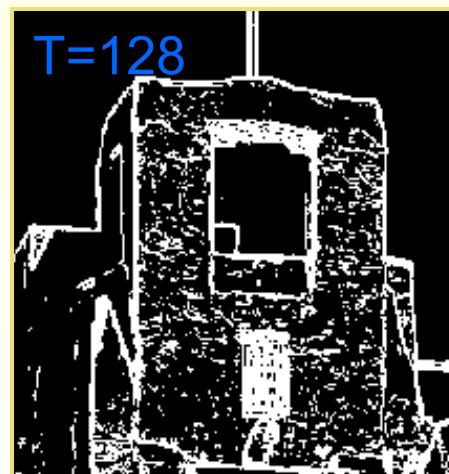
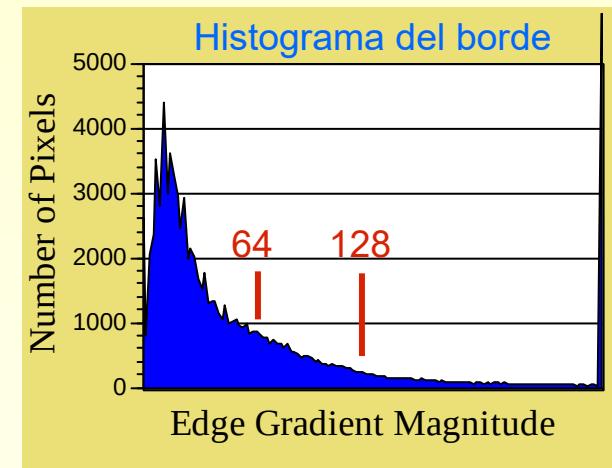
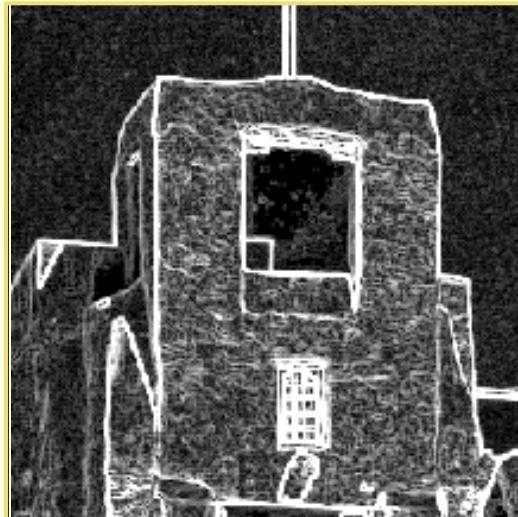
- Usualmente, los operadores gradiente 3×3 funcionan mejor que los 2×2 .
- Prewitt y Sobel funcionan mejor que cualesquiera otros operadores 3×3 de estimación del gradiente.
- En determinadas situaciones, pueden usarse operadores de estimación de gradiente de un tamaño mayor de 3×3 para mejorar el resultado.
- En máscaras grandes, es beneficioso ponderar por la distancia desde el pixel central.



Extracción de Fronteras Umbralización del borde



■ Aproximación global



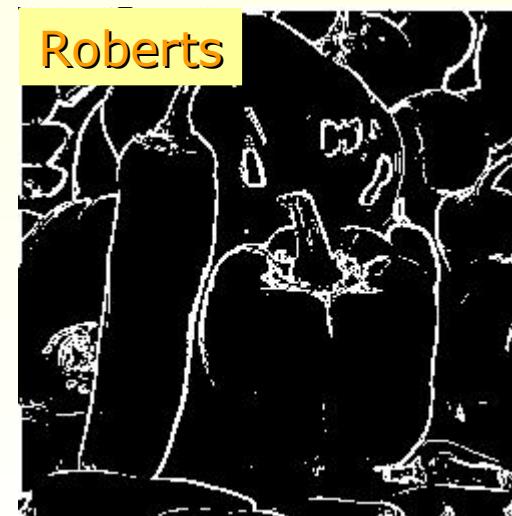
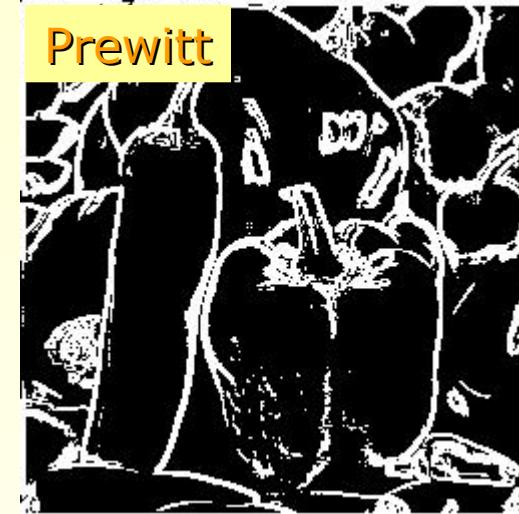


Extracción de Fronteras

Umbralización de la magnitud



Más ejemplos de umbralizaciones a partir de la magnitud del gradiente para obtener los puntos frontera.

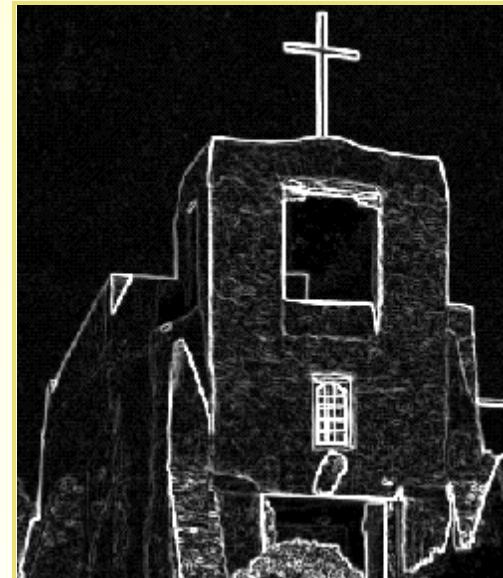
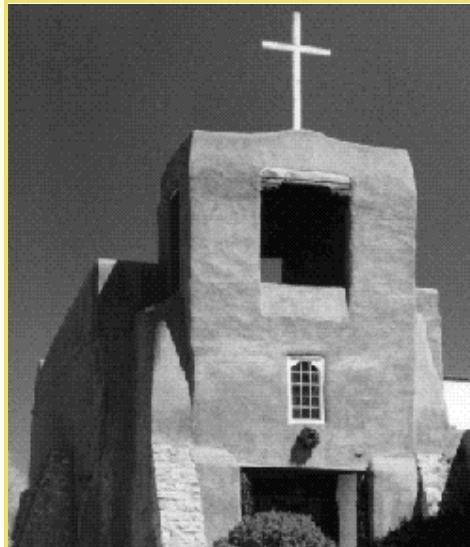




Extracción de Fronteras Supresión de No-Máximos



- Las máscaras grandes, gradientes locales, y píxeles mezclados pueden causar respuestas múltiples de la máscara para el mismo borde:



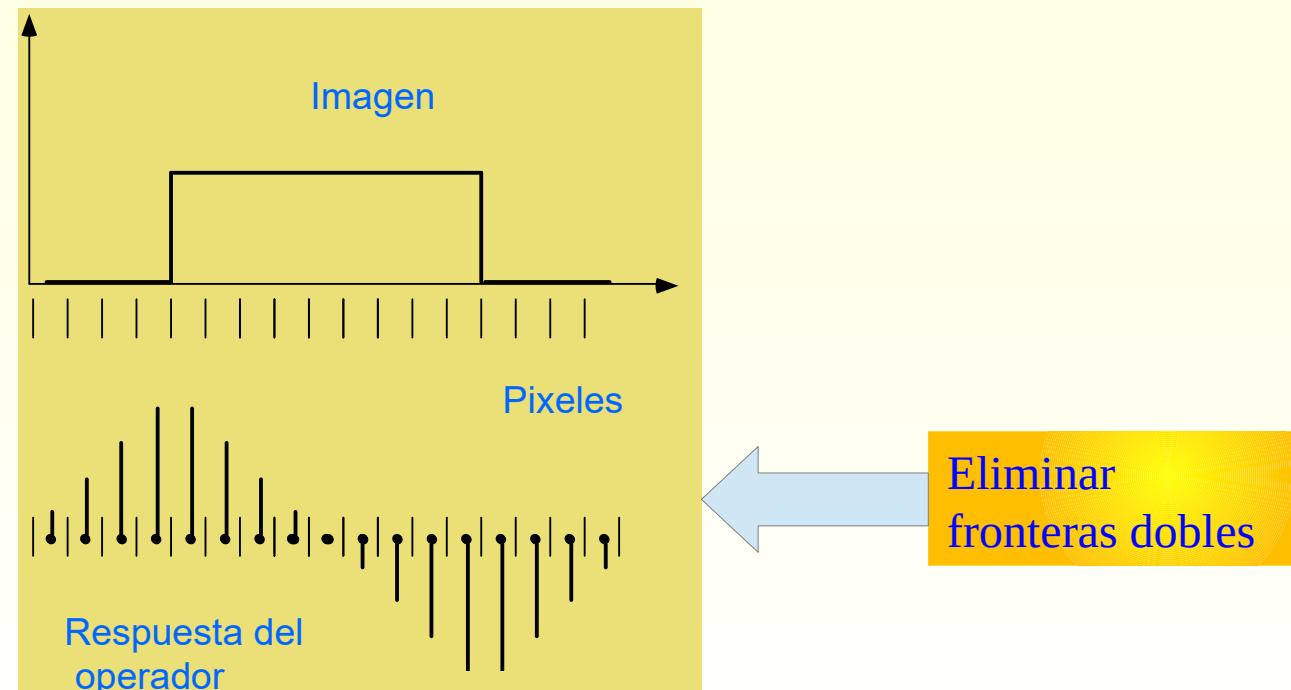
- ¿Podemos reducir el problema eliminando algunos de los bordes duplicados?



Extracción de Fronteras Supresión de No-Máximos

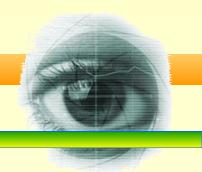


- **OBJETIVO:** conservar el mejor ajuste a un borde eliminando bordes redundantes sobre la base de un análisis local.
- Consideraremos un caso 1-D y un operador de tamaño (anchura) 9: {-1 -1 -1 -1 0 1 1 1 1}



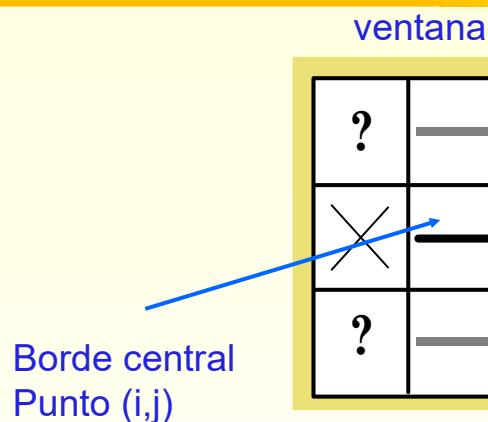


Extracción de Fronteras Supresión de No-Máximos



■ Algoritmo:

1. En paralelo, en cada pixel de la imagen de bordes, aplicar la selección de una ventana W como una función de la orientación del borde:



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



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



$E(i,j)$ la magnitud del gradiente en (i,j) que se obtenido al convolucionar la imagen en la dirección x del gradiente, siendo este el máximo. La arista posible tiene orientación vertical.



Extracción de Fronteras Supresión de No-Máximos



2. No considerar en el futuro aquellos puntos (n,m) en la ventana W cumpliendo

$(n,m) \in W, (n,m) \neq (i,j)$ para los cuales:

$\text{sign } E(n,m) \neq \text{sign } E(i,j) \longrightarrow \theta(n,m) \neq \theta(i,j)$

{diferentes direcciones
de gradiente} {diferentes
orientaciones del borde}

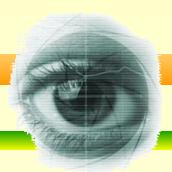
3.-De los restantes bordes, hacer $E(i,j) = 0$ si, para algún $(n,m) \in W$,
 $|E(n,m)| > |E(i,j)|$

4.-Aplicar una umbralización convencional a los bordes si se estima oportuno.

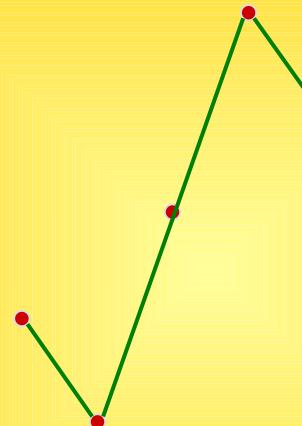
Hay muchas variaciones sobre el algoritmo básico.



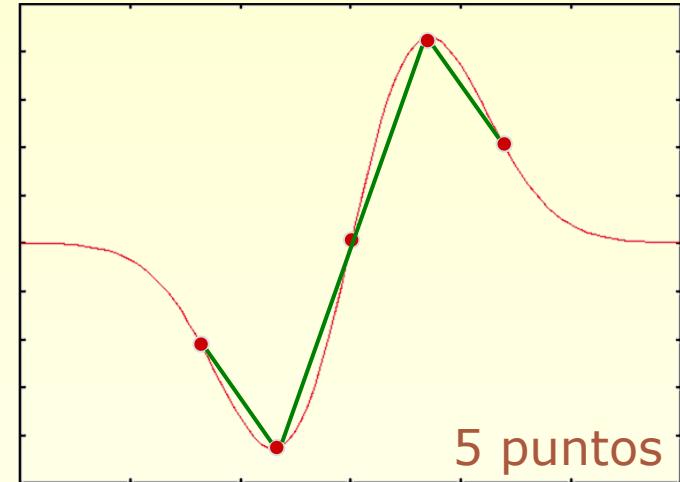
Otros filtros para calcular el gradiente



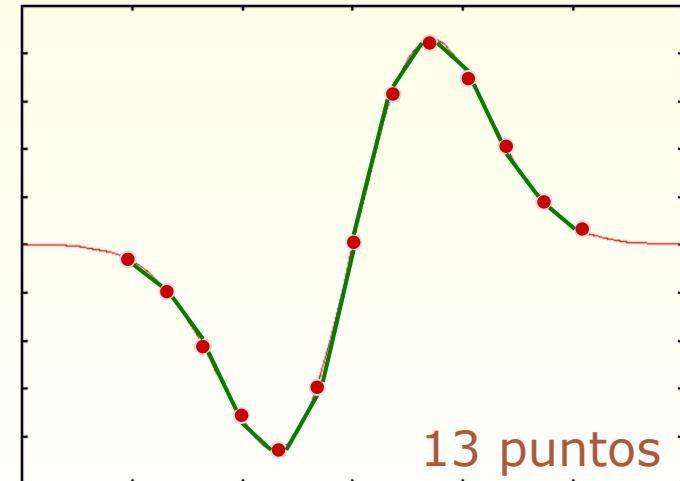
El perfil 1D de los filtros gradiente siempre es de la forma:



Es una media que pondera de manera negativa a un lado del centro y positiva al otro (antisimétrica).



5 puntos



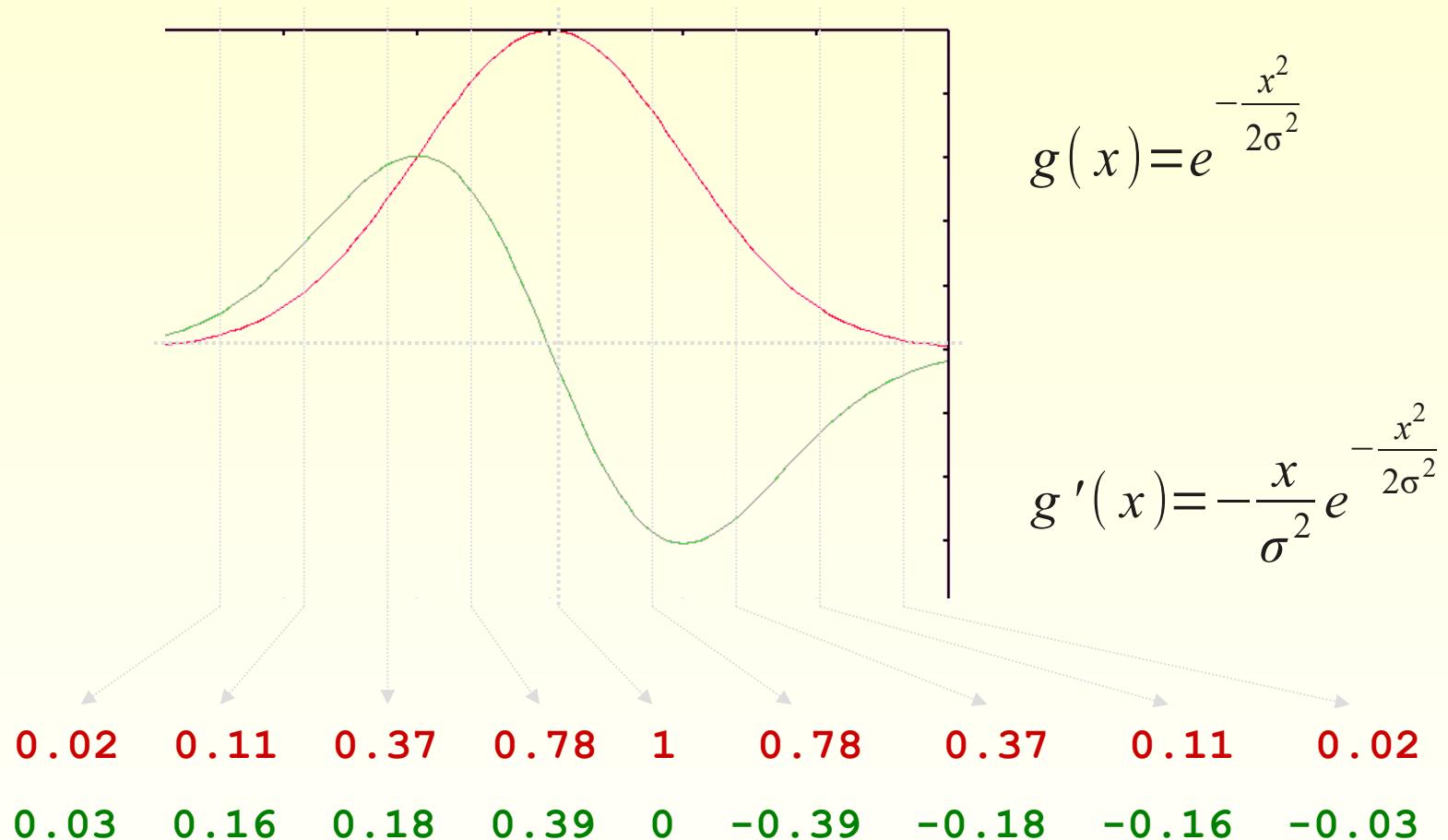
13 puntos



Otros filtros para calcular el gradiente



Derivada de gaussiana (DoG)



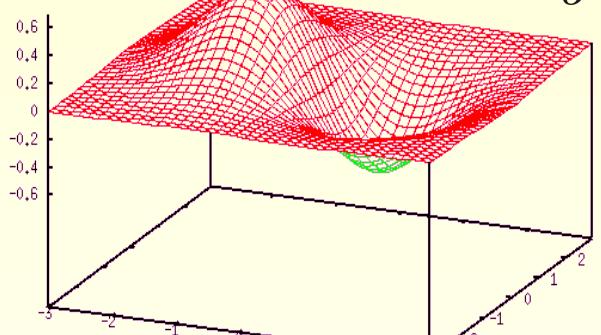
Si muestreamos las funciones obtenemos un kernel.



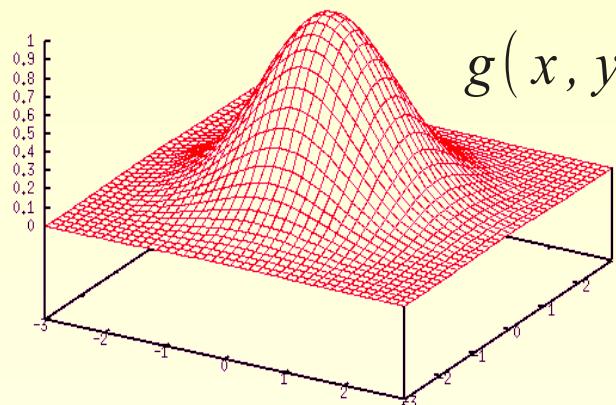
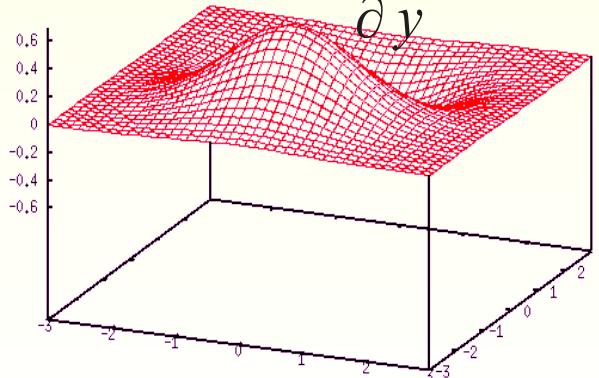
Otros filtros para calcular el gradiente

Derivada de gaussiana 2D

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



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



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

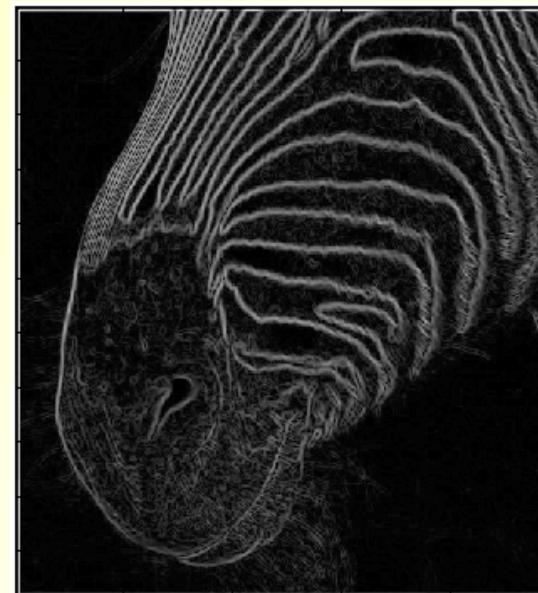
Cuanto mayor sea σ mayor es la gaussiana y más suaviza la imagen.

- + suavizado → - ruido
- + suavizado → - localización

Gaussiana: optimiza la localización y la detección simultáneamente.

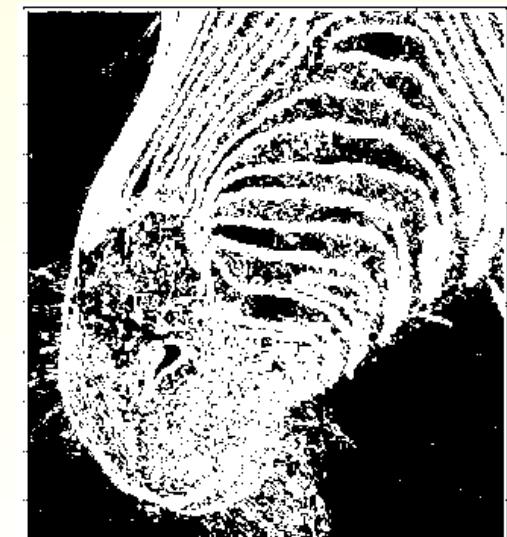


Extracción de Fronteras Umbralización de la magnitud



Derivada de gaussiana
 $\sigma=1$

Ejemplo de tres posibles
umbralizaciones.



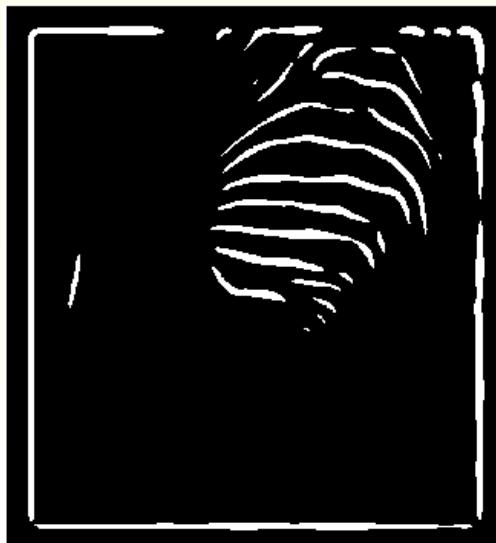


Extracción de Fronteras Umbralización de la magnitud



Derivada de gaussiana
 $\sigma=5$

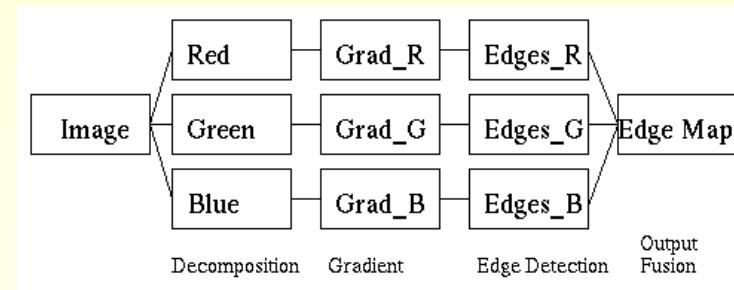
Ejemplo de tres posibles
umbralizaciones.



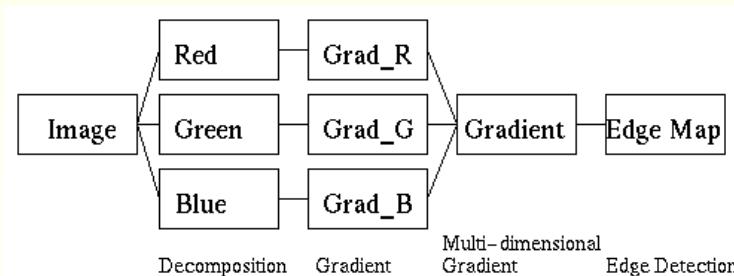


Aproximaciones típicas

- Fusión de resultados sobre las bandas R, G, B separadamente

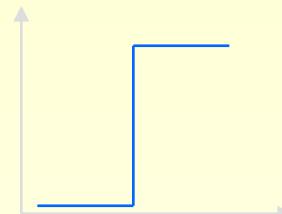


- Métodos de gradiente multi-dimensional

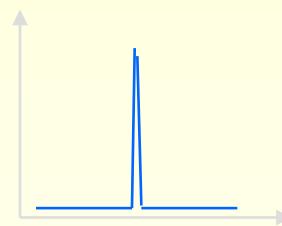




$f(x)$ = borde tipo salto



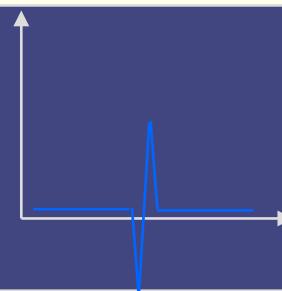
1^a Derivada $f'(x)$



máximo

METODOS
DE
GRADIENTE

2^a Derivada $f''(x)$

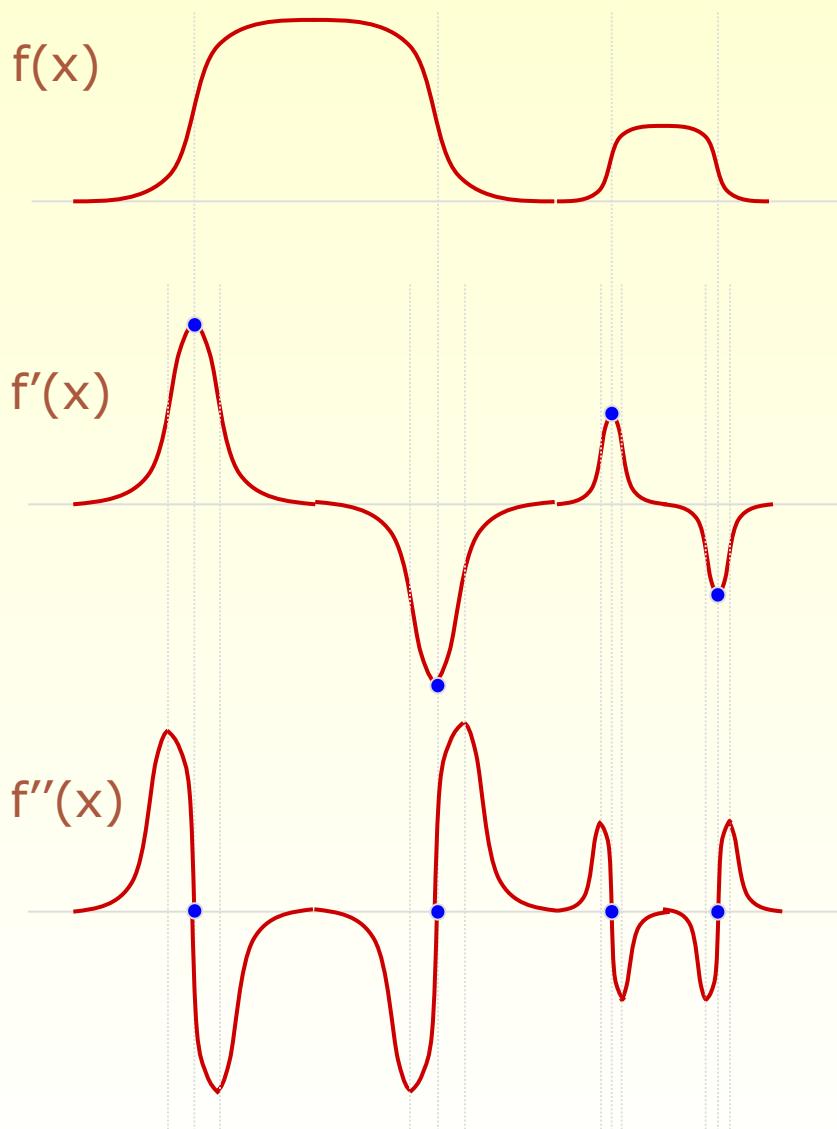


Cruces por cero
(zero-crossings)

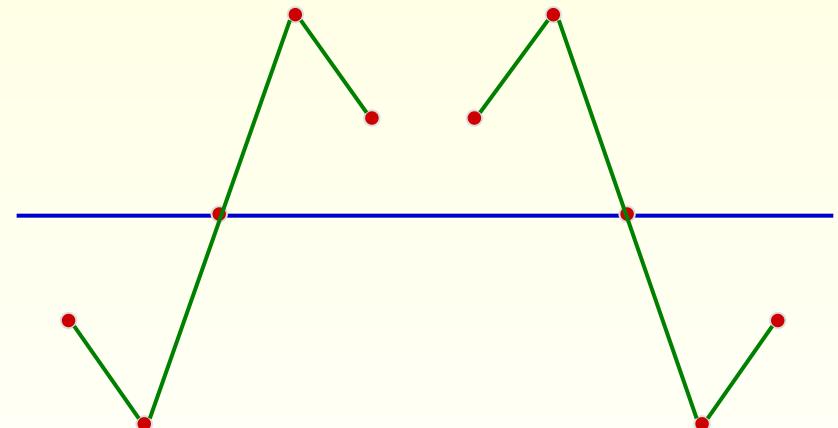


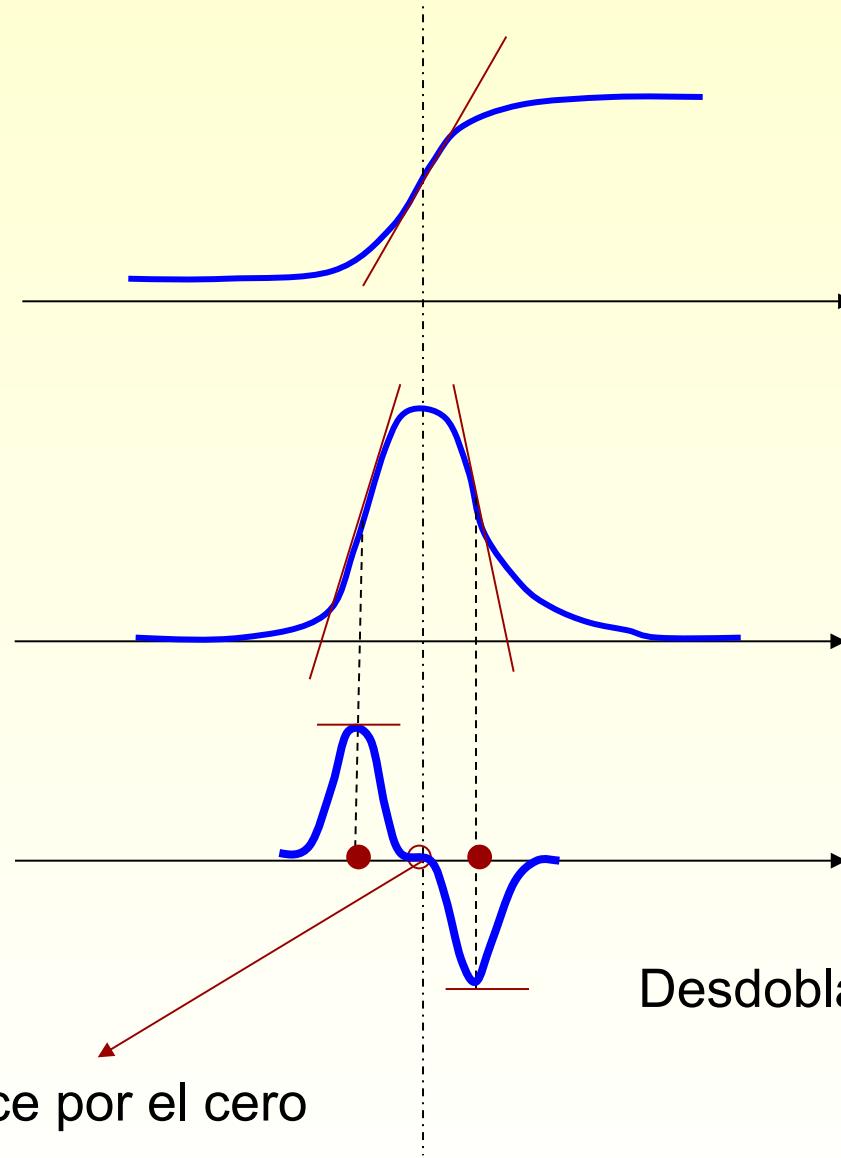
Extracción de Fronteras

La segunda derivada



En el caso de la segunda derivada, las fronteras vienen dadas por los cruces por cero de la misma.





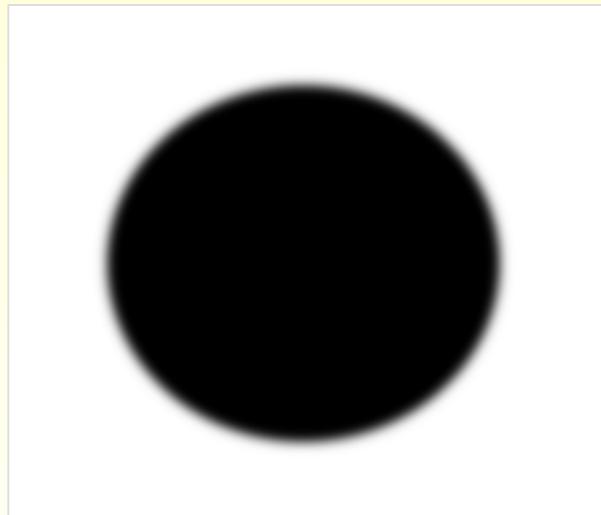


Extracción de Fronteras

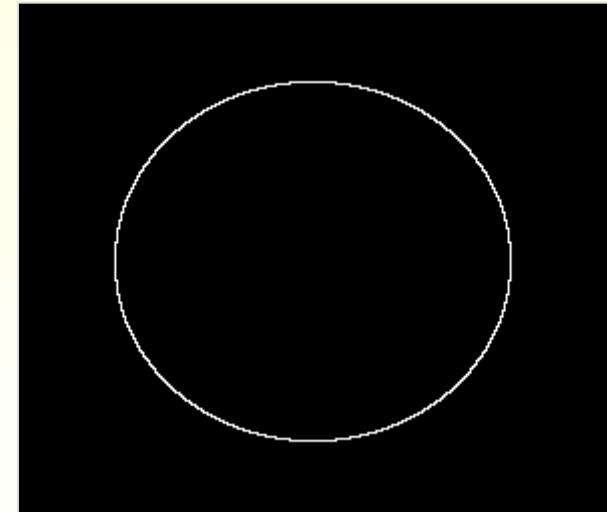
La segunda derivada



Valor absoluto de un filtro segunda derivada
(laplaciana)



Original



Cruces por cero



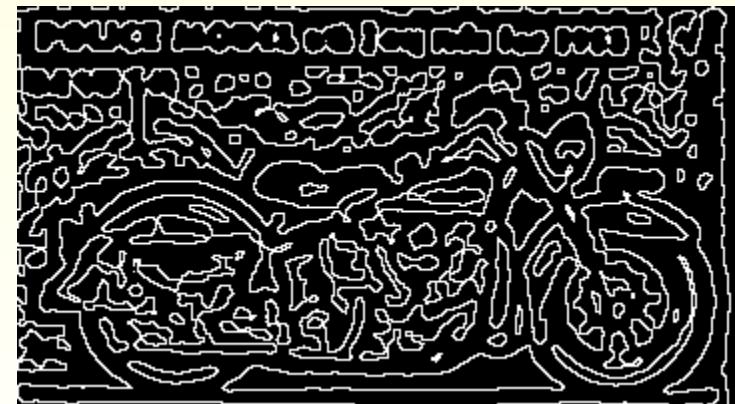
Extracción de Fronteras La segunda derivada



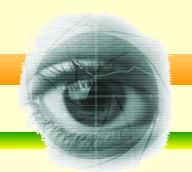
Valor absoluto de un filtro segunda derivada
(laplaciana)



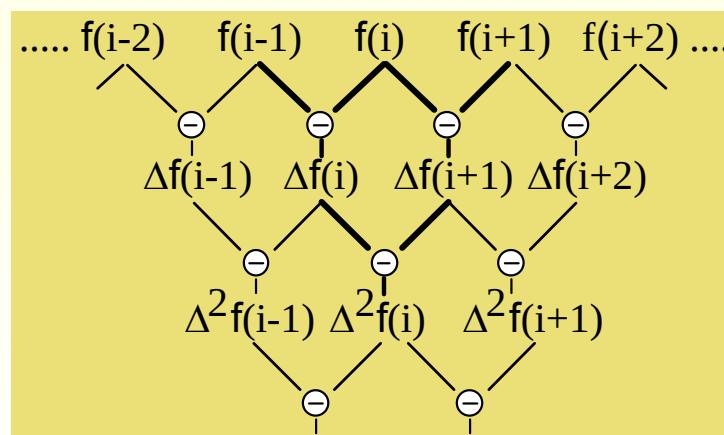
Original



Cruces por cero



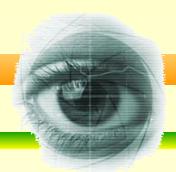
- Segunda derivada = nivel de cambio de la primera derivada.
- Máximos de la primera derivada = cruces por cero de la segunda derivada.
- Para una función discreta, las derivadas pueden aproximarse por diferencias.
- Considerar el caso unidimensional:



$$\begin{aligned}\Delta^2 f(i) &= \Delta f(i+1) - \Delta f(i) \\ &= f(i+1) - 2f(i) + f(i-1)\end{aligned}$$

Máscara:

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



- Consideremos ahora una función 2-D $f(x,y)$.
- Las segundas derivadas parciales $f(x,y)$ no son isotrópicas.
- Puede demostrarse que el operador segunda derivada isotrópico menor posible es el laplaciano.

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

- La aproximación bidimensional discreta es:

	1	
1	-4	1
	1	



Extracción de Fronteras

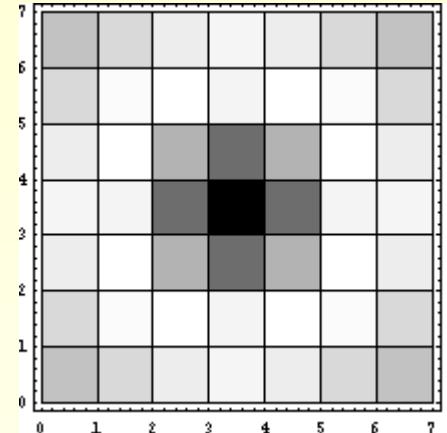
La segunda derivada



0	1	0
1	-4	1
0	1	0

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

Es isotrópico para rotaciones de 90°



Otros filtros laplacianos:

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

1	1	1
1	-8	1
1	1	1

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

“Diferencia entre el píxel central y la media
de sus vecinos...”



Extracción de Fronteras

La segunda derivada



Ejemplos de máscaras de Laplace.

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

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

“Diferencia entre el
píxel central y la
media de sus
vecinos...”

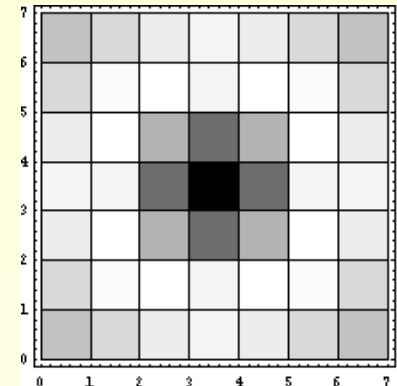


Imagen de entrada



Laplaciana 2 (3x3)

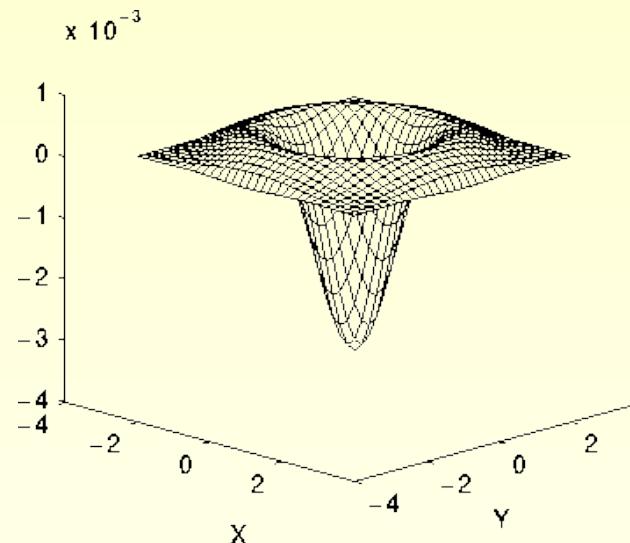


Extracción de Fronteras

La segunda derivada



Versión
contínua:



Versiones discretas

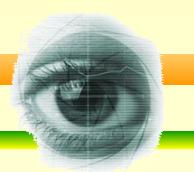
0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1



Extracción de Fronteras

La segunda derivada



Laplaciano 3×3



Laplaciano 5×5



Laplaciano 7×7



Detección de
cruces por cero



Detección de
cruces por cero

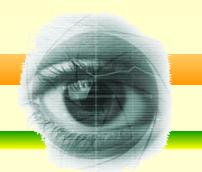


Detección de
cruces por cero



Extracción de Fronteras

La segunda derivada



Podemos considerar entornos mayores para el cálculo:

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	24	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

5x5

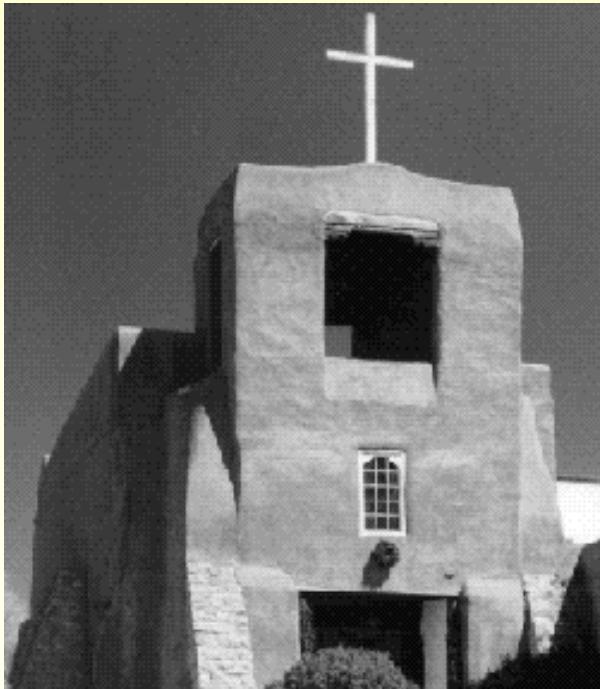
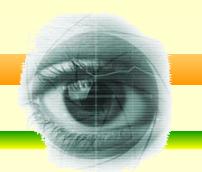
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	8	8	8	-1	-1	-1
-1	-1	-1	8	8	8	-1	-1	-1
-1	-1	-1	8	8	8	-1	-1	-1
-1	-1	-1	8	8	8	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

9x9



Extracción de Fronteras

La segunda derivada



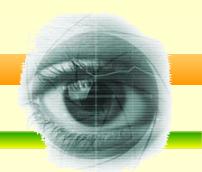
Al aumentar el tamaño:
→ Perdemos detalle
→ Menos sensible al ruido



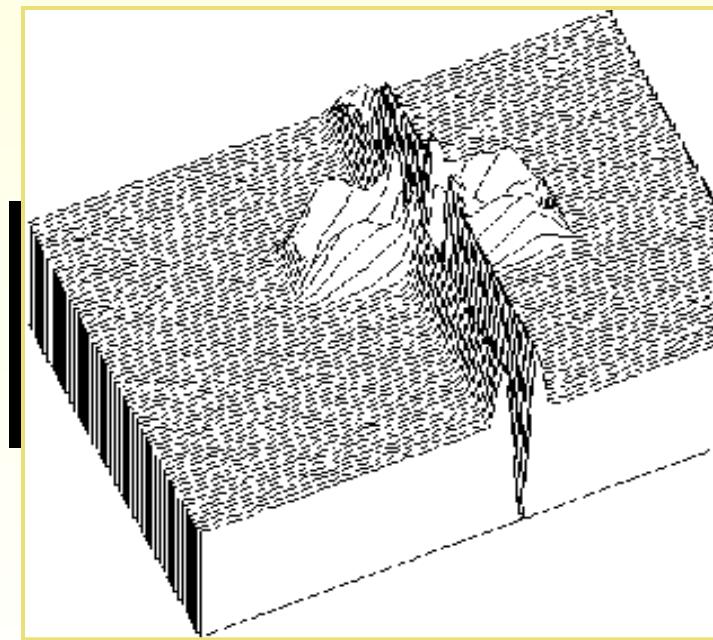
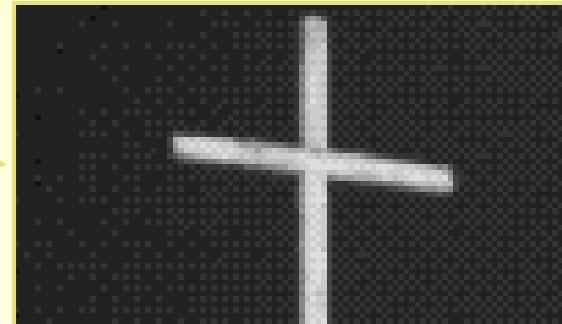
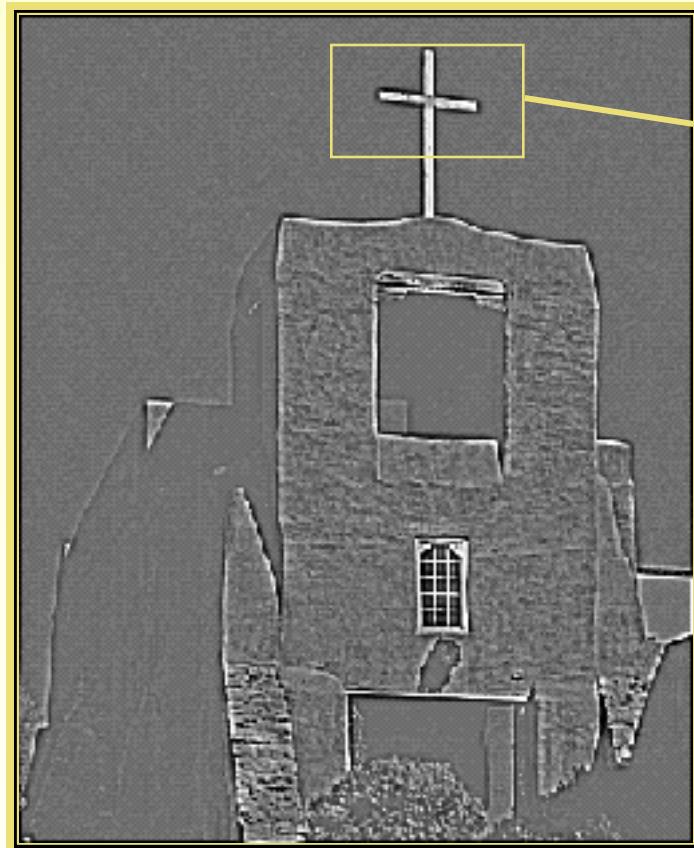


Extracción de Fronteras

La segunda derivada



Vista detallada de los resultados





Implementación de la Laplaciana

- Considerar la definición de la Laplaciana discreta:

$$\nabla^2 I = \frac{I(i+1,j) + I(i-1,j) + I(i,j+1) + I(i,j-1) - 4I(i,j)}{ }$$

Parece una suma en una ventana

- Reescribirla como: $\nabla^2 I = I(i+1,j) + I(i-1,j) + I(i,j+1) + I(i,j-1) + I(i,j) - 5I(i,j)$
- Sacar factor común -5 y obtener: $\nabla^2 I = -5(I(i,j) - \mu)$
- El Laplaciano puede obtenerse (dejando al margen la constante -5), restando el valor medio alrededor del punto (i,j) del valor de la imagen en ese (i,j) !



Realce de imágenes usando la laplaciana

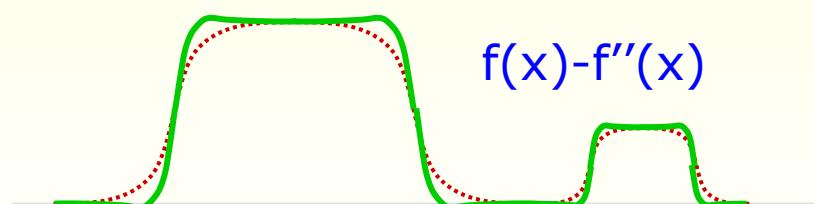
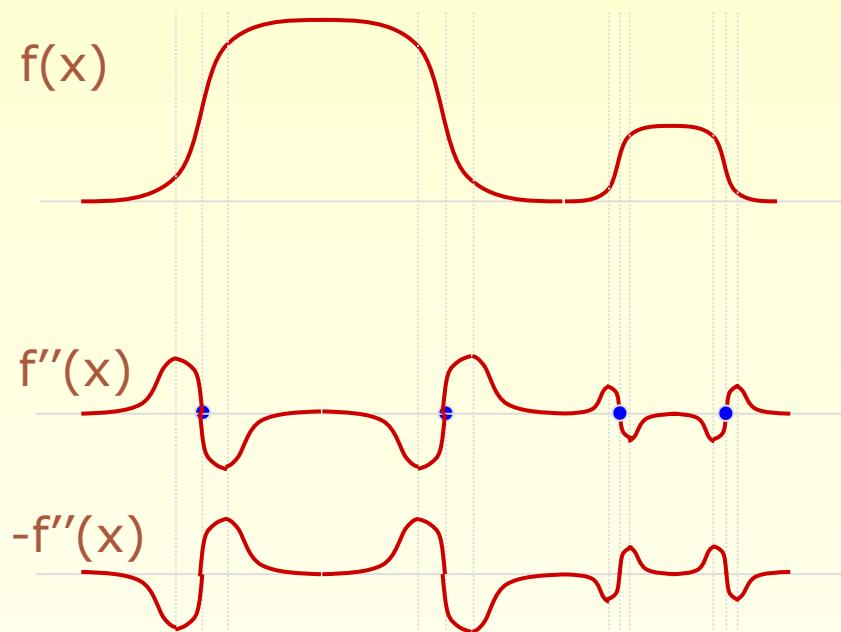
- La Laplaciana puede usarse para realzar imágenes:

$$\begin{aligned} I(i,j) - \nabla^2 I(i,j) = \\ 5 I(i,j) \\ -[I(i+1,j) + I(i-1,j) + I(i,j+1) + I(i,j-1)] \end{aligned}$$

- Si (i,j) está en medio de una región plana o de una “rampa” grande: $I - \nabla^2 I = I$
- Si (I,j) está en el extremo inferior (o superior) de una rampa o borde: $I - \nabla^2 I < I$ (ó $I - \nabla^2 I > I$)
- El efecto es el de eliminar “emborronamiento” en la imagen.



Realce de imágenes usando la laplaciana



Se realizan las fronteras:
→ La imagen es más nítida
→ La imagen es menos suave.



Realce de imágenes usando la laplaciana



Original

Laplaciana

Original - Laplaciana



Realce de imágenes usando la laplaciana





Realce de imágenes usando la laplaciana

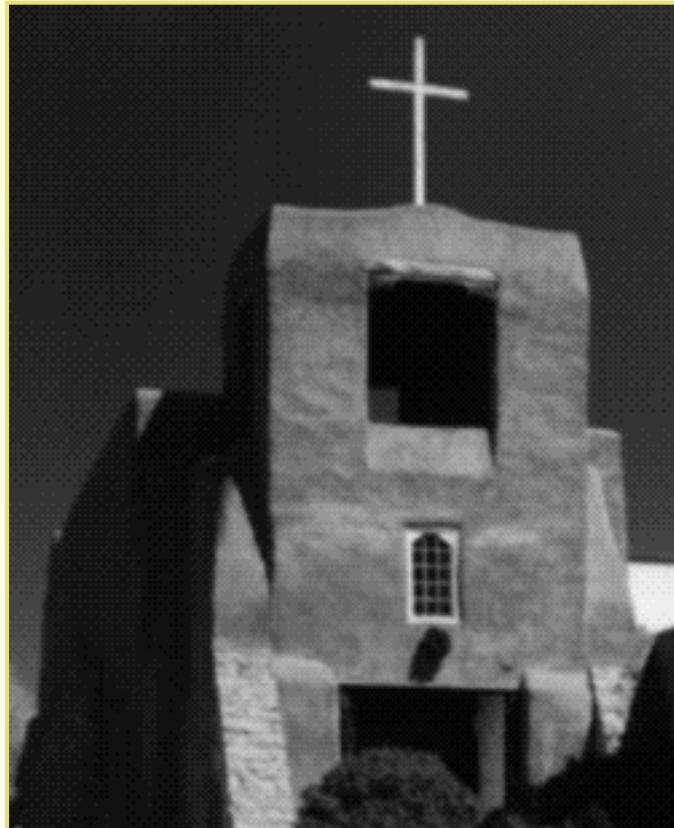


Imagen Original Emborronada

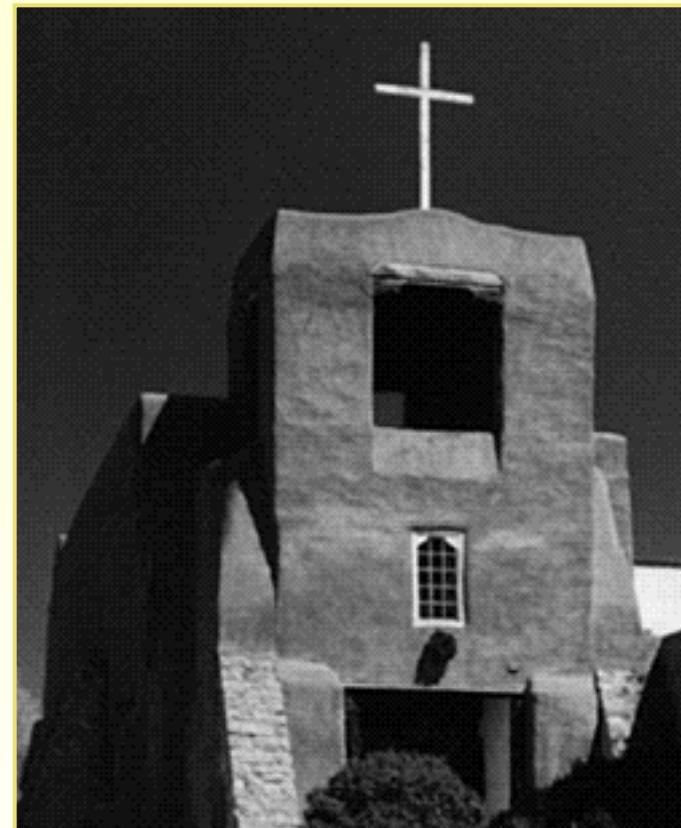


Imagen realzada con un
Laplaciano 3x3



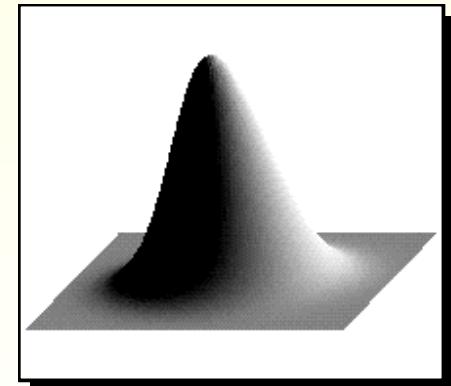
Segunda Derivada y el Ruido

- La segunda derivada, al igual que la primera, realza el ruido.
- Combinar el operador segunda derivada con un operador de alisamiento.

Idea:

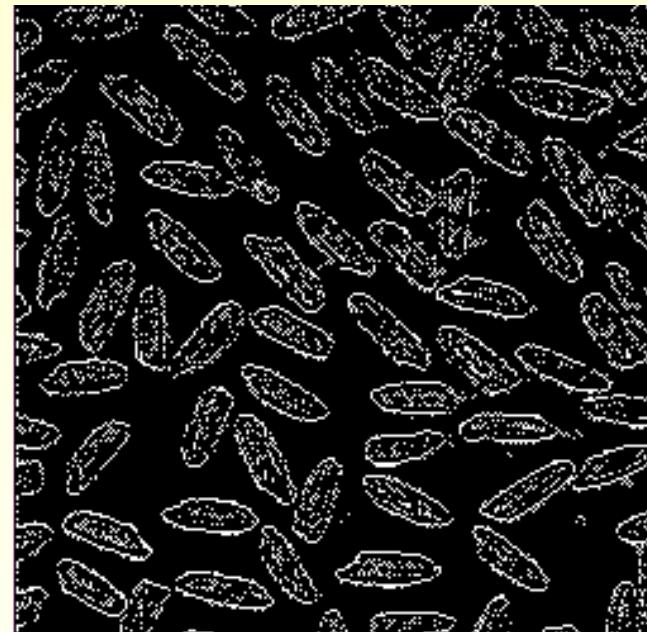
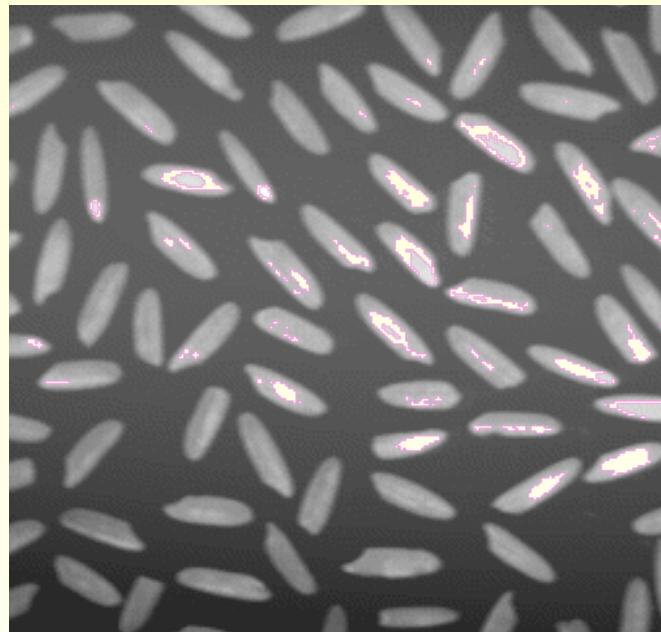
- **Alisar** la imagen.
- Calcular la **segunda derivada** de la imagen (2D),
- Encontrar los píxeles donde la función “cruza” el 0 y marcarlos
- Cuestiones:
 - Naturaleza del filtro de alisamiento óptimal.
 - Cómo detectar cambios de intensidad a una escala dada.
 - Cómo combinar la información a lo largo de múltiples escalas.

El operador de alisamiento que suele usarse por sus propiedades es la Gaussiana





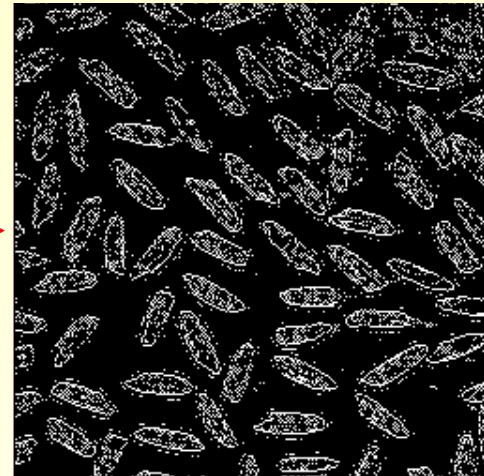
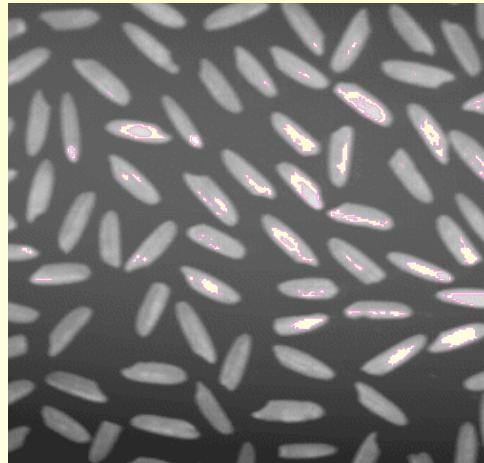
La segunda derivada



Bordes: Cruce por el cero

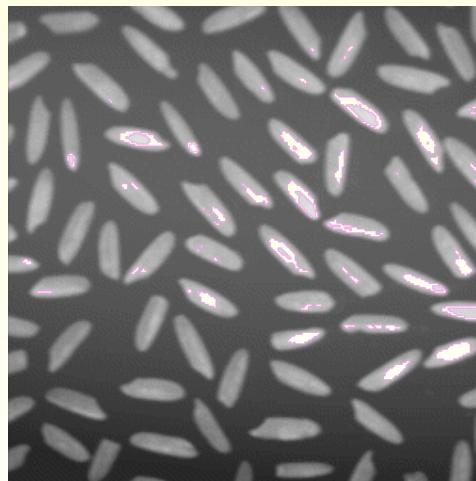


La segunda derivada



Bordes: Cambios de signo

Sensible al ruido



Bordes: Cambios de signo

Imagen suavizada con
un filtro gaussiano



Justificación del uso de una gaussiana

- La Gaussiana no es la única elección para eliminar ruido, pero tiene un buen número de importantes propiedades
 - Si convolucionamos una Gaussiana con otra Gaussiana, el resultado es otra Gaussiana
 - ◆ Esto es llamado espacio de escalas lineal

$$G_{\sigma_1} * G_{\sigma_2} = G_{\sqrt{\sigma_1^2 + \sigma_2^2}}$$

- Eficiencia: separable



La segunda derivada

Filtro $\nabla^2 G$

- Aproximación de Marr y Hildreth:

1. Aplicar alisamiento Gaussiano usando gaussianas de tamaño creciente:

$$G \otimes I$$

2. Tomar la Laplaciana de las imágenes resultantes:

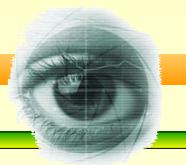
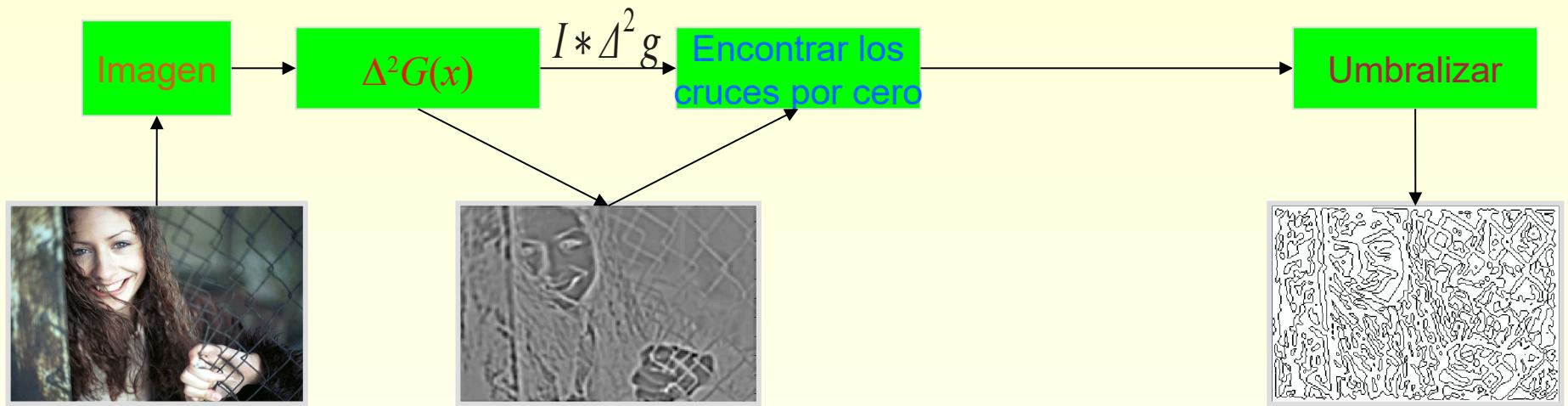
$$\nabla^2 (G \otimes I)$$

3. Buscar los cruces por cero.

- La segunda expresión puede reescribirse como:

$$(\nabla^2 G) \otimes I$$

- Por tanto, podemos tomar la Laplaciana de la Gaussiana y usarla como un operador.

Filtro $\nabla^2 G$. Esquema visual



■ Deducir la expresión de la Laplaciana de Gaussiana (LoG)

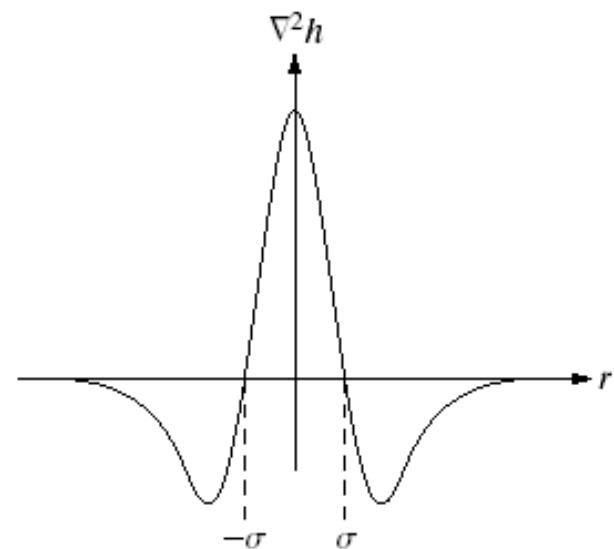
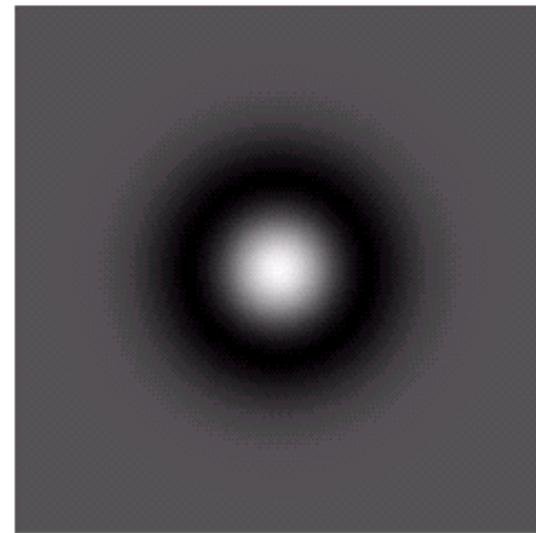
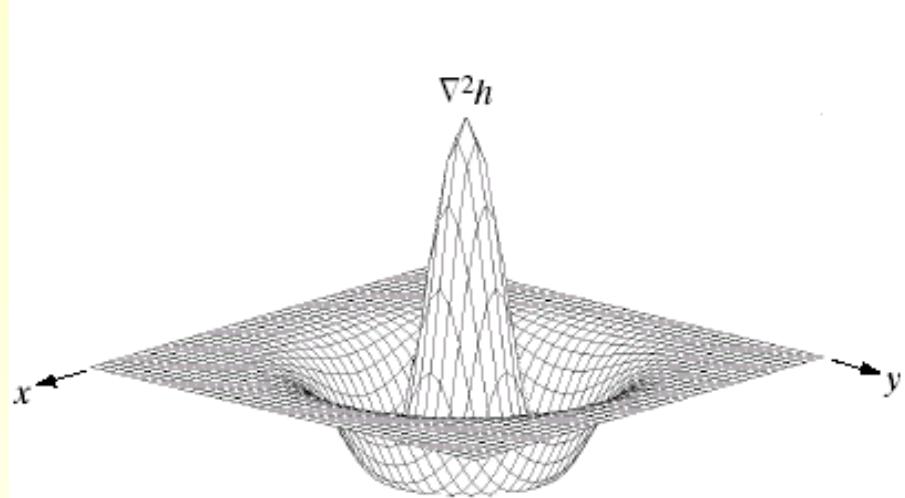
$$\Delta^2 S = \Delta^2(g * I) = (\Delta^2 g) * I$$

Ejercicio

$$\Delta^2 g = -\frac{1}{\sqrt{2\pi}\sigma^3} \left(2 - \frac{x^2 + y^2}{\sigma^2} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$



La segunda derivada



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

a
b
c
d

FIGURE 10.14
Laplacian of a Gaussian (LoG).
(a) 3-D plot.
(b) Image (black is negative, gray is the zero plane, and white is positive).
(c) Cross section showing zero crossings.
(d) 5×5 mask approximation to the shape of (a).



La segunda derivada

Kernels

5x5

$$\begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

17 x 17

0	0	0	0	0	0	-1	-1	-1	-1	0	0	0	0	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	0	0	0	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-2	-1	-1	0	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-2	-1	0	0
0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-2	-1	0	0
0	-1	-1	-2	-3	-3	-3	-2	-3	-2	-3	-3	-2	-1	0
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	0
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-1
-1	-1	-3	-3	-3	4	12	21	24	21	12	4	-3	-3	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-1
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-1
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	0
0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-3	-2	-1	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-2	-1	-1	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-2	-1	-1	-1	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	0	0	0	0



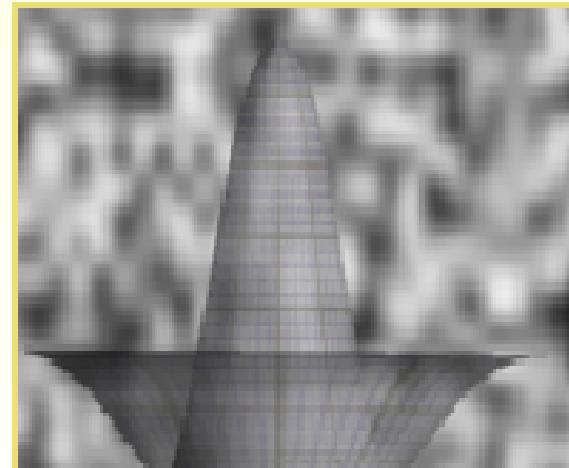
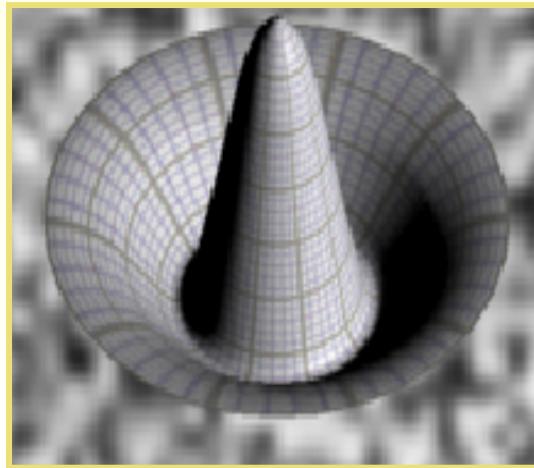
La segunda derivada

Filtro sombrero Mexicano

- Laplaciana de Gaussiana

$$\Delta^2 g = -\frac{1}{\sqrt{2\pi}\sigma^3} \left(2 - \frac{x^2 + y^2}{\sigma^2} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

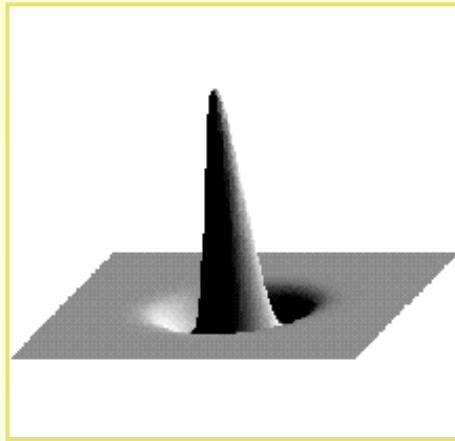
- $\nabla^2 G$ es un operador circularmente simétrico.
- También llamado sombrero u operador sombrero mexicano.



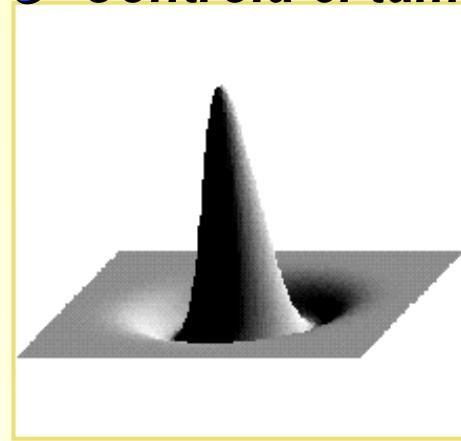


La segunda derivada

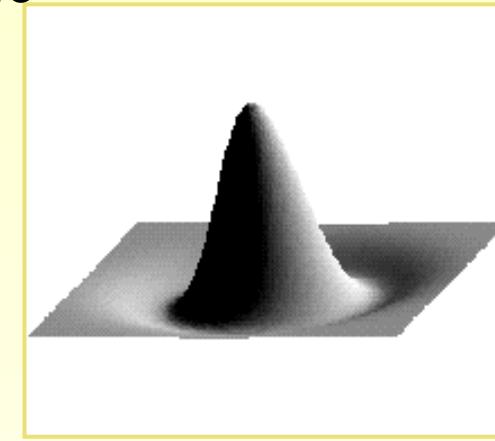
σ^2 Controla el tamaño



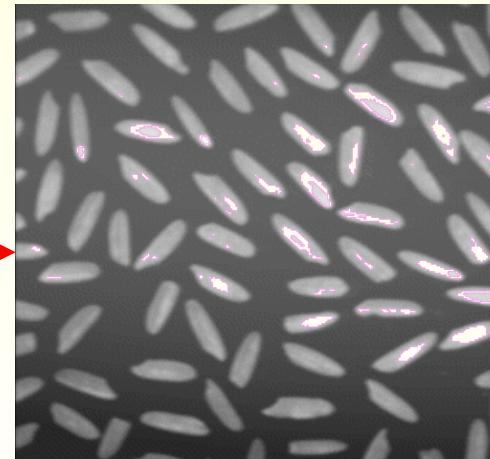
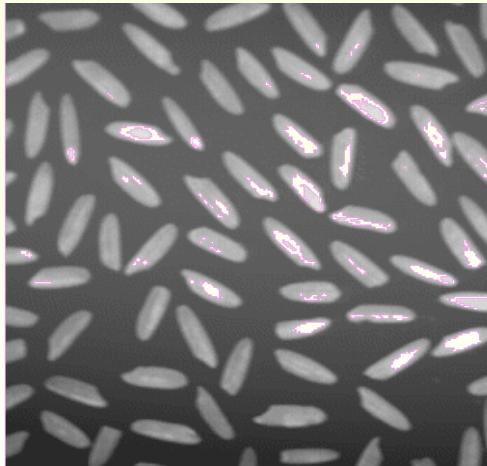
$$\sigma^2 = 0.5$$



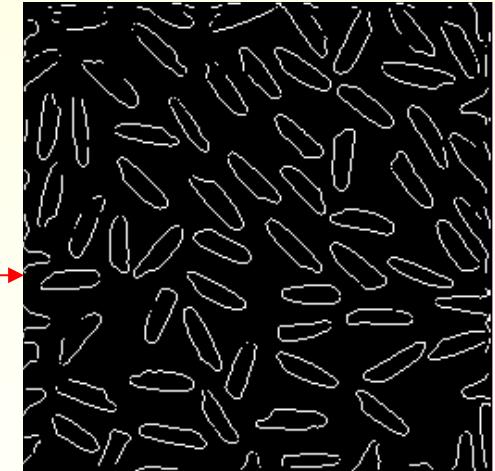
$$\sigma^2 = 1.0$$



$$\sigma^2 = 2.0$$



Filtro Gaussiano
 $\sigma = 2$



Filtro Laplaciano

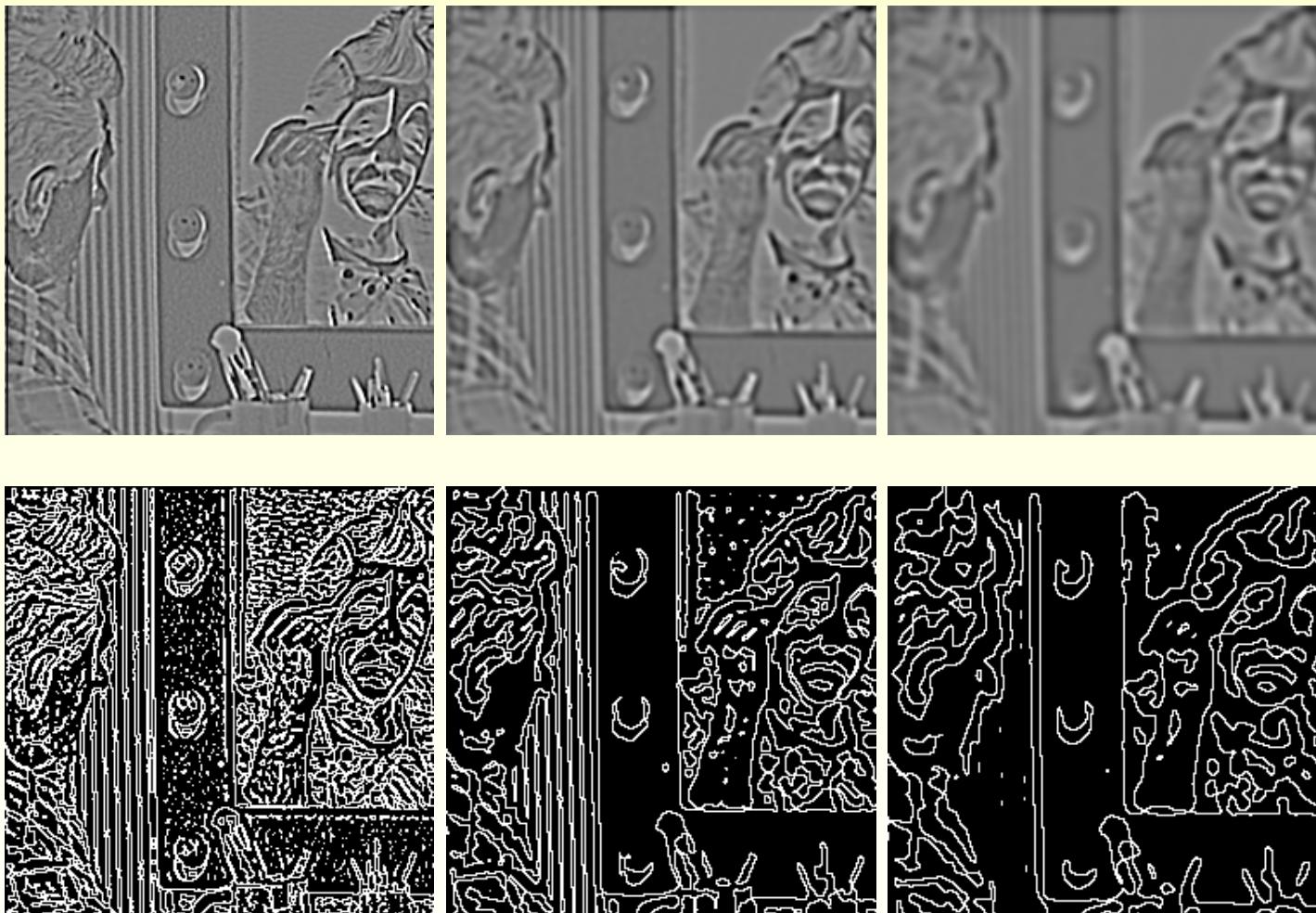


La segunda derivada

$\sigma=1$

$\sigma=2$

$\sigma=3$



Cruces por cero de la laplaciana



Extracción de Fronteras

Encontrando los cruces por cero

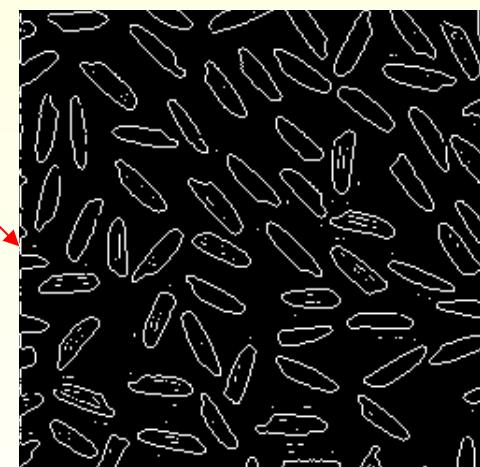
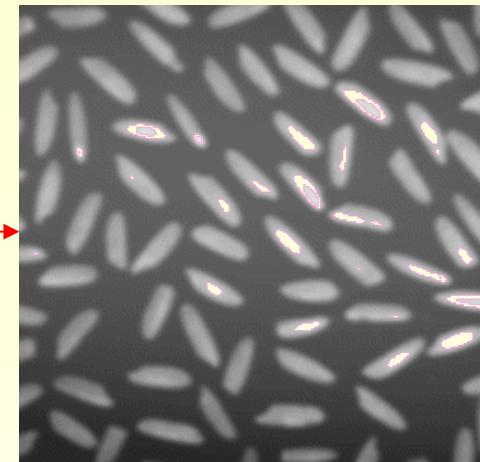
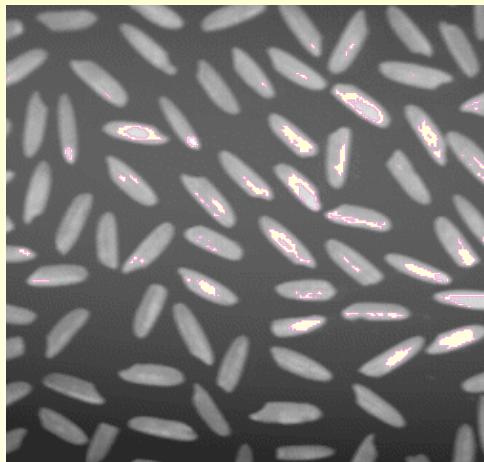
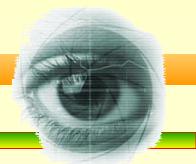


- Recorrer las filas de la imagen y almacenar como punto frontera cada localización de un cruce por cero. Repetirlo para las columnas.
- Casos de cruces por cero (zero-crossings):
 - $\{+, -\}$ $\{+, 0, -\}$
 - $\{-, +\}$ $\{-, 0, +\}$
- La pendiente del zero-crossing $\{a, -b\}$ es $|a+b|$.
- Para marcar una frontera
 - Calcular la pendiente del zero-crossing
 - Aplicar un umbral a la pendiente



Extracción de Fronteras

La segunda derivada



En blanco los valores positivos y
en negro los valores negativos



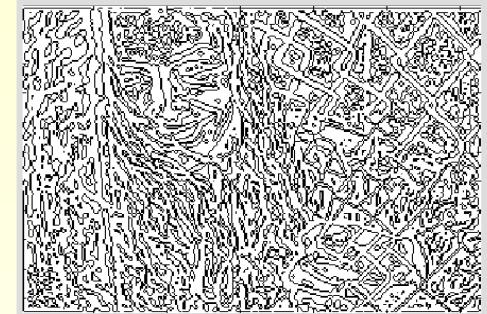
La segunda derivada

I

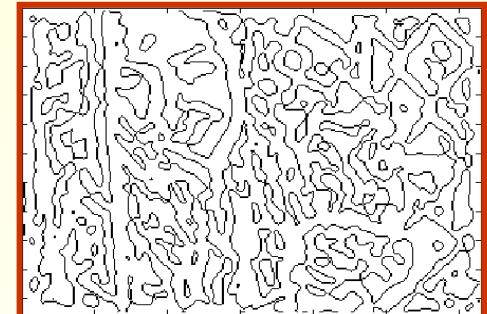
Ejemplo de uso del filtro LoG



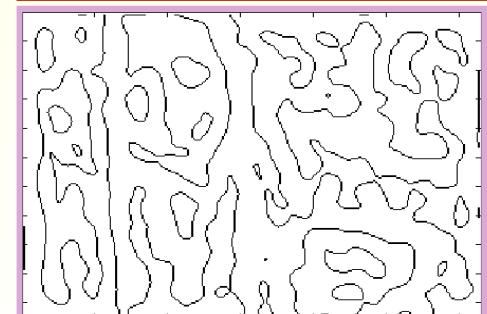
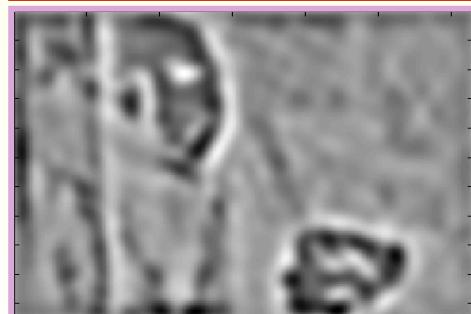
$$\sigma = 1$$



$$\sigma = 3$$



$$\sigma = 6$$





Ejemplo

La segunda derivada



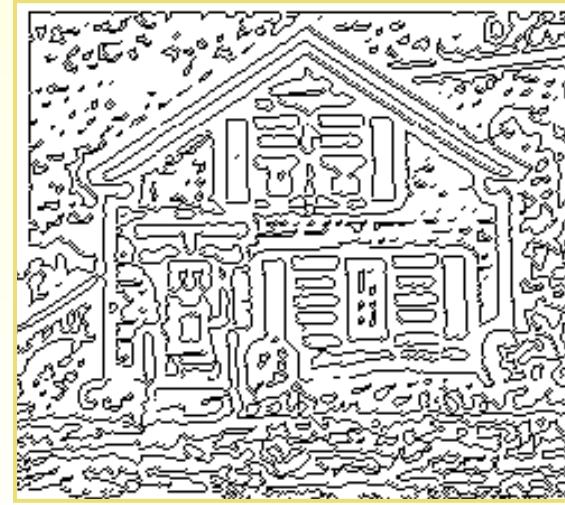
Filtro LoG 13x13



Umbralización Positiva



Umbralización Negativa

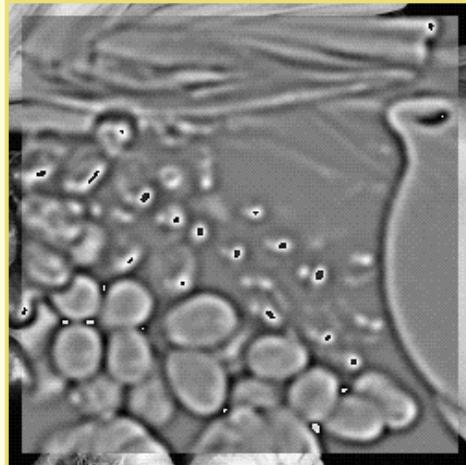


Cruces por cero



La segunda derivada

Otro ejemplo



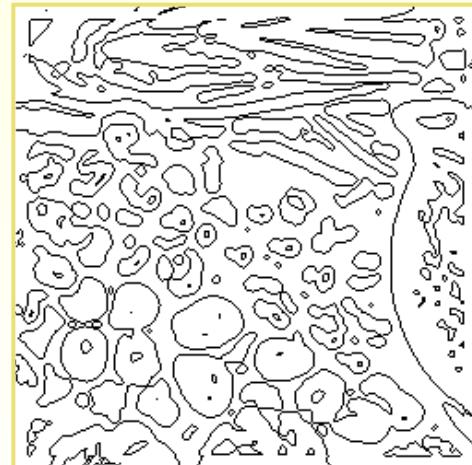
Filtro LoG 17x17



Umbralizada Positiva



Umbralizada Negativa



Cruces por cero



Separabilidad del filtro LoG derivada

- Similar a la separabilidad del filtro Gaussiano
 - Una Gaussiana 2D puede separarse en dos Gaussianas 1D.

$$h(x, y) = I(x, y) * g(x, y)$$

n^2 multiplicaciones

$$h(x, y) = (I(x, y) * g_1(x)) * g_2(y)$$

$2n$ multiplicaciones

$$g_1 = [0.011 \quad 0.13 \quad 0.6 \quad 1 \quad 0.6 \quad 0.13 \quad 0.011]$$

$$g_2 = \begin{bmatrix} 0.011 \\ 0.13 \\ 0.6 \\ 1 \\ 0.6 \\ 0.13 \\ 0.011 \end{bmatrix}$$

$$\Delta^2 S = \Delta^2(g * I) = (\Delta^2 g) * I = I * (\Delta^2 g)$$

Requiere n^2 multiplicaciones

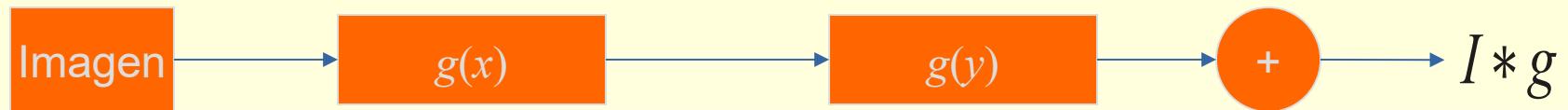
$$\Delta^2 S = (I * g_{xx}(x)) * g(x) + (I * g_{yy}(y)) * g(y)$$

Requiere $4n$ multiplicaciones

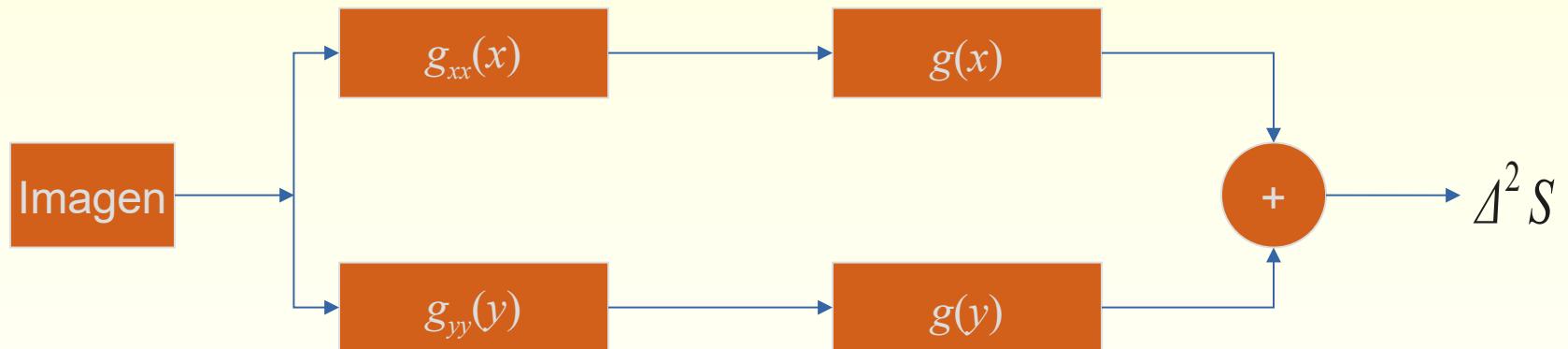


Separabilidad del filtro LoG derivada

Filtro Gaussiano

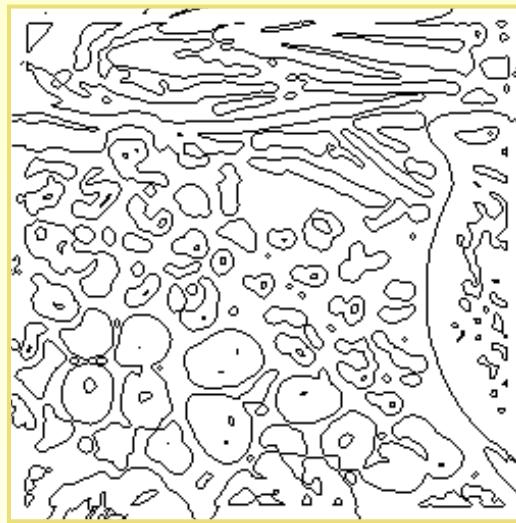


Filtro Laplaciana de Gaussiana

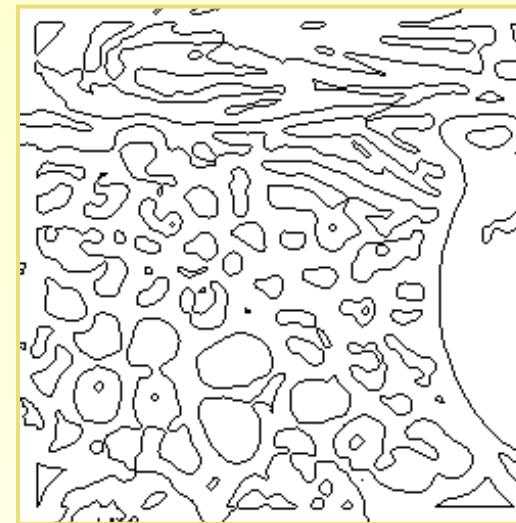




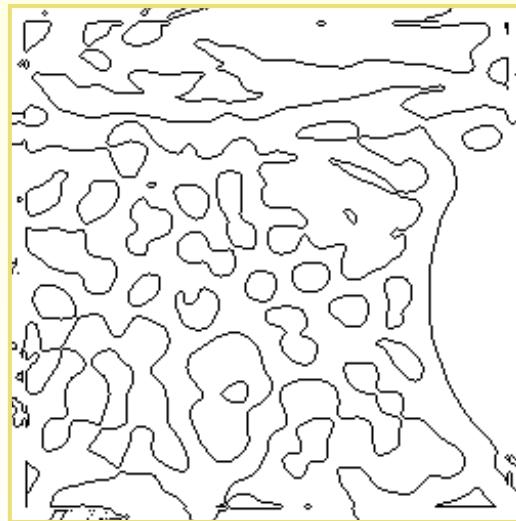
Espacio de Escalas: Multi-Resolución



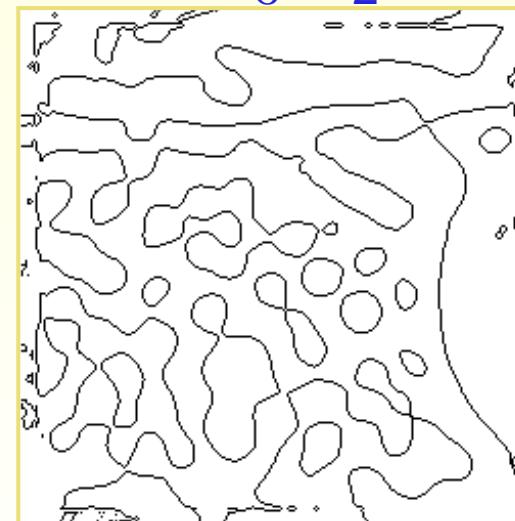
$$\sigma^2 = \sqrt{2}$$



$$\sigma^2 = 2$$



$$\sigma^2 = 2\sqrt{2}$$



$$\sigma^2 = 4$$



Evaluación de los detectores de fronteras

- ¿Cómo podemos decidir qué tipo de detector de fronteras deberíamos usar en cada aplicación?
- Para este propósito, necesitamos encontrar un método para evaluar el funcionamiento de los detectores.
- En primer lugar, necesitamos crear una o más imágenes para las que conozcamos las localizaciones y orientaciones de los bordes.
- Una forma simple de conseguirlo es construir una imagen solo con objetos geométricos.
- Una posible medida para evaluar los detectores de bordes se denomina **valor de bondad**.



Evaluación de los detectores de fronteras

- El **valor de bondad** VB se calcula como:

$$VB = \frac{1}{\max(I_A, I_I)} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha d_i^2}$$

con

- I_A , el número de bordes detectado
- I_I , el número ideal de bordes
- d , la distancia entre los bordes detectados y los ideales (determinada mediante un algoritmo adecuado) y
- α , una constante para penalizar los bordes desplazados



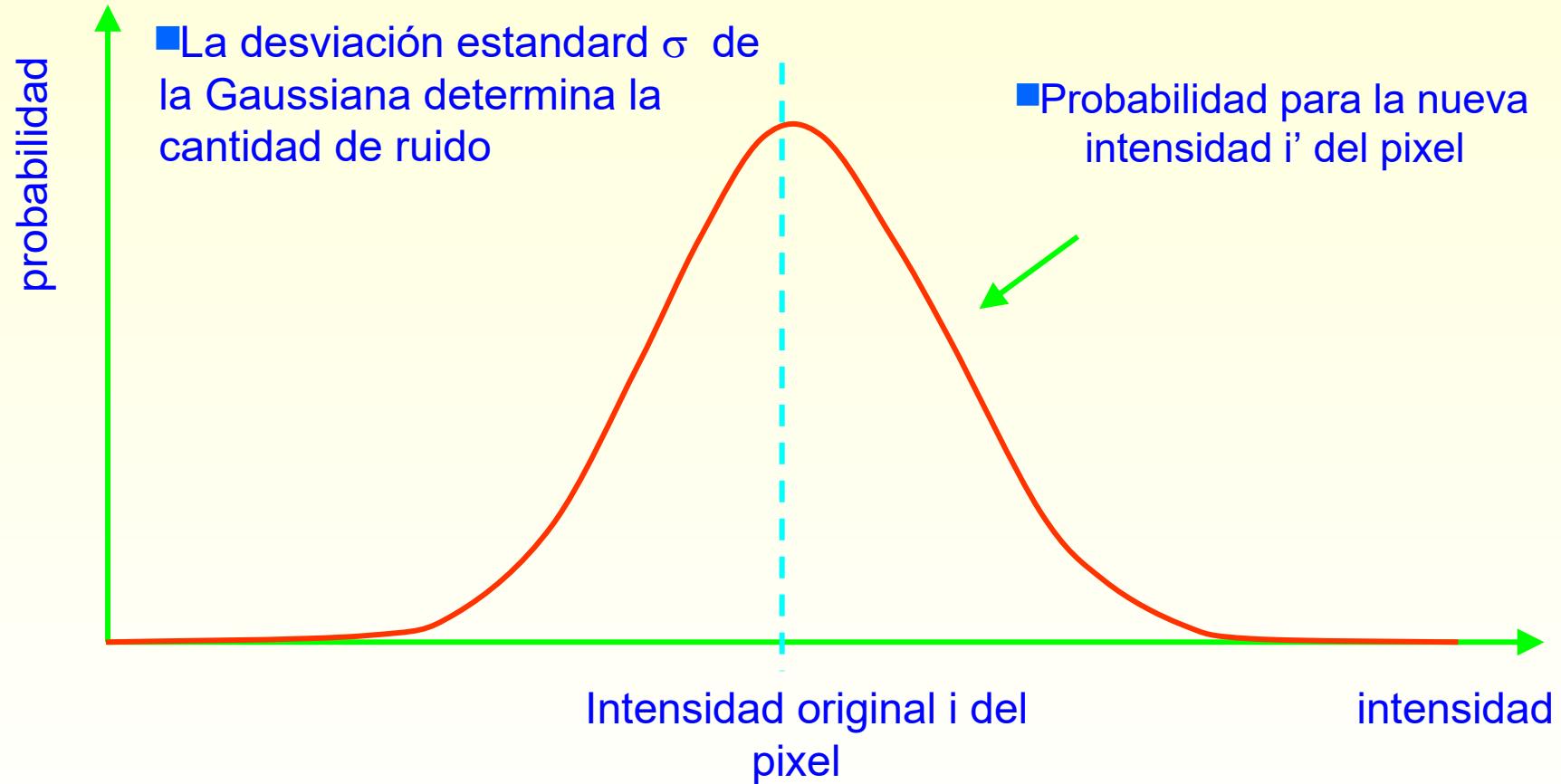
Evaluación de los detectores de fronteras

- El valor máximo de VB es 1 lo que indica funcionamiento perfecto.
- Cuanto más cercano a 0 esté VB , peor será el funcionamiento.
- Por supuesto puede ocurrir que un detector de bordes pobre funcione muy bien en imágenes artificiales que contengan p.ej. solo rectángulos perfectos.
- Para hacer un análisis más significativo, deberíamos **sumar ruido** a nuestras imágenes test y ver cómo de bien funciona con ellas nuestro detector.
- Una posibilidad es simular grados variables de **ruido Gaussiano** en nuestras imágenes test.



Evaluación de los detectores de fronteras

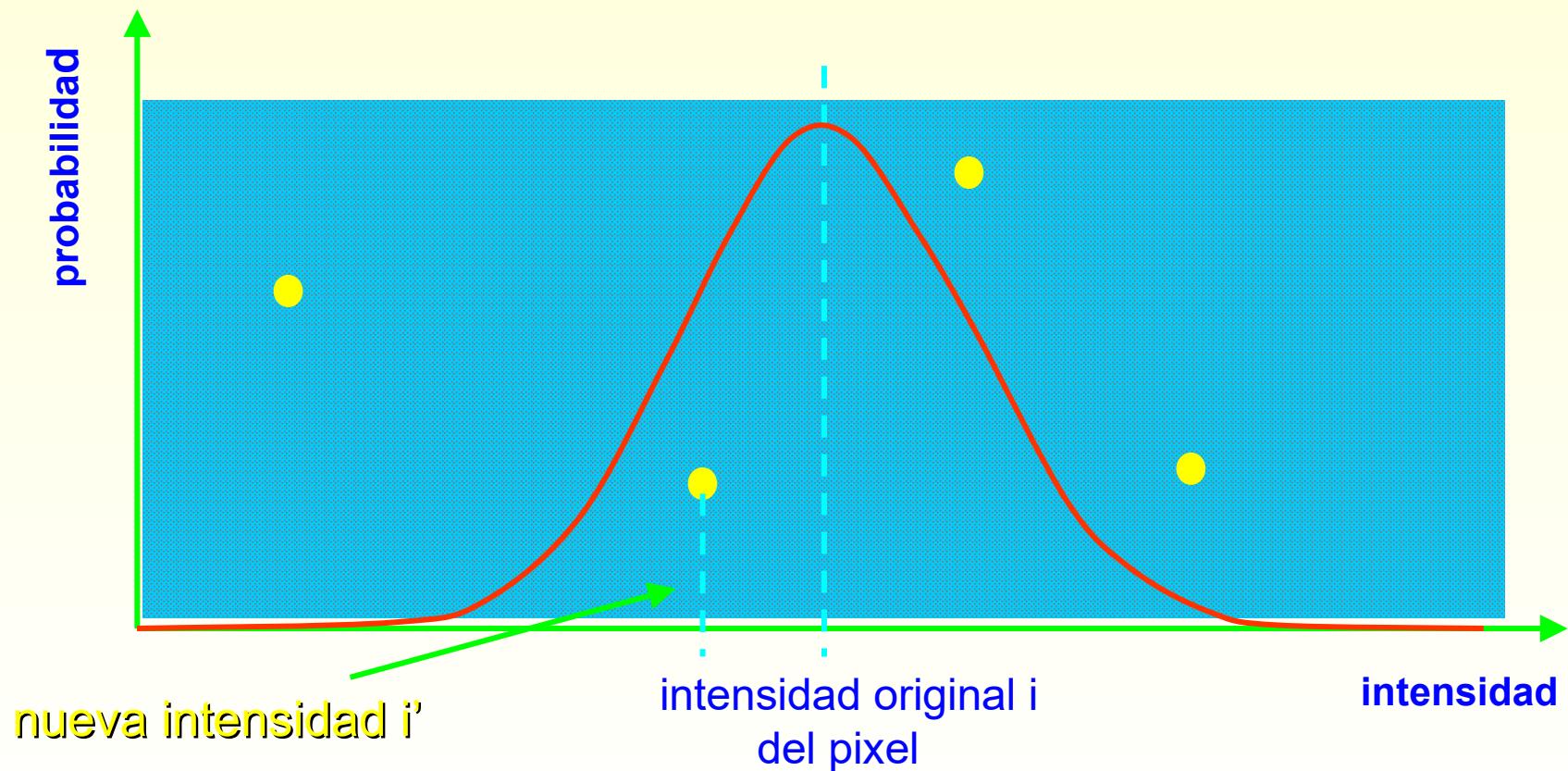
- Para simular ruido Gaussiano, hay que cambiar la intensidad de cada pixel de acuerdo a una distribución aleatoria Gaussiana:





Evaluación de los detectores de fronteras

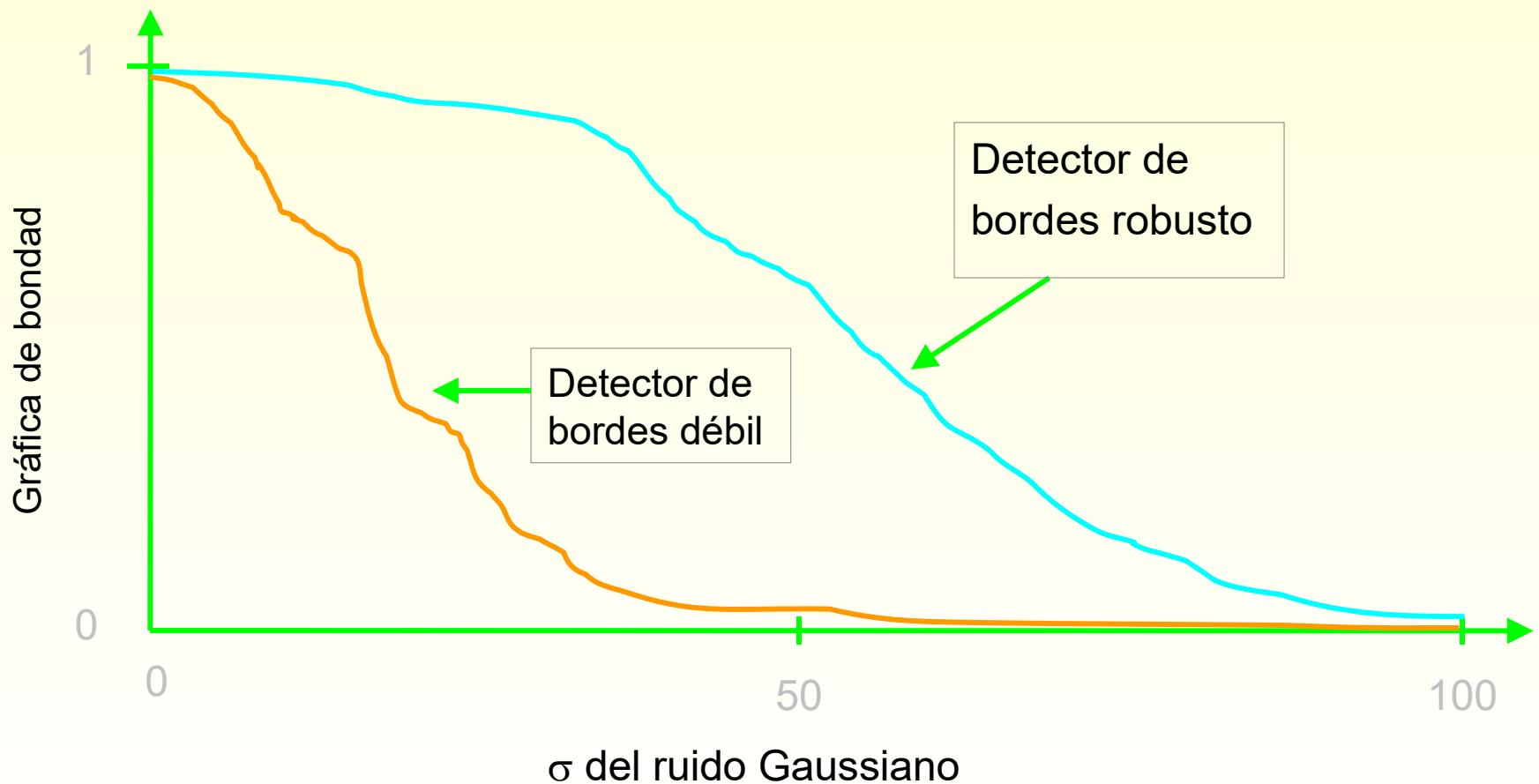
- Para simular una distribución aleatoria Gaussiana, generar puntos aleatorios en el área rectangular 2D marcada hasta que uno de ellos quede bajo la curva Gaussiana.
- Poner i' a la coordenada horizontal de ese punto.





Evaluación de los detectores de fronteras

- Podemos entonces calcular la gráfica de bondad como una función del ruido Gaussiano sumado a la imagen de entrada:

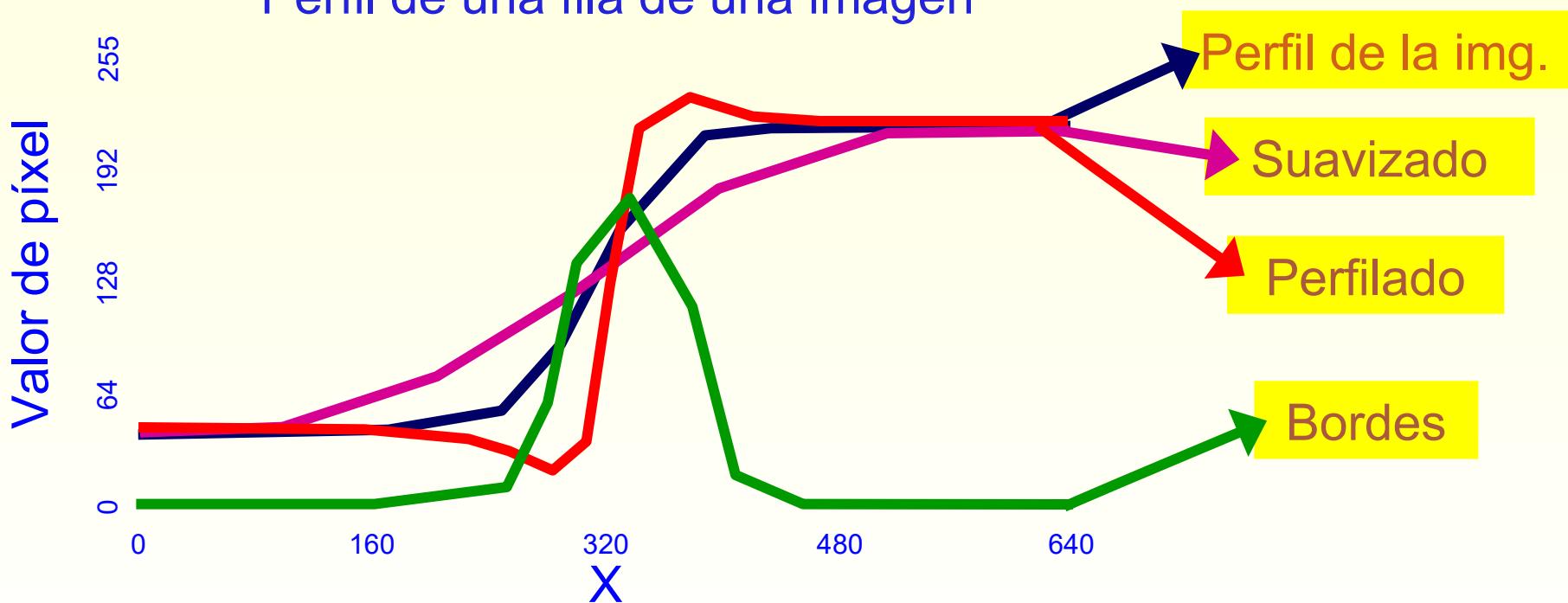




- **Perfilado** y **detección de bordes** están relacionados con el suavizado:

- **Suavizado**: reducir las variaciones en la imagen.
- **Perfilado**: aumentar las variaciones en la imagen.
- **Bordes**: encontrar las zonas de variación.

Perfil de una fila de una imagen





Perfilado



- **Perfilado:** destacar y hacer más visibles las variaciones y bordes de la imagen. Es lo contrario al suavizado.
- Permite eliminar la apariencia borrosa de las imágenes, debida a imperfecciones en las lentes.
- ... aunque tampoco se pueden hacer milagros...



← Suavizado

Original

Perfilado →



- El perfilado se puede conseguir sumando a la imagen **original**, la **laplaciana** ponderada por cierto factor.
- Lo cual equivale a usar una máscara de convolución adecuada:

$$1 \cdot \begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix} + \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix} = \begin{matrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{matrix}$$

- Más o menos perfilado dando distintos pesos, **a**.

$$a \cdot \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} + \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix} = \begin{matrix} 0 & -a & 0 \\ -a & 4a+1 & -a \\ 0 & -a & 0 \end{matrix}$$



Perfilado

- **Ejemplos.** Variando pesos y tamaño de la laplaciana.



Imagen de entrada



Perfilado 33%, 3x3



Perfilado 60%, 3x3

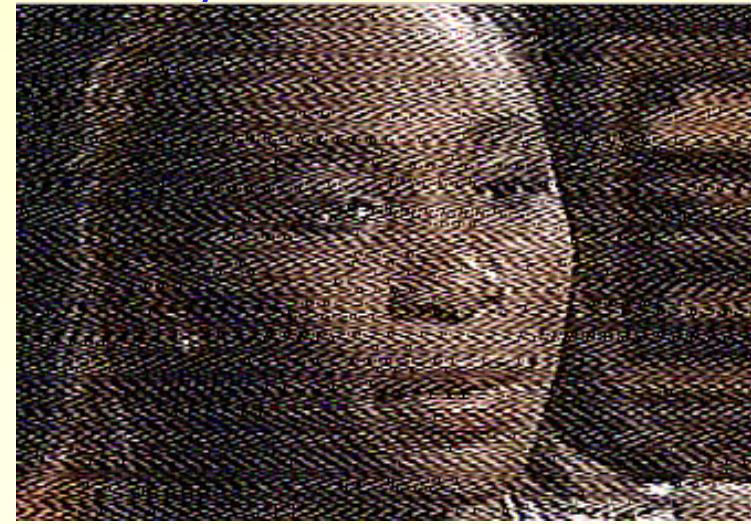


Perfilado 15%, 7x7

- **Cuidado con el perfilado.** La operación de perfilado aumenta el nivel de ruido de la imagen.



Jen con ruido
interferencias
TV



Perfillado 33%, 3x3



**magen con ruido
por compresión**



Perfilado 60%, 3x3



Perfilado

Conclusiones:

- Las **convoluciones** son una herramienta fundamental en procesamiento de imágenes.
 - **Una misma base común:** combinaciones lineales de una vecindad local de los píxeles (de cierto tamaño).
 - **Diversos usos:** según los valores de los coeficientes: suavizado, eliminación de ruido, bordes, perfilado, etc.
- Se pueden definir **operaciones similares** sobre **vídeo** (usando la dimensión temporal, por ejemplo, suavizado a lo largo del tiempo), y sobre **audio digital** (por ejemplo, suavizado de la señal o introducción de eco).
- Es importante conocer el **significado matemático** de los procesos aplicados (derivadas, gradientes, integrales,...).



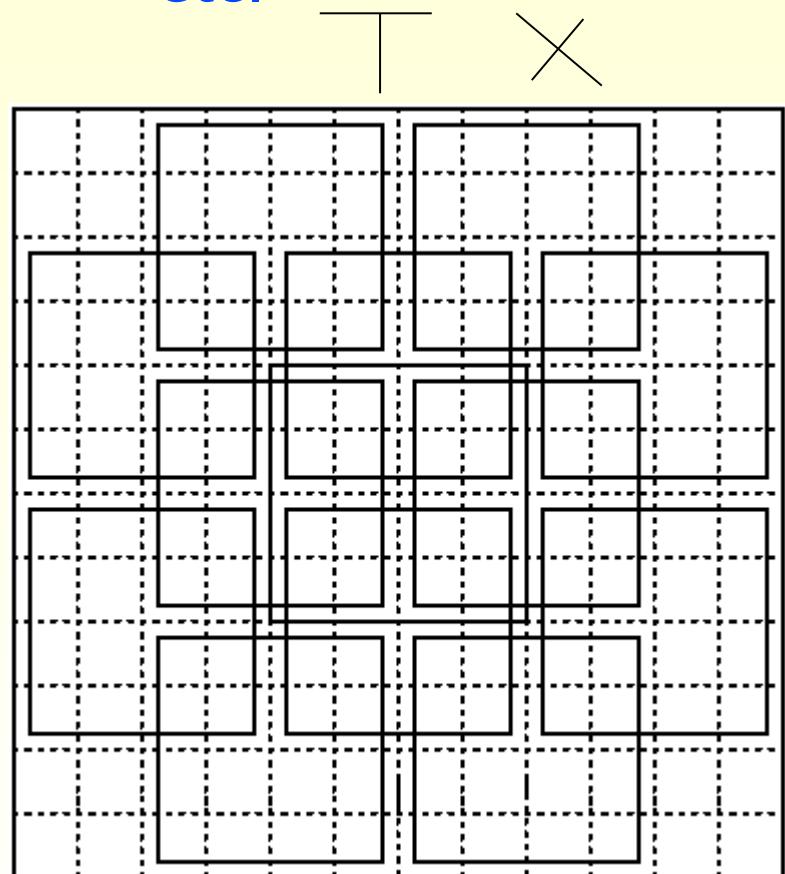
Detección de Rasgos Operador de Moravec

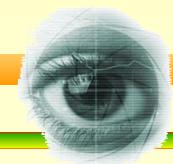


Una esquina puede definirse como la intersección de dos bordes. También puede definirse como un punto para el que hay dos direcciones de bordes dominantes y diferentes en una vecindad local del punto.

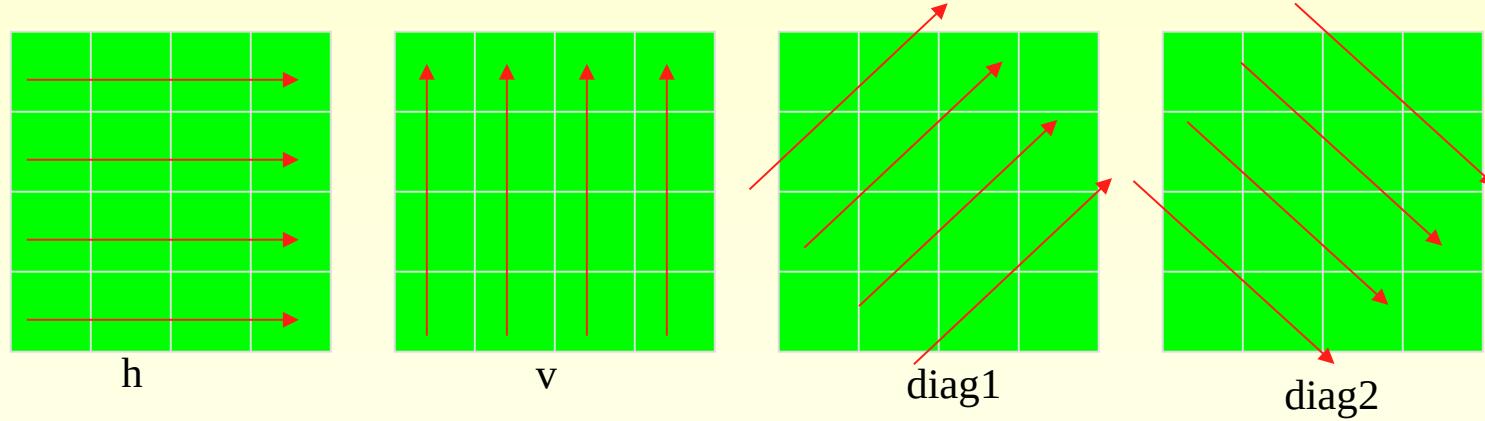
Un punto de interés es un punto en una imagen que tiene una posición bien definida y puede ser detectado de forma robusta.

- T-uniones, cruces etc.





- Calcular las variaciones de intensidad en 4 direcciones

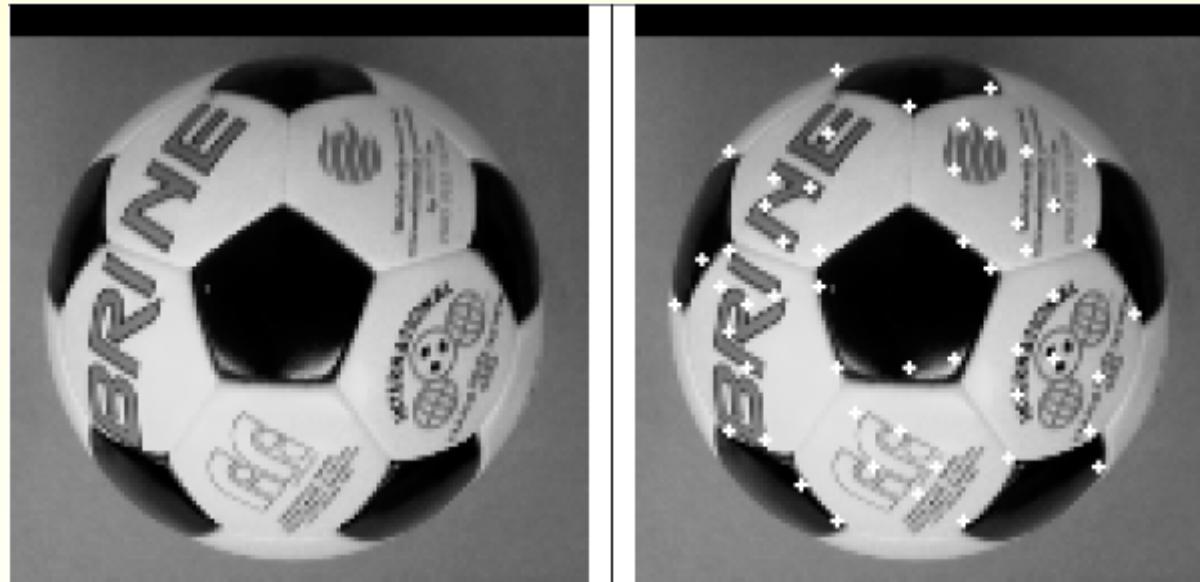


$$\begin{aligned} hh(i, j) &= \sum_{r, c \in \{-1, 0, 1\}} |h(i-r, j-c)| & vv(i, j) &= \sum_{r, c \in \{-1, 0, 1\}} |v(i-r, j-c)| \\ d1(i, j) &= \sum_{r, c \in \{-1, 0, 1\}} |diag1(i-r, j-c)| & d2(i, j) &= \sum_{r, c \in \{-1, 0, 1\}} |diag2(i-r, j-c)| \end{aligned}$$

- $c(i,j)=\min(hh(i,j),vv(i,j),d1(i,j),d2(i,j))$
- Quedarnos con las esquinas más fuertes. Si el máximo en el entorno de $c(i,j)$ es diferente de $c(i,j)$ poner $c(i,j)=0$



Ejemplo





- Calcular las derivadas de la imagen I_x, I_y
- Construir la matriz de momentos del gradiente

$$\Delta I = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \quad M = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} \quad M = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad M_H = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

- Calcular los valores propios de M_H
 - Si ambos valores propios son altos, esto indica un punto de esquina
- Se puede calcular lo mismo usando

$$R = \min(\text{valores propios}(M_H)) > Th$$

$$R = \det(M_h) - k \operatorname{traza}(M_h)^2 > Th$$



Ejemplo

