



Extracción de Características: Bordes, Esquinas y Puntos Singulares

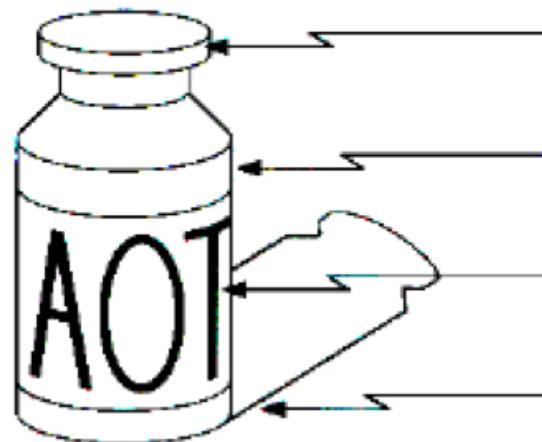
Referencias

- R. C. Gonzalez and R. E. Woods. Digital Image Processing. Addison-Wesley Longman Publishing Co., Inc., 2001.
- M. Sonka, V. Hlavac, and R. Boyle. Image Processing: Analysis and Machine Vision. Thomson-Engineering, 1998.
- E. Trucco and A. Verri. Introductory techniques for 3D computer Vision, Prentice-Hall, 1998.



Detectores de bordes

- Un **borde** se corresponde con un cambio fuerte en intensidad en la imagen
 - El cambio puede deberse a la aparición de diferentes objetos, cambio en la orientación de superficie, sombras etc
- Los bordes determinan rasgos relevantes de la imagen
- En reconocimiento de objetos se usa muy frecuentemente información de bordes (tanto en visión artificial como biológica)





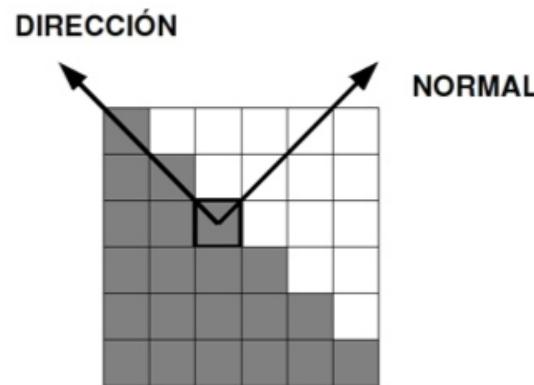
Detectores de bordes

• Motivación

- Recopilar información de bordes nos ayuda a identificar contornos relevantes de la imagen y caracterizar regiones englobadas en dichos contornos
- Una imagen de bordes representa un nivel superior de abstracción ⇒ menos información a procesar
- Los bordes son invariantes a la iluminación absoluta de la escena (en oposición a información de color, por ejemplo)



Detectores de bordes

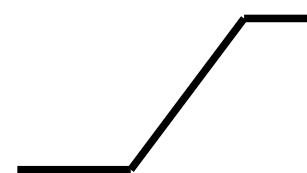
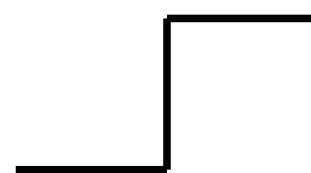
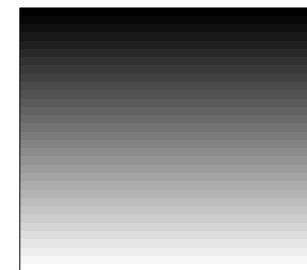
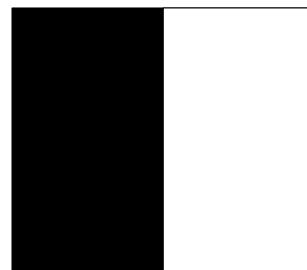


- Normal al borde: perpendicular al contorno del borde.
- Dirección del borde: tangente al contorno del borde.
- Intensidad: fuerza de la variación del nivel de gris.
- Posición: píxel en el que se produce el cambio en la intensidad.



Detectores de bordes

- Detección de discontinuidades en niveles de gris
 - Borde “ideal” → escalón en la intensidad
 - Borde “real” → rampa en la intensidad



Perfiles de nivel de gris



Detectores de bordes y derivadas

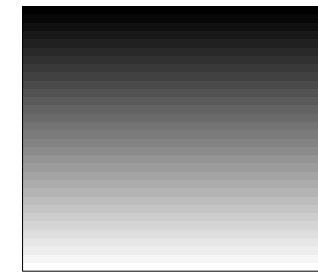
- Detección usando derivadas

- **Primera derivada**

- Positiva → entrar/salir transición
 - Constante → rampa
 - Cero → nivel de gris constante

- **Segunda derivada**

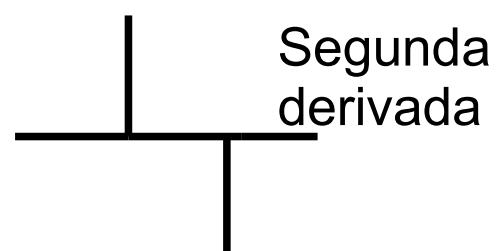
- Positiva → transición lado más oscuro
 - Negativa → transición lado más claro
 - Cero → rampa y nivel de gris constante



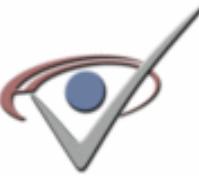
Perfil de nivel de gris



Primera derivada

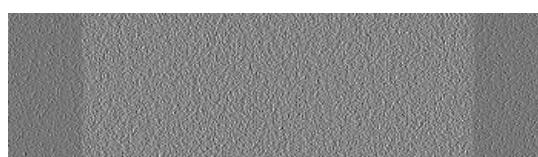
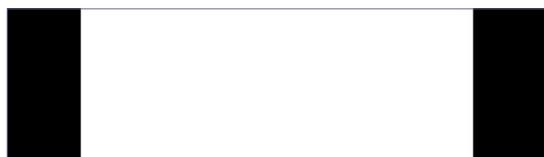


Segunda derivada



Detectores de bordes con ruido

- Problemas de las derivadas en imágenes con ruido
 - Suavizado





Operadores de primera derivada

- Primera derivada

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix}$$

- **Magnitud**

$$|\nabla f| = \sqrt(G_x^2 + G_y^2) \quad |\nabla f| \approx |G_x| + |G_y|$$

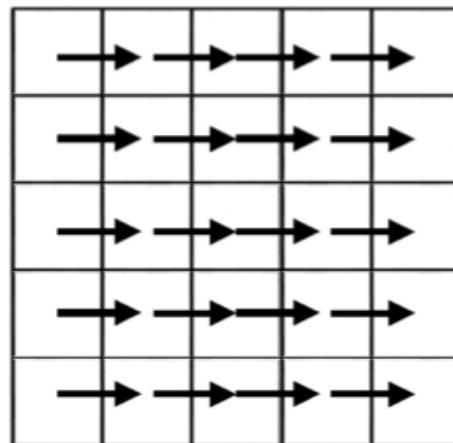
- **Orientación**

$$\phi = \arctan\left(\frac{G_y}{G_x}\right)$$



Operadores de primera derivada

- En procesado de imágenes es necesario discretizar las funciones
 - Cálculo por diferencias finitas entre píxeles cercanos



$$\Delta_i f(i, j) = f(i, j) - f(i - n, j)$$

$$\Delta_j f(i, j) = f(i, j) - f(i, j - n)$$

- El cálculo de estas diferencias a lo largo de la imagen: convolución con máscaras direccionales

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

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

Operadores de primera derivada: Roberts



- Aproximación primera derivada

Región de la imagen

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

- Máscaras de convolución

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

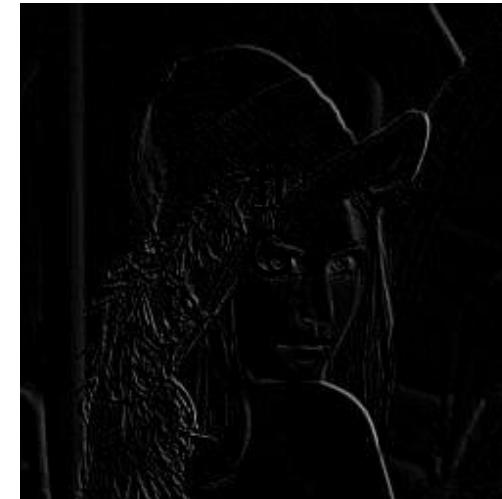
Operadores primera derivada: Roberts



G_x



G_y



Magnitud



Umbralizada





Operadores de primera derivada

- **Operador de Prewitt**

- Basado en la idea de diferencias centrales

$$\frac{\partial I}{\partial x} \approx [I(x+1, y) - I(x-1, y)]/2$$

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

- Muy sensible al ruido
- Se calcula un promedio para reducir la sensibilidad al ruido

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

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

$$\partial/\partial x$$

$$\partial/\partial y$$



Operadores de primera derivada

- **Operador de Sobel**

- Similar a Prewitt pero en el promedio pondera más los píxeles centrales

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

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

- Ventaja de Prewitt y Sobel: Separabilidad de las máscaras. Reducción de la complejidad

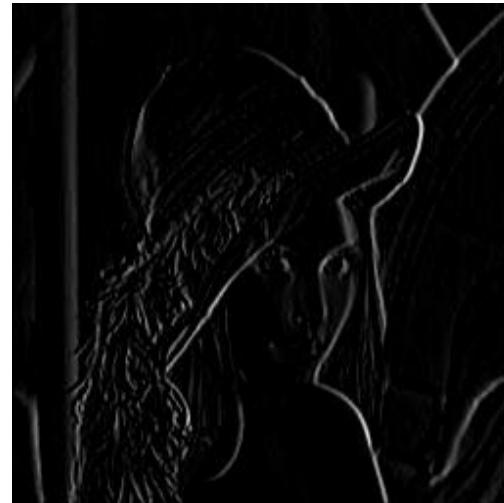
$$\Delta_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$$

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

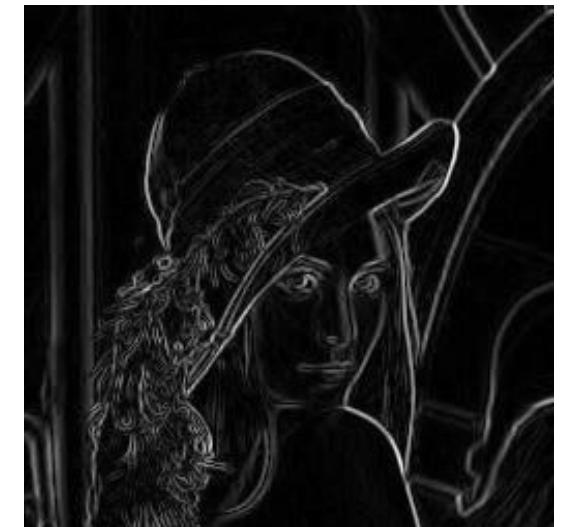
Operadores de primera derivada: Sobel



G_x



G_y





Otros operadores de primera derivada

- Operadores de compás
 - Operadores no lineales
 - La imagen se convoluciona con n máscaras de convolución en *n direcciones espaciales*
 - Magnitud

$$|G| = \max(|(G_i)|, i=1..n)$$

- Orientación

$$k, \text{ tal que } |(G_k)| = \max(|(G_i)|, i=1..n)$$



Otros operadores de primera derivada

- Máscaras de compás

- Robinson

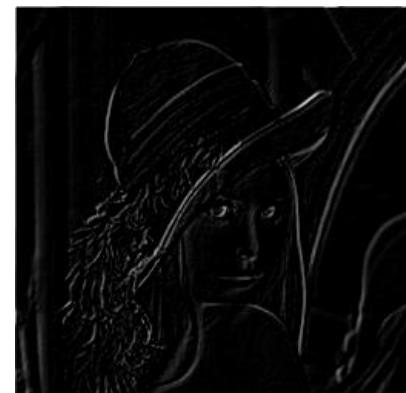
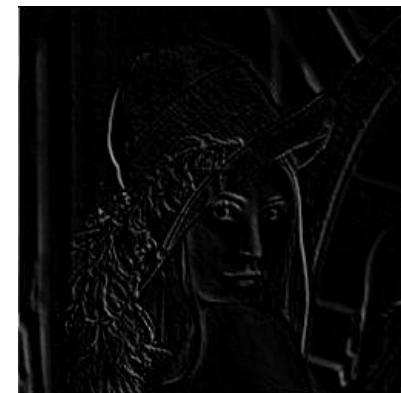
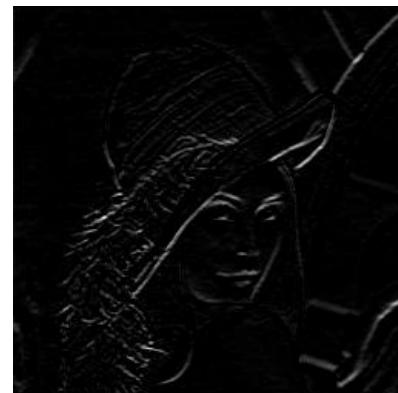
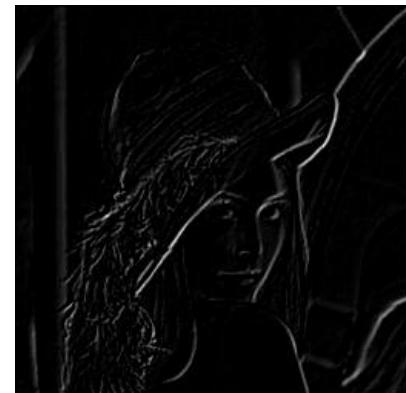
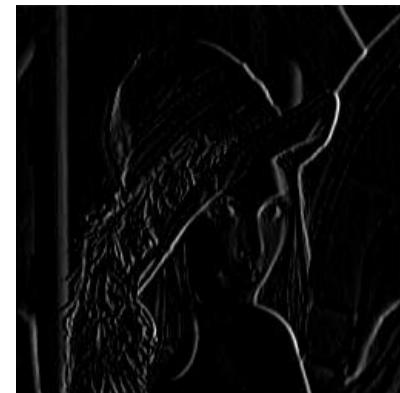
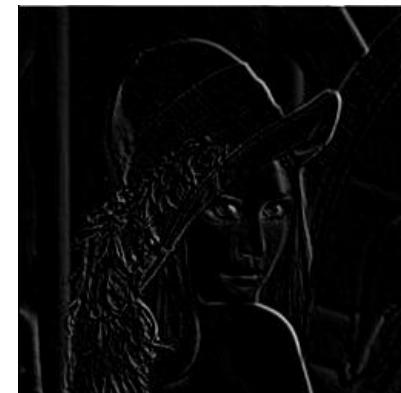
$$G_0 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} \quad G_{\pi/4} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix} \quad G_{\pi/2} = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \quad G_{3\pi/4} = \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$G_\pi = \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad G_{5\pi/4} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix} \quad G_{3\pi/2} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix} \quad G_{7\pi/4} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$$

- Kirsch

$$G_0 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} \quad G_{\pi/4} = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} \quad G_{\pi/2} = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix} \quad G_{3\pi/4} = \begin{bmatrix} -5 & -5 & 3 \\ -5 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix} \dots$$

Otros operadores de primera derivada: Kirsch





Operadores de primera derivada



Roberts



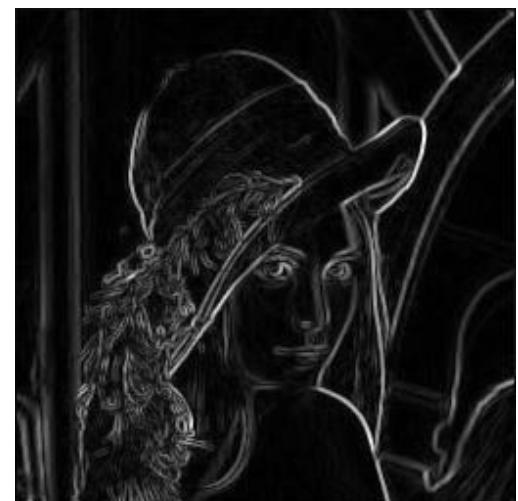
Prewitt



Sobel



Robinson



Kirsch



Operadores de primera derivada

- Roberts
 - Sensible al ruido
- Sobel, Prewitt
 - Producen imágenes de bordes buenos
- Kirsch, Robinson
 - Producen imágenes de bordes buenos
 - Elevado coste computacional
 - Convolución con n máscaras



Ejercicio

- Definir una máscaras 3x3 que permita detectar únicamente líneas verticales

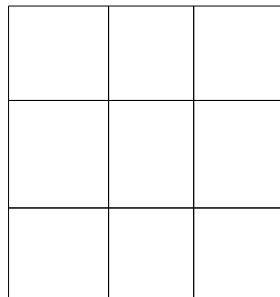


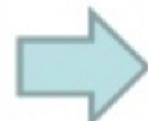
Imagen de prueba

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



Operadores de segunda derivada

Operador
Laplaciano



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

Segunda derivada en los puntos $x+1$ e $y+1$

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

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

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

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

Segunda derivada en los puntos x e y

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

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

Función que define la máscara

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



Operadores de segunda derivada

- Laplaciano

Invariantes a rotaciones

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

Más importancia al pixel central
Pierde invariancia a rotaciones

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

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



Produce bordes dobles



Operadores de segunda derivada

- Ejemplo de Laplaciano

2	2	2	2	2	2	8	8	8	8	8	8
2	2	2	2	2	2	8	8	8	8	8	8
2	2	2	2	2	2	8	8	8	8	8	8
2	2	2	2	2	2	8	8	8	8	8	8
2	2	2	2	2	2	8	8	8	8	8	8
2	2	2	2	2	2	8	8	8	8	8	8

0	0	0	6	-6	0	0	0
0	0	0	6	-6	0	0	0
0	0	0	6	-6	0	0	0
0	0	0	6	-6	0	0	0

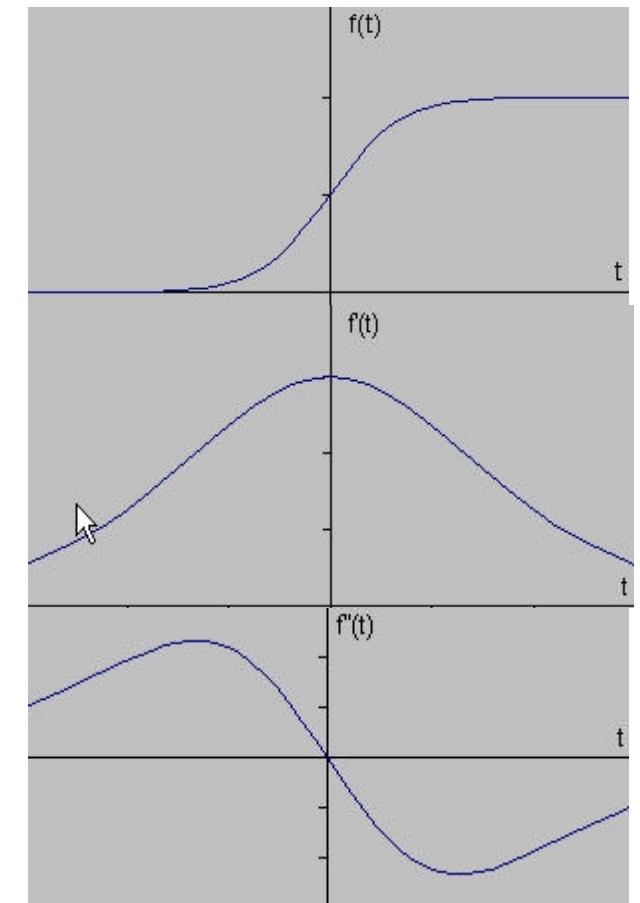
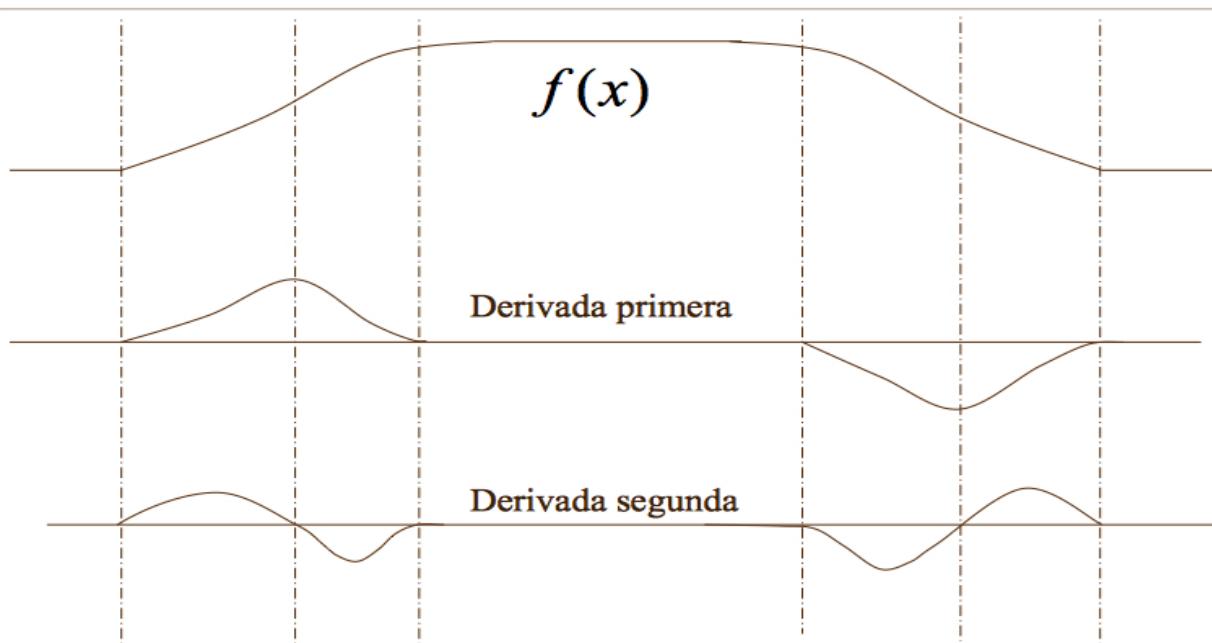
- Problemas

- Muy sensible al ruido.
- Bordes dobles. Interpolar posición borde.
- No información sobre orientación.



Operadores de segunda derivada

- Borde → cambio brusco en la intensidad
 - Primera derivada → valor extremo
 - Segunda derivada → cruce por cero





Operadores de segunda derivada

- Algoritmo
 - 1. Calcular la segunda derivada de la imagen
 - Imagen suavizada para minimizar el efecto del ruido
 - 2. Localizar los cruces por cero en la segunda derivada



Operadores de segunda derivada

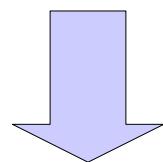
- **Laplaciana de Gaussiana (LOG)
(MARR-HILDRETH)**

1. Calcular la segunda derivada de una imagen suavizada con una máscara gaussiana

- σ indica el nivel de suavizado

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

$$\nabla^2[G(x, y) * I(x, y)]$$



$$\nabla^2[G(x, y)] * I(x, y)$$

Operadores de segunda derivada: LoG



- Cálculo de $\nabla^2[G(x, y)]$

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

$$G(r) = e^{-r^2/2\sigma^2}$$

$$G'(r) = -\frac{1}{\sigma^2} r e^{-r^2/2\sigma^2}$$

$$G''(r) = -\frac{1}{\sigma^2} \left(\frac{r^2}{\sigma^2} - 1 \right) e^{-r^2/2\sigma^2}$$

$$\nabla^2 G(x, y) = c \frac{x^2 + y^2 - \sigma^2}{\sigma^4} e^{-(x^2 + y^2)/2\sigma^2}$$



Normaliza la suma de los elementos de la máscara a 0



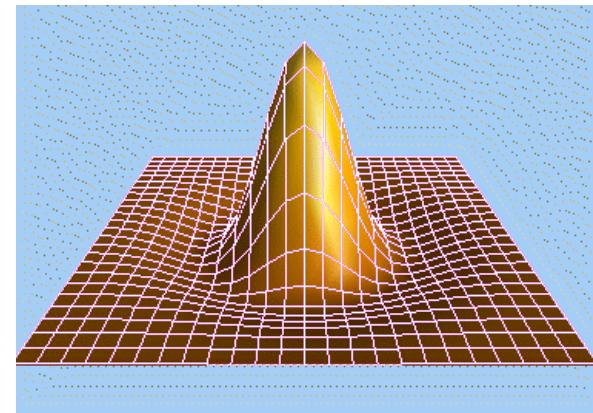
Operadores de segunda derivada: LoG

- Algoritmo

1. Convolucionar la imagen con una máscara obtenida a partir de

$$\nabla^2 G(x, y) = c \frac{x^2 + y^2 - \sigma^2}{\sigma^4} e^{-(x^2 + y^2)/2\sigma^2}$$

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



Sombrero mexicano

Operadores de segunda derivada: LoG



- Algoritmo

2. Localizar los cruces por cero en la imagen resultante del paso 1

- Existe un cruce por cero en una posición (x,y) de la imagen si su valor asociado es inferior a un umbral predefinido $-t$ y alguno de sus ocho vecinos tiene un valor superior a t

Operadores de segunda derivada: LoG



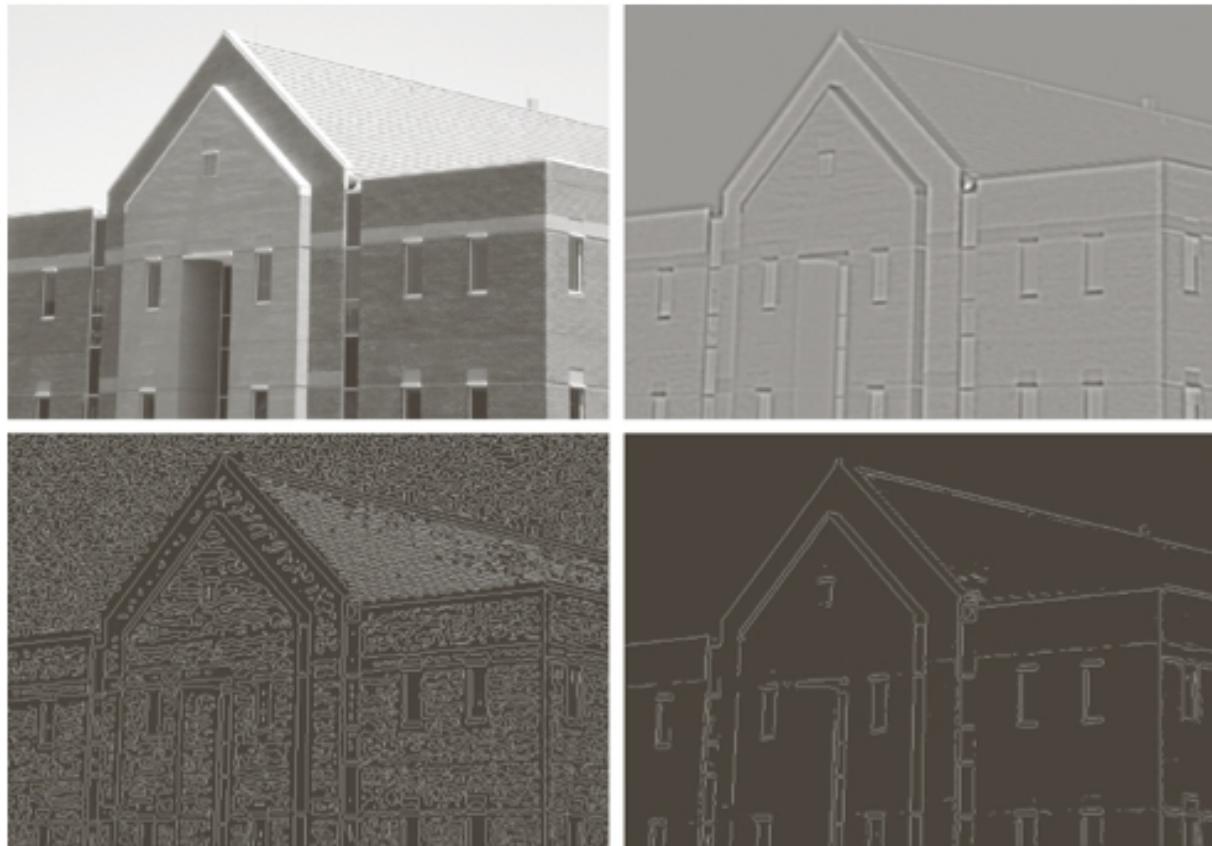
5×5
 $\sigma = 0.5$



11×11
 $\sigma = 1.5$



Operadores de segunda derivada: LoG



$$\begin{array}{c|c} a & b \\ \hline c & d \end{array}$$

- ▶ A 834×1114 image (2) is processed with a Laplacian of Gaussian filter ($\sigma = 4$, $n = 25$) (b).
- ▶ Zero crossing with threshold 0 (c) and 4% of the maximum value in (b).

Operadores de segunda derivada: LoG



- Mayor suavizado
 - Eliminar ruido
 - Eliminar bordes espúreos
 - Detectar bordes más gruesos



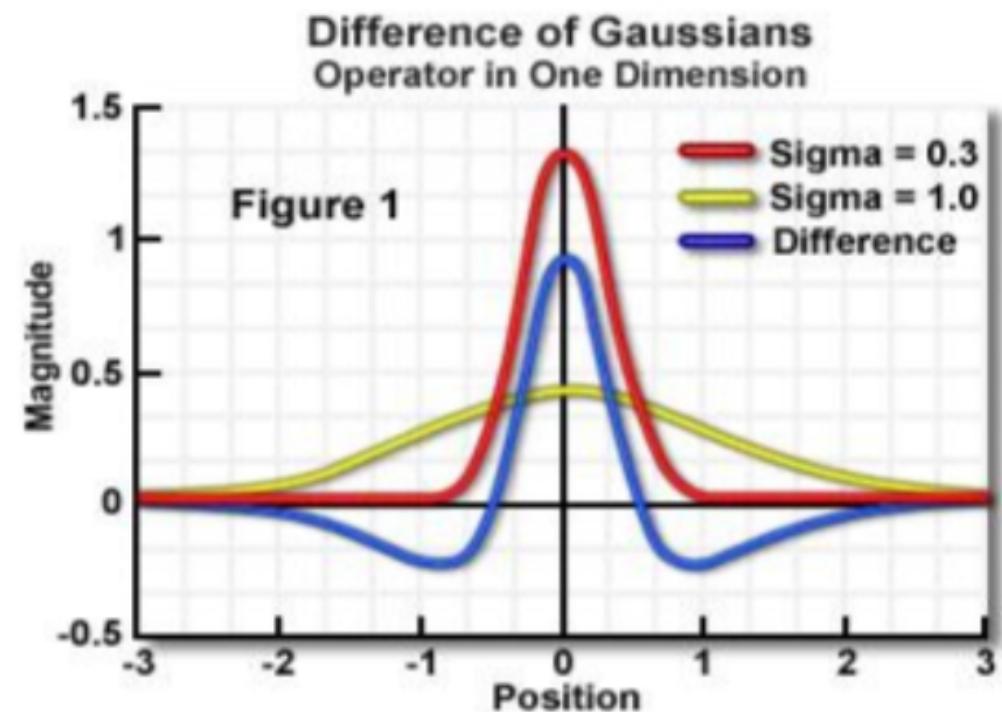
Operadores de segunda derivada

- **Diferencia de Gaussianas (DoG)**

- El operador LoG puede ser aproximado de forma eficiente por una convolución con una máscara que es la diferencia de dos Gaussianas con σ diferentes

$$DoG(x, y) = G(x, y, \sigma_1)$$

$$-G(x, y, \sigma_2)$$



Operadores de segunda derivada: DoG



- Algoritmo

1. Convolucionar la imagen con la máscara de convolución DoG
2. Calcular los cruces por cero en la imagen resultante

11×11
 $\sigma_1 = 1.2$
 $\sigma_2 = 1.7$





Operador de Canny

- Criterios para construir un detector óptimo
 - Criterio 1: **Detección**
 - Bordes importantes no deben ser eliminados
 - No deben producirse respuestas espúreas
 - Criterio 2: **Localización**
 - La distancia entre el borde real y el localizado debe ser mínima
 - Criterio 3: **Una respuesta**
 - Minimizar múltiples respuesta a un borde único



Operador de Canny

- Algoritmo

1. Mejora de la imagen

- Suavizar la imagen y eliminar el ruido
- Localizar los bordes en la imagen mejorada

2. Supresión no máxima

- Producir bordes de 1 pixel de grosor

3. Umbralización con histéresis

- Reducir la probabilidad de falsos contornos

Operador de Canny: Mejora de la imagen



Objetivo

- Eliminar posible ruido que pueda afectar a la detección de bordes

Método

1. Aplicar a la imagen I un suavizado gaussiano

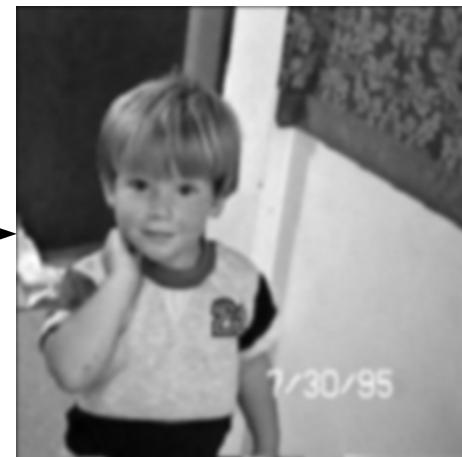
$$J(i, j) = I(i, j) * G(i, j)$$

2. Para cada pixel (i,j)

- Usar un detector de bordes para calcular las componentes del gradiente J_x y J_y
- Calcular la magnitud y orientación de los bordes

$$E_m(i, j) = \sqrt{J_x^2(i, j) + J_y^2(i, j)} \quad E_o(i, j) = \arctan(J_y / J_x)$$

Operador de Canny: Mejora de la imagen

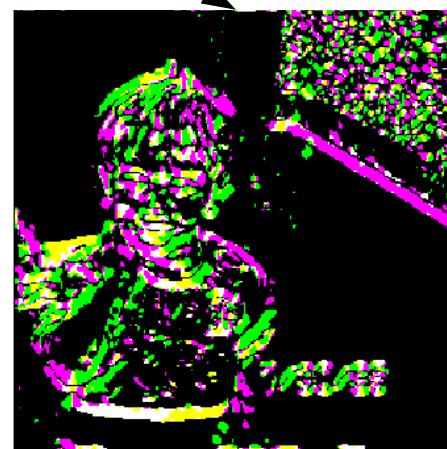


1. Suavizado

Máscara gaussiana 11×11 , $\sigma = 1.4$



Magnitud



Orientación

2. Detección de bordes

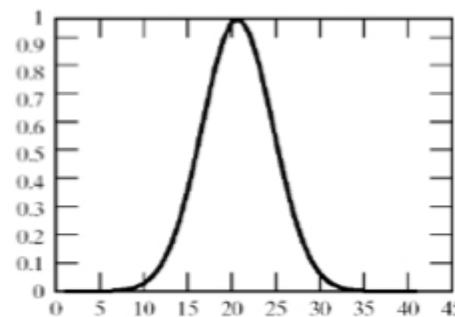
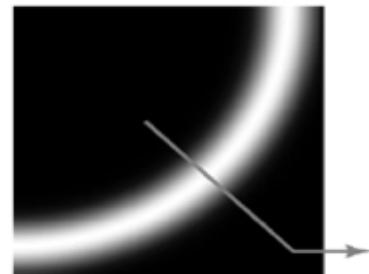
Operador de Canny: Supresión no máxima

Objetivo

- Obtener bordes de 1 pixel de grosor

Método

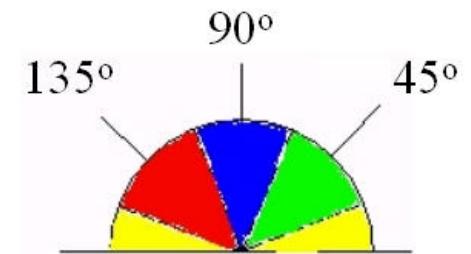
- Un borde puede tener varios píxeles de grosor
- Considerar sólo píxeles cuya magnitud es máxima
- Descartar aquellos píxeles cuyas magnitudes no alcancen ese máximo



Operador de Canny: Supresión no máxima

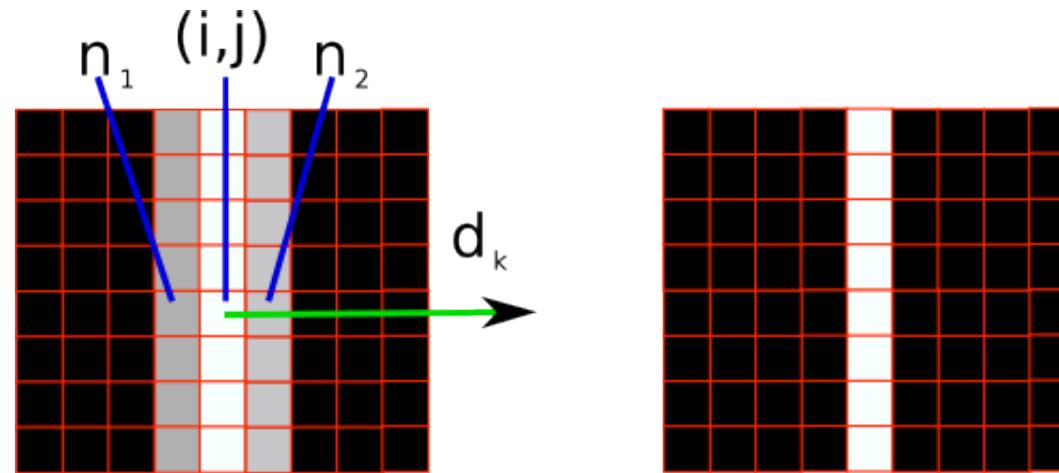
Método

- Para cada pixel (i,j) :
 1. Encontrar la dirección d_k que mejor aproxime la dirección $E_o(i,j)$
 - $E_o(i,j) \rightarrow$ normal al borde
 2. Asignar el valor al nuevo pixel



$$I_N(i, j) = \begin{cases} 0 & \text{Si } E_s(i, j) < E_s(n_1) \text{ y } E_s(i, j) < E_s(n_2) \\ & n_1, n_2 \text{ pixeles vecinos en dirección } d_k \\ E_s(i, j) & \text{En otro caso} \end{cases}$$

Operador de Canny: Supresión no máxima



Operador de Canny: Umbralización con histéresis



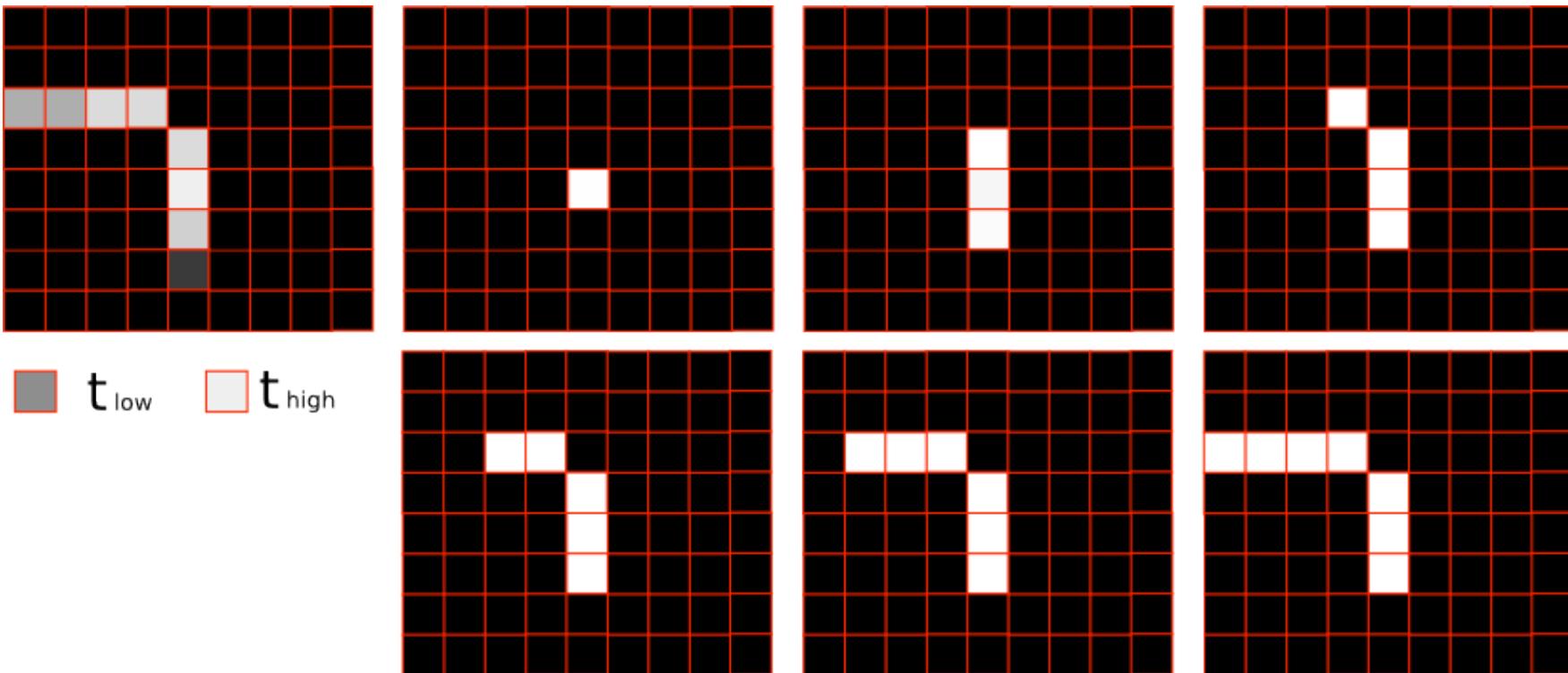
Objetivo

- Descartar los posibles máximos locales creados por ruido sin eliminar bordes débiles
- Se utilizarán dos umbrales t_{low} y t_{high} , $t_{low} < t_{high}$

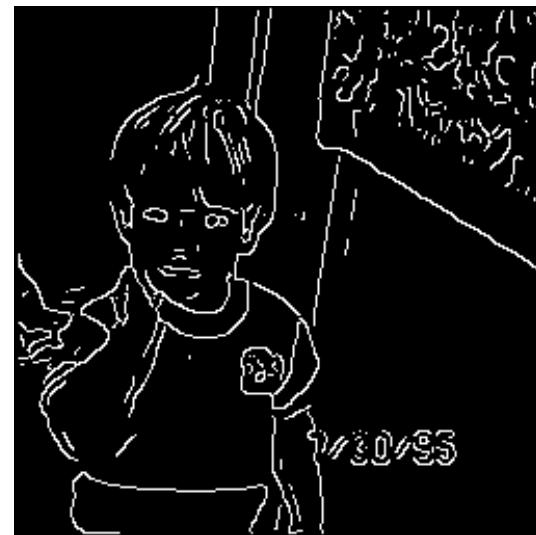
Método

- Para todos los puntos (i,j) en I_N
 1. Localizar el próximo punto borde no visitado, $I_N(i,j)$ tal que $I_N(i,j) > t_{high}$
 2. A partir del punto $I_N(i,j)$ recorrer los pixels conectados
 - En ambas direcciones perpendiculares a la normal del borde,
 - Mientras se cumpla que $I_N > t_{low}$
 - Indicar la presencia de borde en la imagen $H(i,j)$
 - Marcar los puntos que se recorren como puntos visitados.

Operador de Canny: Umbralización con histéresis

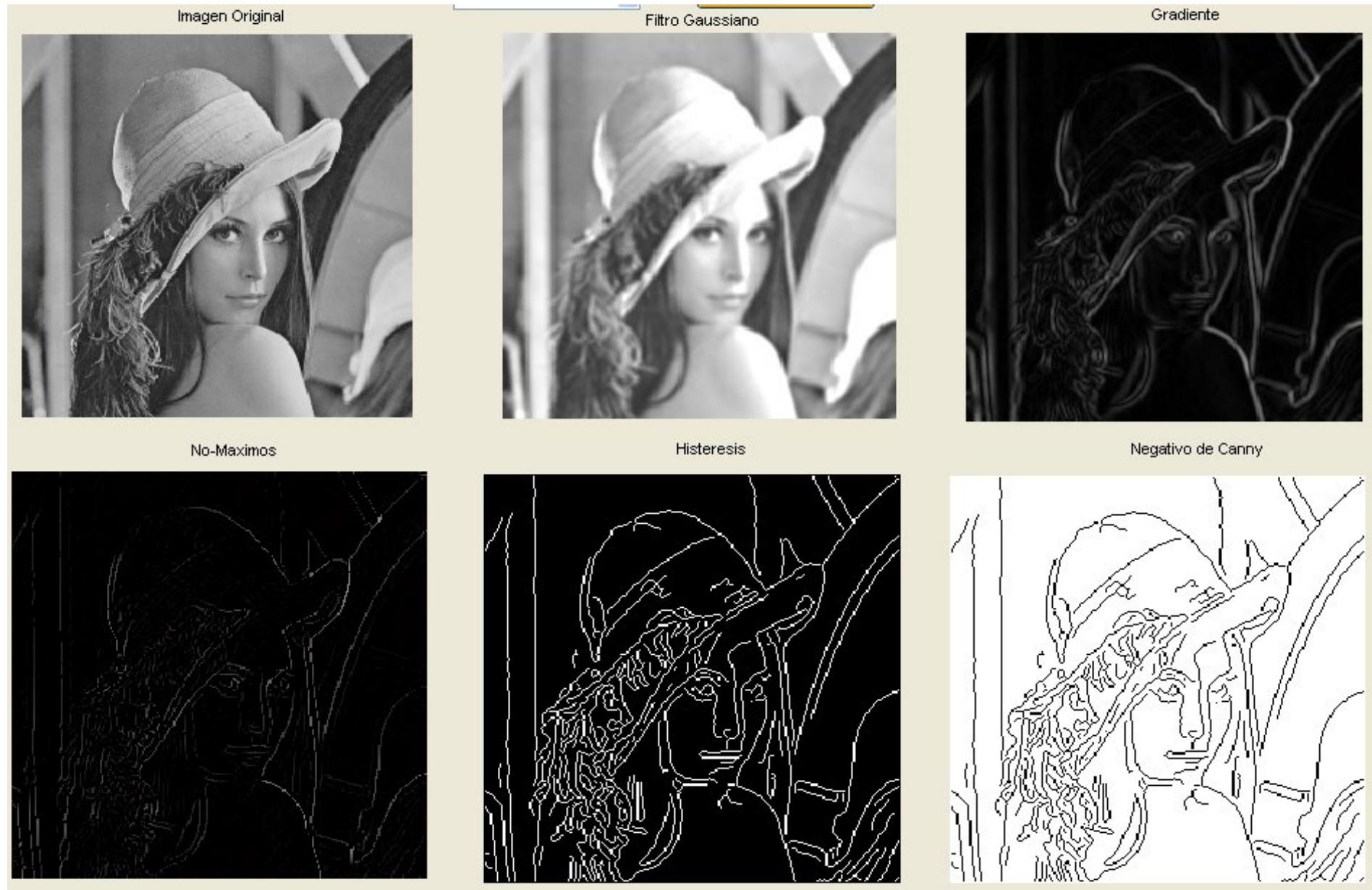


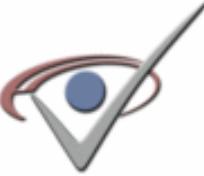
Operador de Canny: Umbralización con histéresis





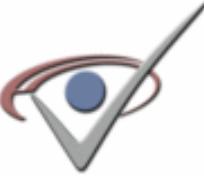
Operador de Canny: Ejemplo





Operador de Canny: Ejemplo





Canny vs Marr

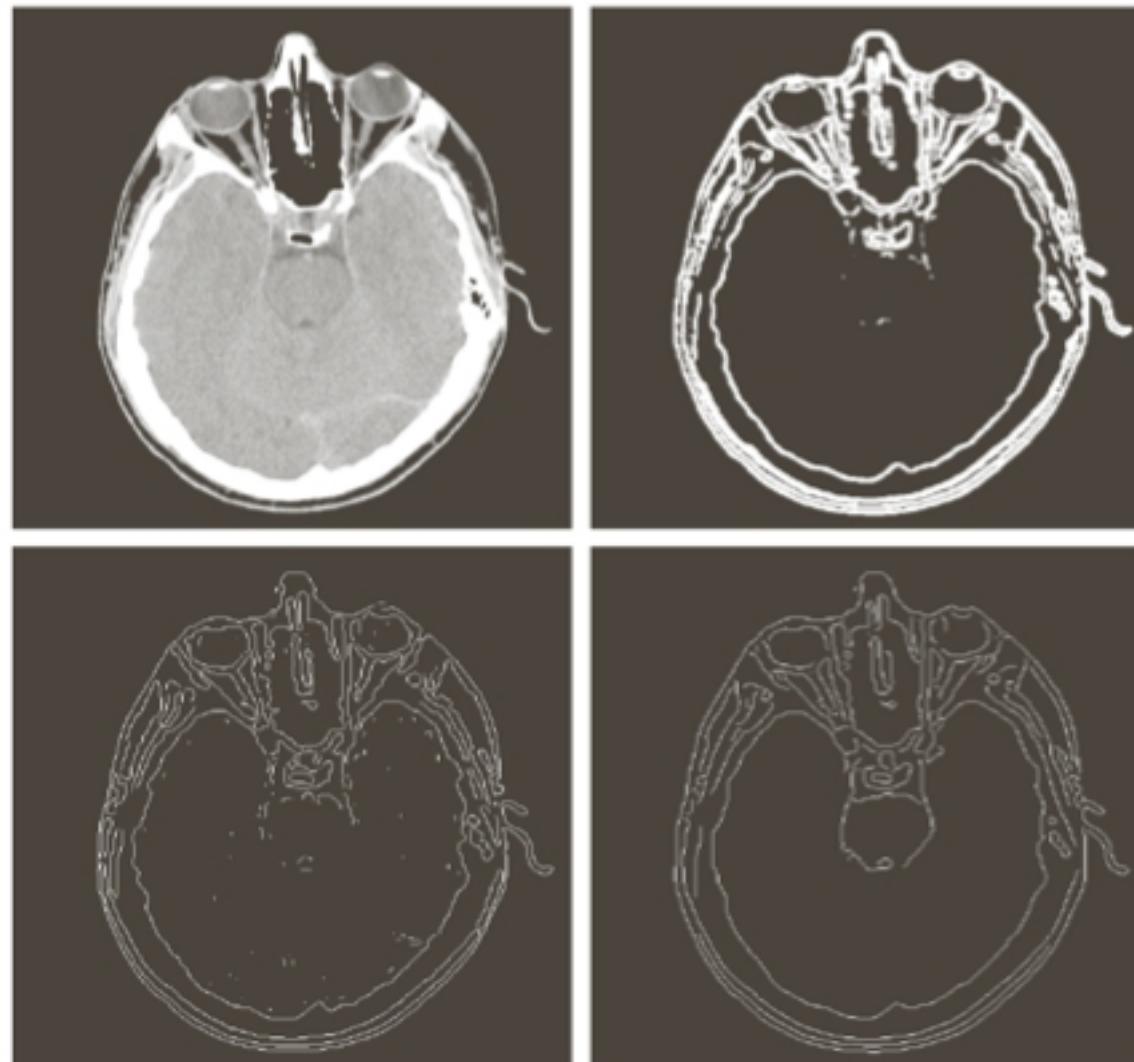


Marr

Canny



Canny -- Marr



Marr

Canny

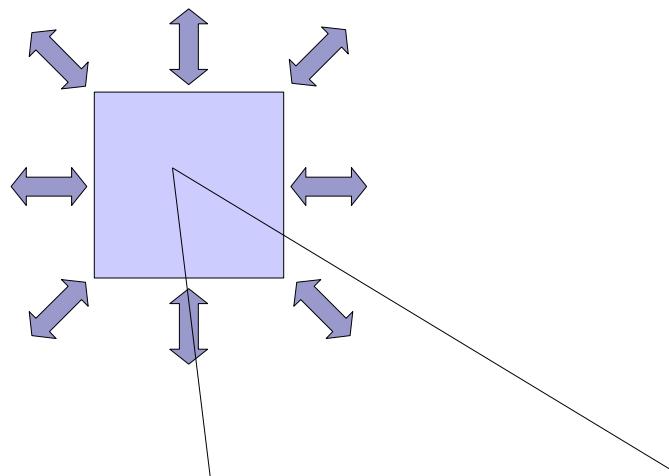


Otros puntos de interés: Esquinas



Detección de esquinas

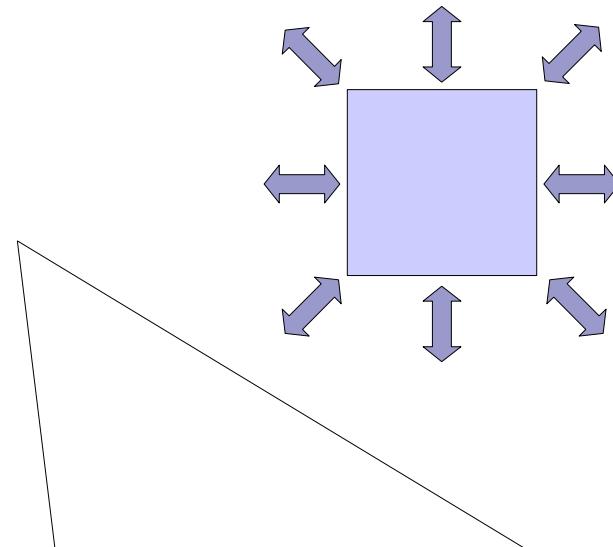
- ***Esquina***
 - Intersección de líneas de la imagen o en estructuras formadas por distintas intensidades
- ***Idea básica para su detección***
 - Analizar las intensidades en todas las direcciones de una ventana





Detección de esquinas

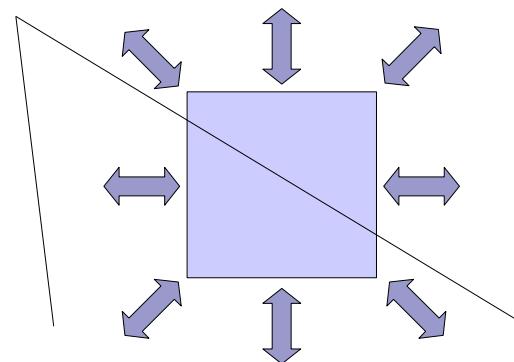
- **Zona “plana”**
 - No hay cambios significativos de intensidad en ninguna dirección





Detección de esquinas

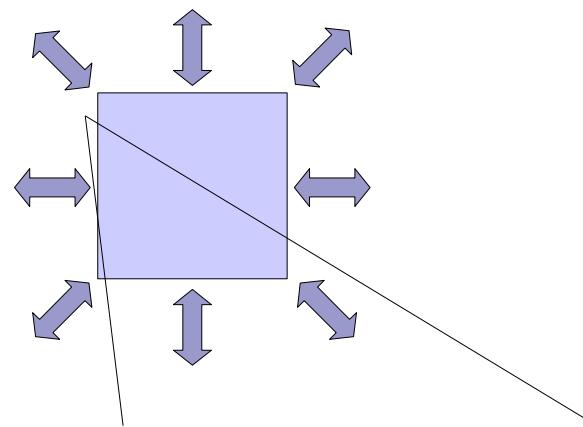
- **Borde**
 - Cambios significativos de intensidad en sólo una dirección





Detección de esquinas

- **Esquina**
 - Cambios significativos de intensidad en todas las direcciones

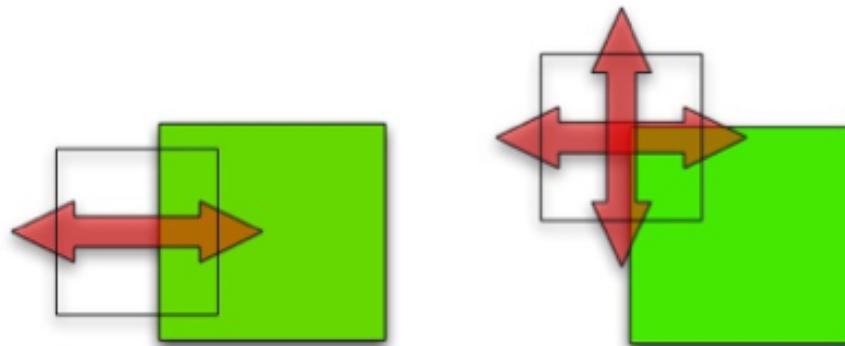


- El detector de esquinas de Harris ofrece una aproximación matemática para determinar estas tres clases

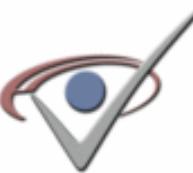


Detección de esquinas: Moravec

Superponer una ventana pequeña en la imagen. Si hay esquina, al desplazar la ventana en cualquier dirección, habrá un cambio de intensidad grande.



- Medida de esquina: menor variación de intensidad en las 8 direcciones principales $\sum_{i=1}^9 (A_i - B_i)^2$



Detección de esquinas: Moravec

- Ejemplo de cálculo de medida de esquina

		B1	B2	B3				
	A1	A2 B4	A3 B5		B6			
	A4	A5 B7	A6 B8		B9			
	A7	A8	A9					

$$V = \sum_{i=1}^9 (A_i - B_i)^2 = 2 * 255^2$$

		B1	B2	B3				
	A1	A2 B4	A3 B5		B6			
	A4	A5 B7	A6 B8		B9			
	A7	A8	A9					

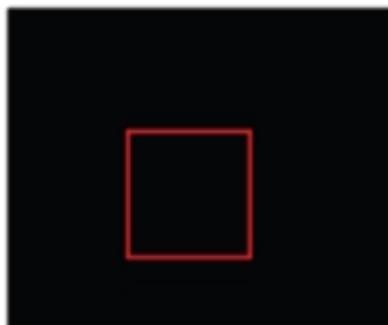
$$V = \sum_{i=1}^9 (A_i - B_i)^2 = 3 * 255^2$$

- Escogemos la menor V de entre las 8 direcciones posibles

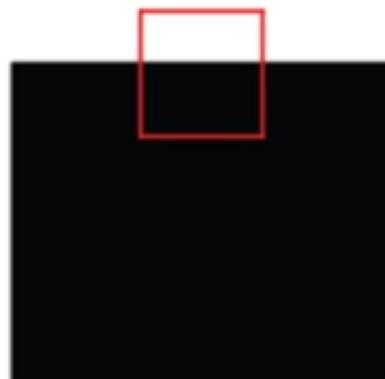


Detección de esquinas: Moravec

- ¿Por qué detecta esquinas este método?



A. Interior Region
Little intensity variation
in any direction



B. Edge
Little intensity variation
along edge, large
variation perpendicular
to edge



C. Edge
Large intensity variation
in all directions



D. Edge
Large intensity variation
in all directions



Detección de esquinas: Moravec

- **Magnitudes de salida**
- Mapa de "intensidad de esquina"

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	0	0	0	0	0	0	0	0	0	1	1	1	X	X	
X	X	0	0	0	0	0	0	1	1	0	0	1	2	1	X	X
X	X	0	0	0	0	0	0	2	1	0	0	1	1	1	X	X
X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

- Esquina es máximo local.
- Medida de esquina de punto aislado idéntica al de la esquina verdadera.
- Imposible analizar el borde de la imagen.



Detección de esquinas: Moravec

- **Limitaciones del método de Moravec**
- Respuesta anisotrópica: no es invariante a rotaciones.
- Respuesta al ruido.
- Larga respuesta a bordes en caso de imperfecciones.



Original Image

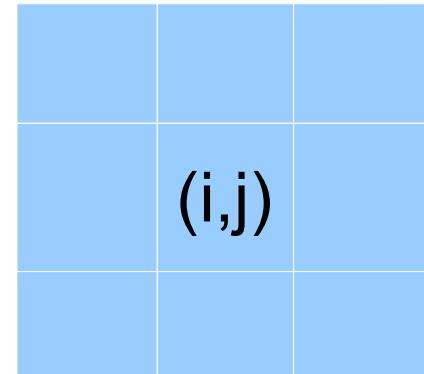


Image Rotated 30°



Detector de Esquinas: Harris

- Intuitivamente
 - Una dirección



$I(p) - I(i, j) \approx 0$ en zonas planas

$I(p) - I(i, j) \uparrow \uparrow$ en zonas con cambios de intensidad
 $p \in \mathcal{N}(i, j)$

- Varias direcciones

Derivadas

$\sum_{p \in \mathcal{N}(i, j)} |I(p) - I(\overset{\blacktriangle}{i}, j)|^2 \approx 0$ en zonas planas

$\sum_{p \in \mathcal{N}(i, j)} |I(p) - I(i, j)|^2 \uparrow \uparrow$ en zonas con cambios de intensidad



Detector de Esquinas: Harris

- Se puede realizar una aproximación:

$$I(p) = I(i+x, j+y) \approx I(i, j) + \frac{\partial}{\partial x} I(i, j)x + \frac{\partial}{\partial y} I(i, j)y$$
$$x = y = 1$$

- Entonces:

$$\sum_{p \in \mathcal{N}(i, j)} |I(p) - I(i, j)|^2 \approx \sum_{p \in \mathcal{N}(i, j)} \left(\frac{\partial}{\partial x} I(i, j) + \frac{\partial}{\partial y} I(i, j) \right)^2$$



Detector de Esquinas: Harris

- **Tensor estructural**

- Representación matricial de la estructura de niveles de gris de un punto y su vecindario

$$S(i, j) = \begin{bmatrix} \sum_{p \in \mathfrak{N}(i, j)} I_x^2(p) & \sum_{p \in \mathfrak{N}(i, j)} I_x(p)I_y(p) \\ \sum_{p \in \mathfrak{N}(i, j)} I_x(p)I_y(p) & \sum_{p \in \mathfrak{N}(i, j)} I_y^2(p) \end{bmatrix}$$

I_x, I_y derivadas en x e y

$\mathfrak{N}(i, j)$ es el vecindario del punto (i,j)

Detector de Esquinas: Harris



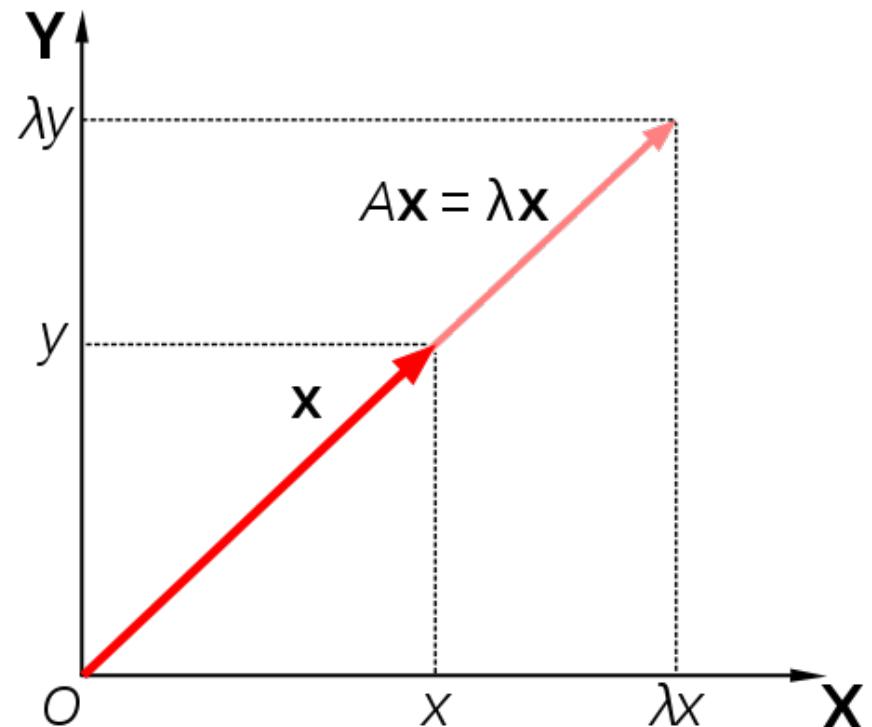
Valores y vectores propios

- Definición
 - Sea A una matriz, se dice que un vector $X \in \mathbb{R}^2 \neq 0$ es un vector propio de la matriz A si satisface
$$AX = \lambda X$$
 - para algún escalar λ , que se denomina valor propio del vector X



Valores y vectores propios

- Geométricamente
 - Bajo la transformación A , los vectores propios experimentan cambios en la magnitud y signo
 - La dirección de AX es la misma que la de X
 - El valor propio λ es la cantidad en que el vector X es estirado o encogido





Cálculo de valores y vectores propios

$$AX = \lambda X$$

$$AX = \lambda I_n X$$

$$AX - \lambda I_n X = 0$$

$$(A - \lambda I_n)X = 0$$

Tiene solución si no es invertible

$$\begin{aligned}(A - \lambda I_n)^{-1}(A - \lambda I_n)X &= (A - \lambda I_n)^{-1}0 \\ X &= 0\end{aligned}$$

$$X \neq 0 \rightarrow A - \lambda I_n = 0 \blacktriangleleft$$

Una matriz no es invertible si $\det(A - \lambda I_n) = 0$

Polinomio característico de A
Sus raíces son los valores propios de A

Valores y vectores propios del tensor estructural

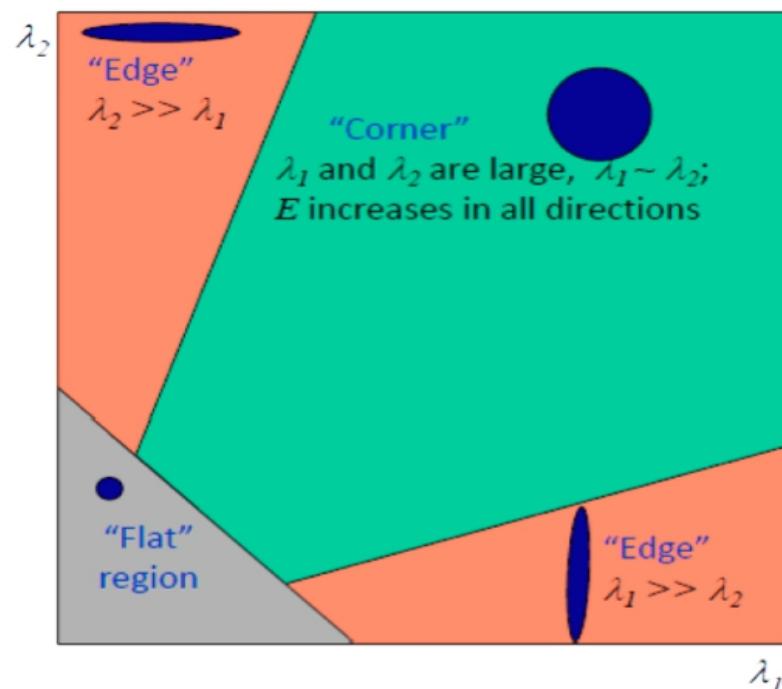
- Vectores propios
 - $e_1 \rightarrow$ dirección normal al borde
 - $e_2 \rightarrow$ dirección tangente al borde
- Valores propios
 - $\lambda_1 \rightarrow$ fuerza del borde en la dirección normal
 - $\lambda_2 \rightarrow$ fuerza del borde en la dirección tangente

Valores y vectores propios del tensor estructural

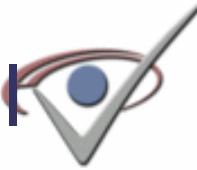


Significado

- Si $\lambda_1 = \lambda_2 = 0 \rightarrow$ superficie plana
- Si $\lambda_1 > 0$ y $\lambda_2 = 0 \rightarrow$ borde
- Si $\lambda_1 > 0$ y $\lambda_2 > 0 \rightarrow$ esquina

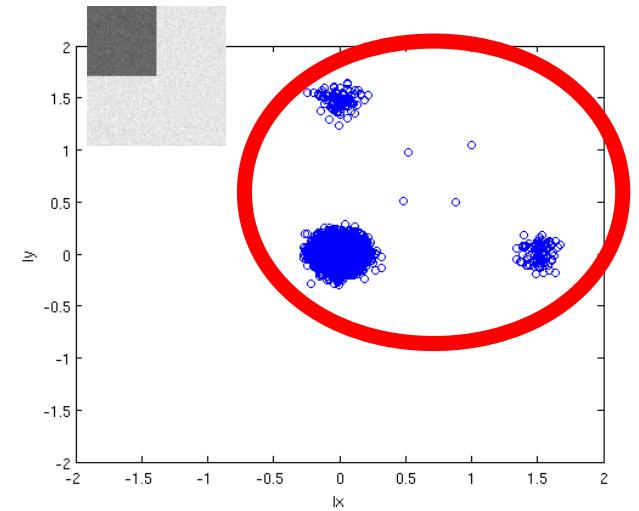
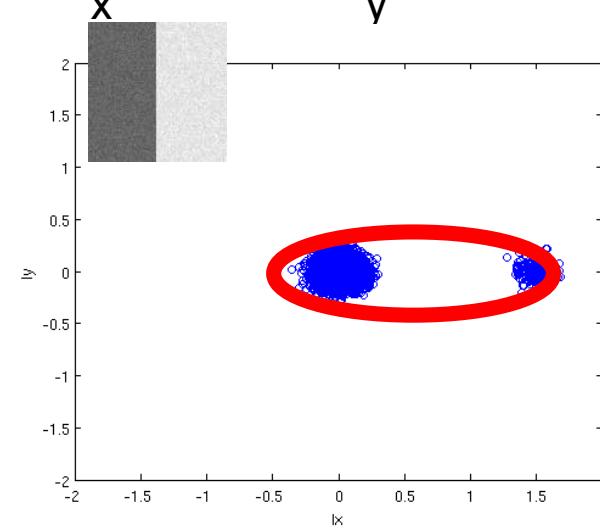
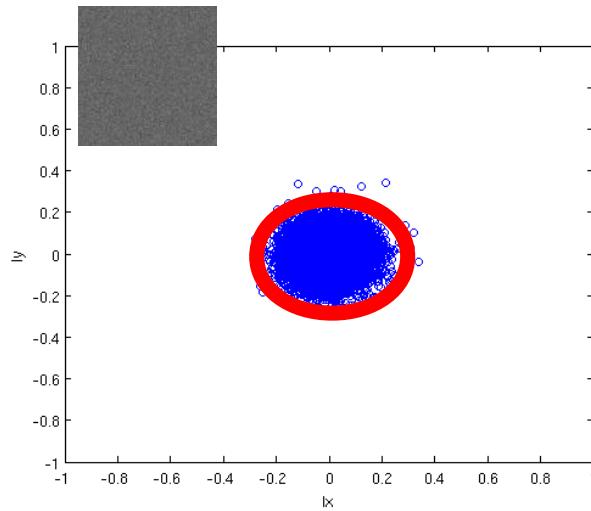


Valores y vectores propios del tensor estructural

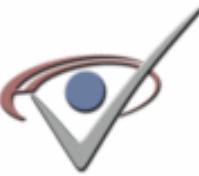


- Gráficamente

- Distribución de I_x frente I_y



- Ajustar elipse a los puntos
 - $\lambda_1 \rightarrow$ radio mayor de la elipse
 - $\lambda_2 \rightarrow$ radio menor de la elipse



Valores propios del tensor estructural

- En la práctica calcular los valores propios es una operación costosa
- Aproximación

$$M = \det(S) - \kappa \operatorname{trace}(S)^2 = \lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2$$
$$\kappa \in [0.04, 0.06]$$



Algoritmo Harris

1. Calcular las derivadas de la imagen

$$I_x = G_x * I \quad I_y = G_y * I$$

2. Para cada punto (i,j) y un vecindario de tamaño N , calcular

$$S(i, j) = \begin{bmatrix} \sum_{p \in \mathcal{N}(i, j)} I_x^2(p) & \sum_{p \in \mathcal{N}(i, j)} I_x(p)I_y(p) \\ \sum_{p \in \mathcal{N}(i, j)} I_x(p)I_y(p) & \sum_{p \in \mathcal{N}(i, j)} I_y^2(p) \end{bmatrix}$$

$$M(i, j) = \det(S(i, j)) - \kappa \operatorname{trace}(S(i, j))^2$$

3. Considerar esquina todos aquellos puntos con $M(i,j) > t$

4. Aplicar supresión no máxima para seleccionar una esquina en cada vecindario



Algoritmo Harris

En la realidad se suele utilizar:

$$M = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} I_x^2(\mathbf{x}, \sigma_D) & I_x(\mathbf{x}, \sigma_D)I_y(\mathbf{x}, \sigma_D) \\ I_x(\mathbf{x}, \sigma_D)I_y(\mathbf{x}, \sigma_D) & I_y^2(\mathbf{x}, \sigma_D) \end{bmatrix}$$

Donde $g(\sigma)$ es una gaussiana y:

$$I_x(\mathbf{x}, \sigma_D) = \frac{\partial}{\partial x} g(\sigma_D) * I(\mathbf{x})$$

$$I_y(\mathbf{x}, \sigma_D) = \frac{\partial}{\partial y} g(\sigma_D) * I(\mathbf{x})$$



Algoritmo Harris

En la realidad se suele utilizar:

$$M = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} I_x^2(\mathbf{x}, \sigma_D) & I_x(\mathbf{x}, \sigma_D)I_y(\mathbf{x}, \sigma_D) \\ I_x(\mathbf{x}, \sigma_D)I_y(\mathbf{x}, \sigma_D) & I_y^2(\mathbf{x}, \sigma_D) \end{bmatrix}$$

Donde $g(\sigma)$ es una gaussiana y:

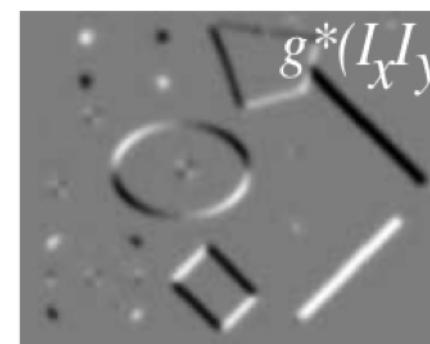
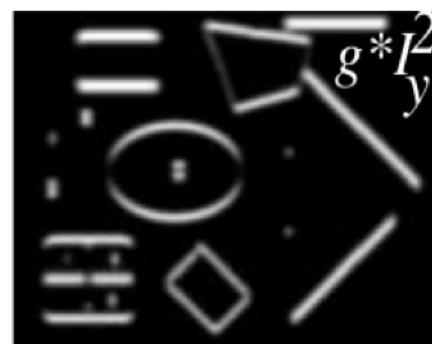
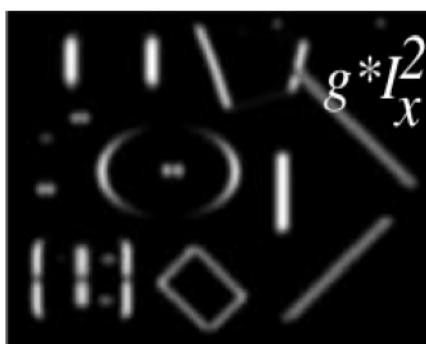
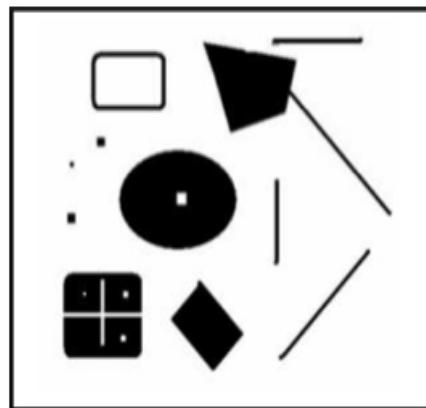
$$I_x(\mathbf{x}, \sigma_D) = \frac{\partial}{\partial x} g(\sigma_D) * I(\mathbf{x})$$

$$I_y(\mathbf{x}, \sigma_D) = \frac{\partial}{\partial y} g(\sigma_D) * I(\mathbf{x})$$

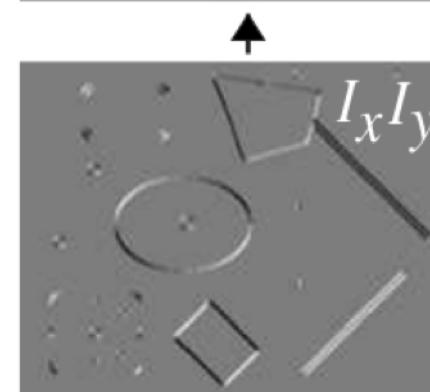
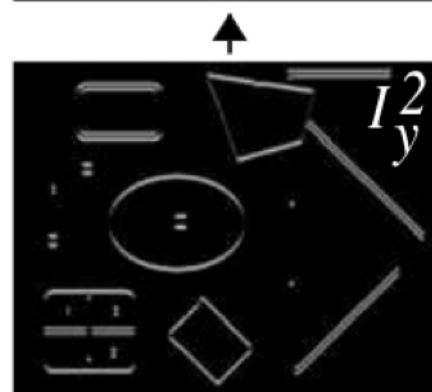
Las derivadas de la imagen están computadas tras la aplicación de un kernel gaussiano de parámetro σ_D (**escala de diferenciación**). Luego las derivadas se ponderan suavizándolas con una gaussiana de parámetro σ_I (**escala de integración**). El objetivo de la primera gaussiana es establecer el tamaño de las características a buscar (de ahí el nombre de diferenciación). Por otro lado, la segunda gaussiana trata de unir características adyacentes (integración de regiones)



Algoritmo Harris



(e) Resultado Final





Algoritmo Harris: Inconvenientes

- Necesario un umbral
- Los píxeles cuyas funciones de respuesta de esquina superan el umbral forman regiones alrededor de la esquina: no se identifican puntos sino regiones



Resaltados en negro los píxeles que superan el umbral de la función de respuesta de esquina del detector de Harris & Plessey.

El umbral en cada imagen es diez veces el de la imagen anterior.

Todo el proceso se aplica sobre la imagen inicial en niveles de gris.





Algoritmo Harris: Soluciones

• Solución a la detección de regiones en lugar de puntos:

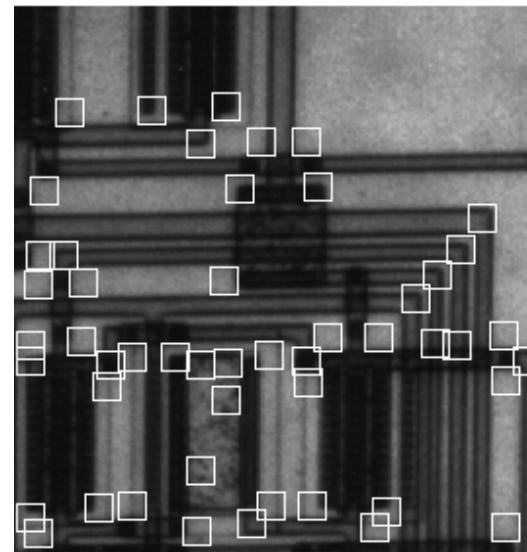
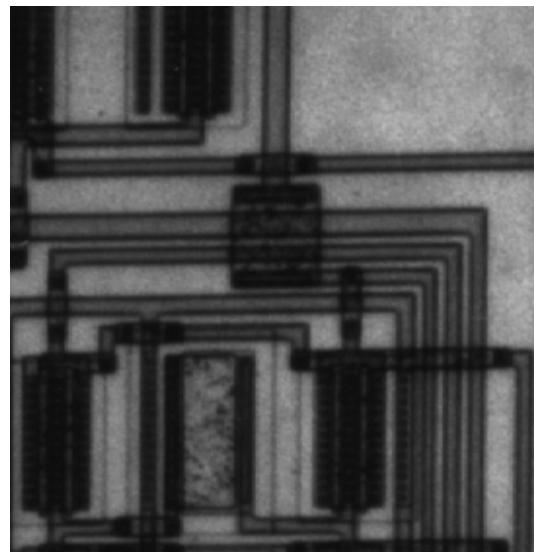
- Seleccionar como esquinas sólo aquellos píxeles cuyo valor de la función de respuesta de esquina sea un máximo local (el valor máximo en un vecindario de por ejemplo 3x3 píxeles).



- Adicionalmente se pueden descartar aquellos píxeles que cumplen el criterio anterior pero estén cercanos a otro píxel previamente seleccionado como esquina. La distancia mínima admisible es un valor escogido empíricamente.



Detector de esquinas de Harris





Detector de Esquinas: Susan

- SUSAN (Smallest Univalue Segment Assimilating Nucleus)
- Detector con invariancia parcial a rotaciones, cambios de escala, variaciones de iluminación y ruido en la imagen.
- Robustez al ruido por no requerir el cálculo de derivadas parciales de la imagen dada.
- Eficiencia elevada por no requerir cálculo de convoluciones.

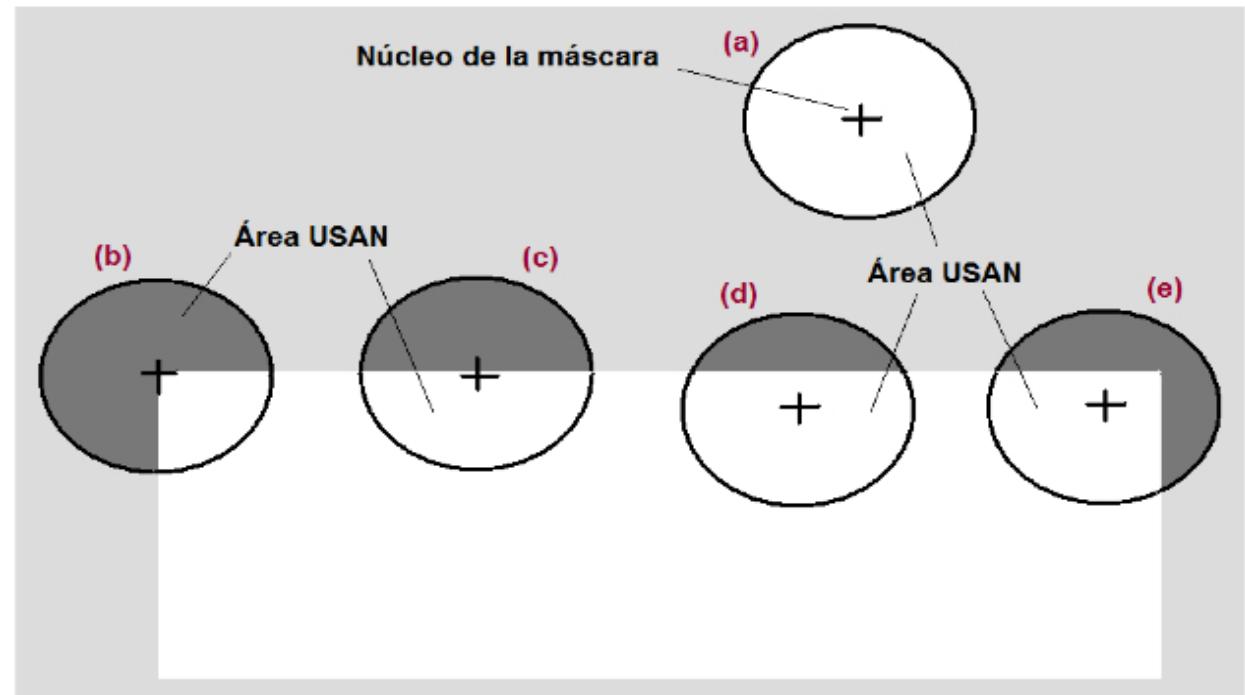
Principio Básico:

Cada punto de la imagen tiene asociado un área local con una luminosidad similar.



Detector de Esquinas: Susan

La luminosidad de cada pixel dentro de la máscara se compara con la luminosidad del núcleo de la máscara (centro de la máscara), cuyo tamaño es de 1 pixel.



Se define el USAN (Univalue Segment Assimilating Nucleus) como el área de la máscara que tiene la misma luminosidad que el núcleo

El área USAN es máxima cuando el núcleo se encuentra en una región lisa de la superficie de la imagen (Figura 2a), es la mitad del máximo muy cerca de un borde recto (Figura 2c) y es menos de la mitad del máximo en una esquina (Figura 2e).

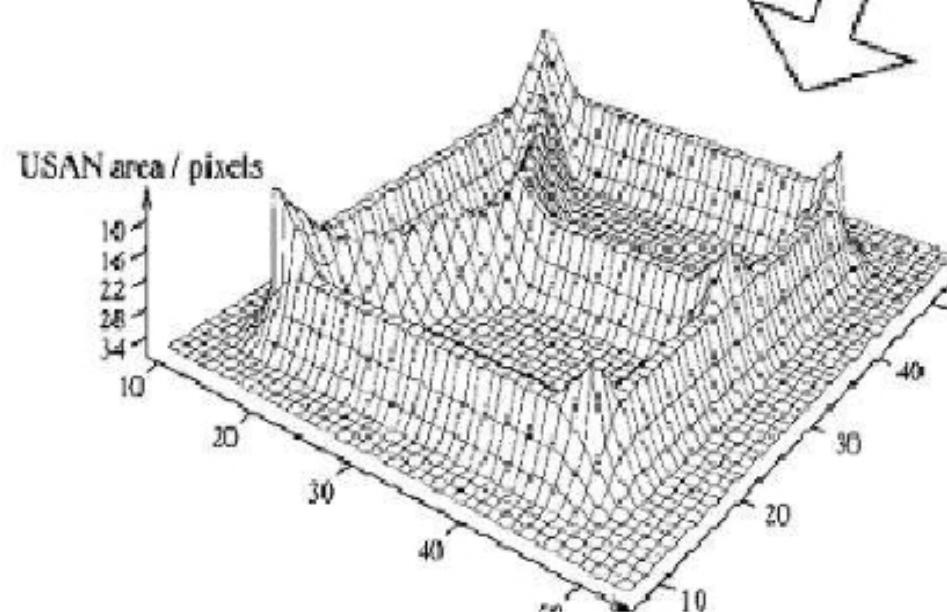


Detector de Esquinas: Susan

Parte de la imagen original



Encuentra el área SUSAN
para cada posición de la
imagen





Detector de Esquinas: Susan

La máscara se sitúa en cada píxel de la imagen y se comparan las luminosidades de todos los pixeles dentro de la máscara con la del núcleo mediante la siguiente fórmula:

$$c(\vec{r}, \vec{r}_0) = \begin{cases} 1 & \text{if } |I(\vec{r}) - I(\vec{r}_0)| \leq t \\ 0 & \text{if } |I(\vec{r}) - I(\vec{r}_0)| > t \end{cases}$$

Siendo **r0** la posición del núcleo, **r** la posición de cualquier punto dentro de la máscara, **I(r0)** la luminosidad del núcleo, **I®** la luminosidad de cualquier píxel, **t** es la diferencia de umbral de luminosidad y **c** la salida de la comparación, que es 1 cuando la diferencia de luminosidad entre el pixel dentro de la máscara y el núcleo no supera un umbral **t**.

T determina la cantidad de ruido ignorado



Detector de Esquinas: Susan

La comparación se realiza para cada píxel dentro de la máscara, y sumando todos se obtiene n, que es el número de píxeles totales del área USAN.

$$n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r}, \vec{r}_0)$$

El parámetro n es comparado con un umbral fijo g (umbral geométrico) que está fijado en 3/4 el valor máximo que puede tomar n.

La respuesta de bordes inicial R(r0) se crea mediante la siguiente fórmula

$$R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0) & \text{if } n(\vec{r}_0) < g \\ 0 & \text{otherwise} \end{cases}$$



Detector de Esquinas: Susan

La comparación se realiza para cada píxel dentro de la máscara, y sumando todos se obtiene n, que es el número de píxeles totales del área USAN.

$$n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r}, \vec{r}_0)$$

El parámetro n es comparado con un umbral fijo g (umbral geométrico) que está fijado en 3/4 (valor máximo que puede tomar n)

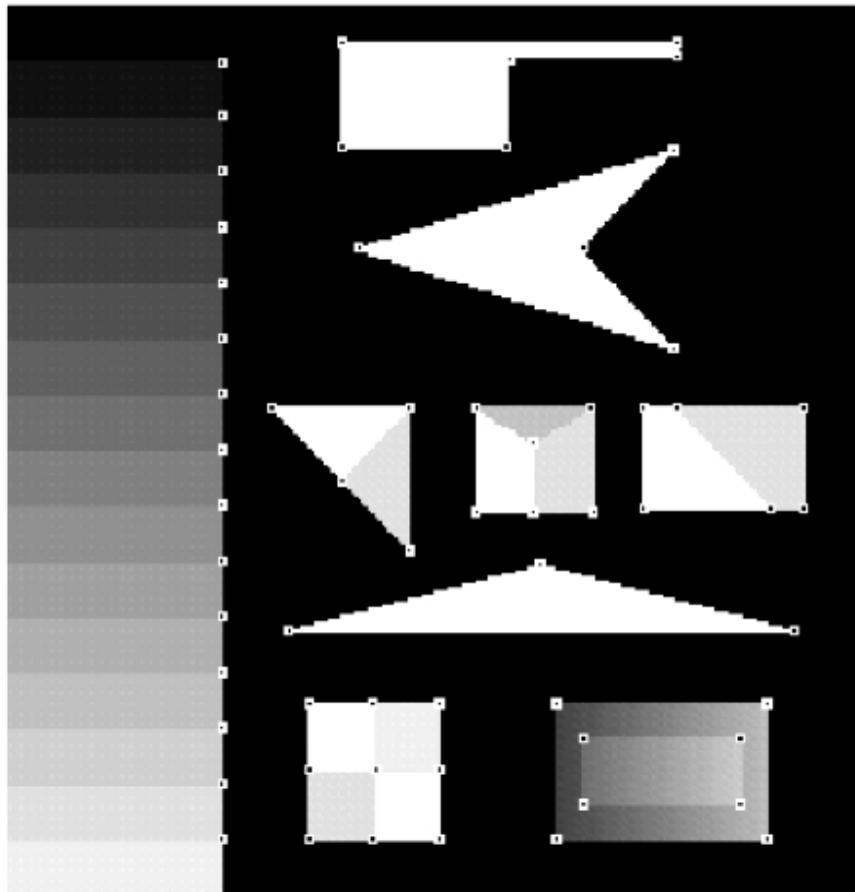
La respuesta de bordes inicial R(r0) se crea mediante la siguiente fórmula

$$R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0) & \text{if } n(\vec{r}_0) < g \\ 0 & \text{otherwise} \end{cases}$$

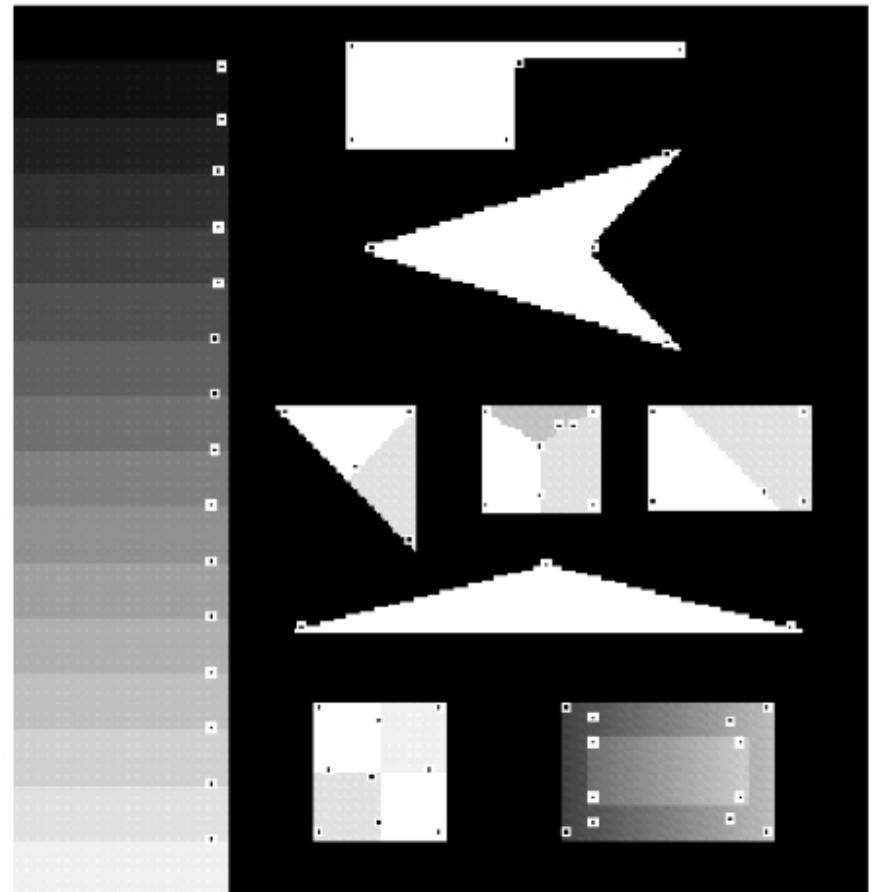
Cuanto menor es el área USAN, mayor es la respuesta de bordes.



Detector de Esquinas: Susan



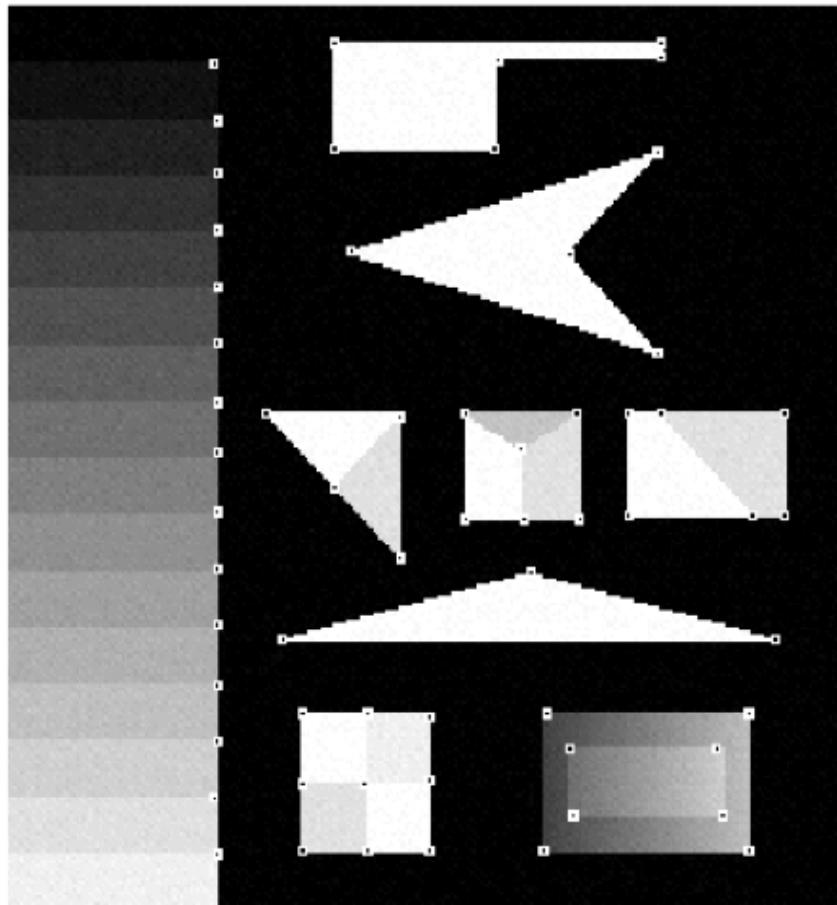
Ejemplo detector SUSAN
(imagen sintética)



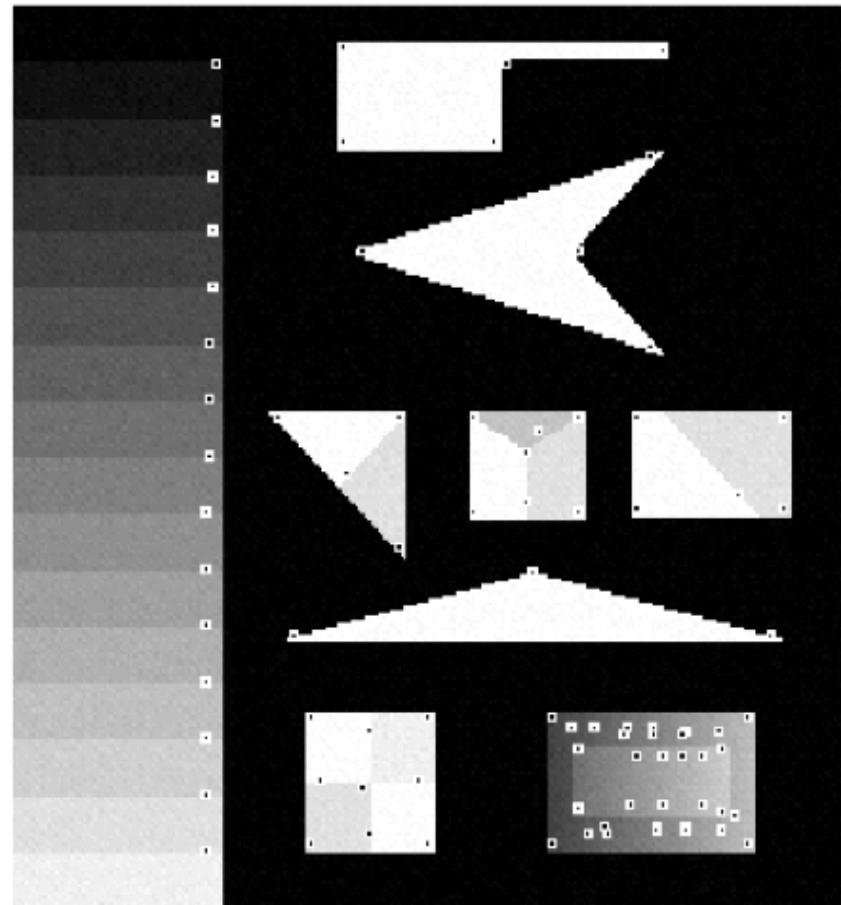
Ejemplo detector Harris & Stephens
(imagen sintética)



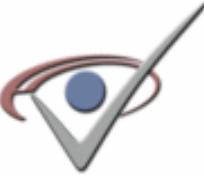
Detector de Esquinas: Susan



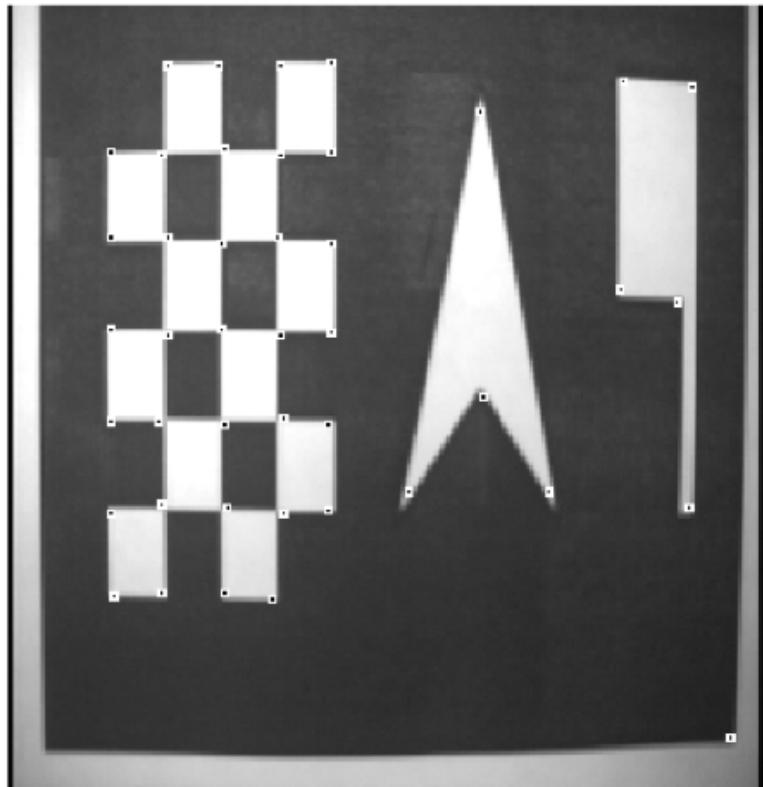
Ejemplo detector SUSAN
(imagen sintética con ruido)



Ejemplo detector Harris & Stephens
(imagen sintética con ruido)



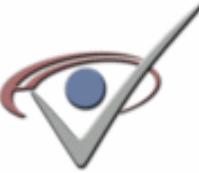
Detector de Esquinas: Susan



Ejemplos detector SUSAN (imágenes reales)



Detectores de Puntos Singulares MultiResolución



Detectores Multi-Resolución

- Para evitar problemas ante cambios de escala hay que usar técnicas de multi-resolución en el detector. La idea es que los mismos puntos se repitan en imágenes similares.
- Multi-resolución
 - Pirámides de Gauss y Laplace
 - Scale space
 - Scale space submuestreado o escalonado



Detectores Multi-Resolución

- Pirámides de Gauss y Laplace
 - Idea: producir una pirámide donde la siguiente imagen tenga la mitad del tamaño de la anterior





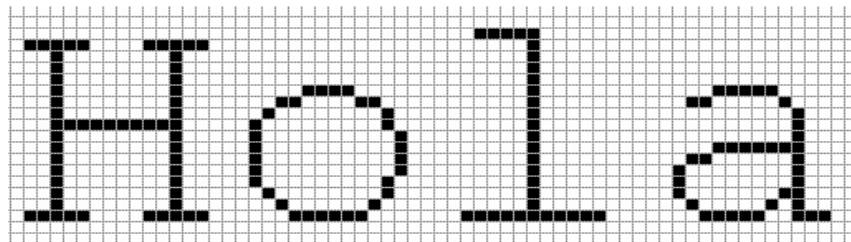
Detectores Multi-Resolución

- No se puede “llegar y achicar la imagen”
(submuestrear: $I'(x,y)=I(2x,2y)$)
 - La imagen discreta es un muestreo, con alguna frecuencia f , de una imagen continua
 - Submuestrear la imagen discreta equivale a muestrear la imagen continua con frecuencia $f/2$ (con menos puntos) => se puede producir aliasing
- Es necesario filtrar pasa bajos la imagen antes de submuestrearla

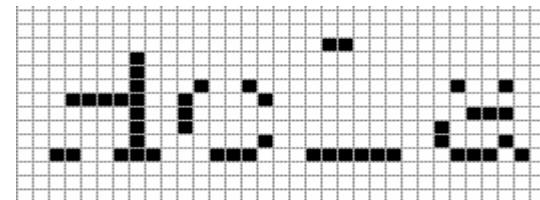


Detectores Multi-Resolución

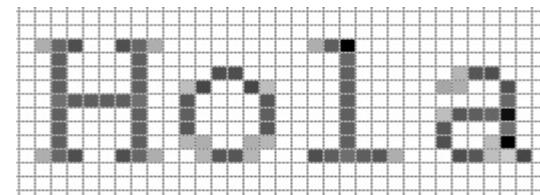
Imagen original

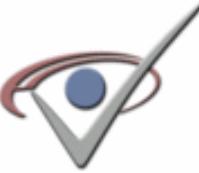


Sólo submuestreo



Filtrado + submuestreo





Detectores Multi-Resolución

- Scale space: Corresponde a ir bajando al ancho de banda de una imagen continua en función de un parámetro continuo σ

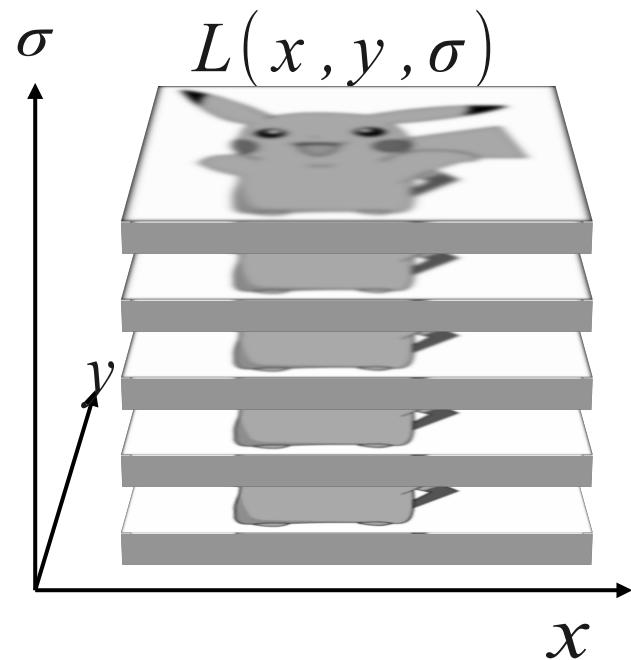
$$L(x, y, \sigma) = I(x, y) * N(x, y, \sigma)$$

- La imagen se ve más borrosa a medida que aumenta σ



Detectores Multi-Resolución

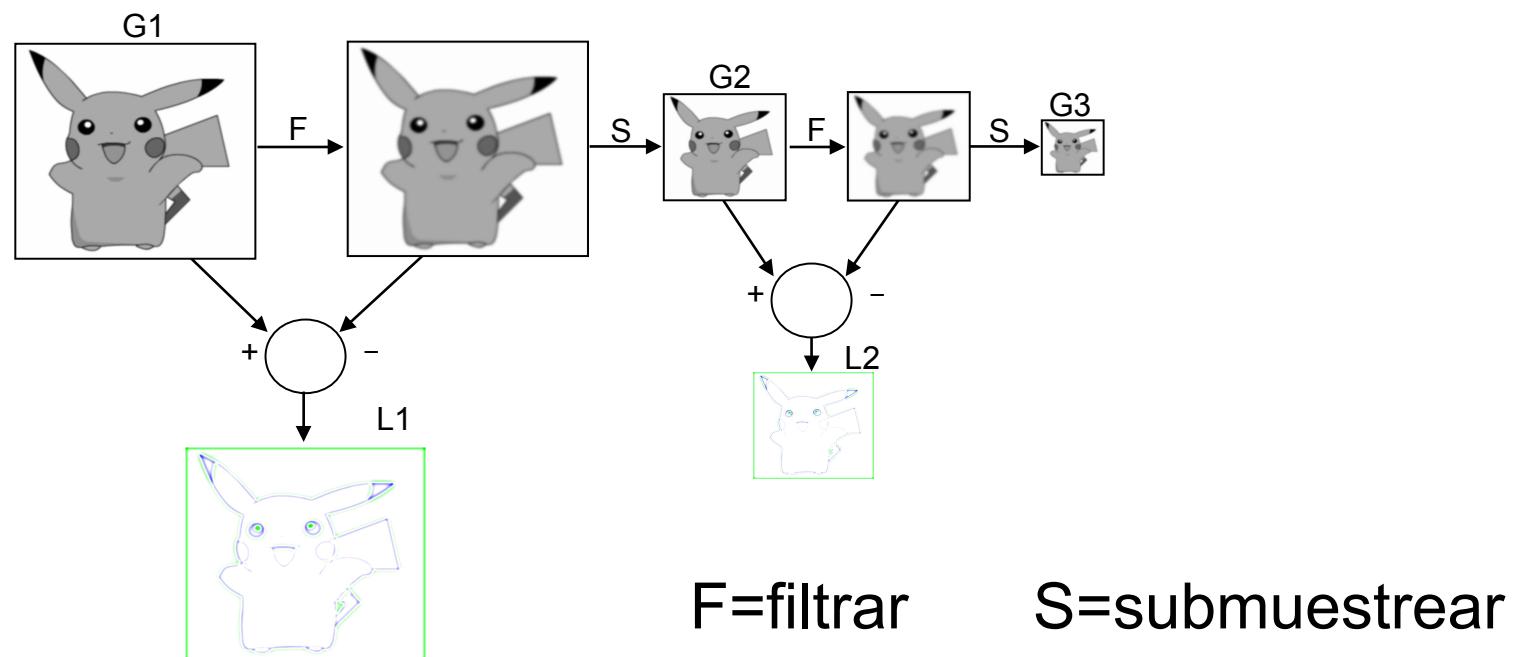
- Este procedimiento permite generarle una dimensión extra a la imagen
- Se van perdiendo los detalles gradualmente)
 - No hay submuestreo, la imagen es continua
 - La pérdida de detalles es continua y no avanza solamente por múltiplos de 2





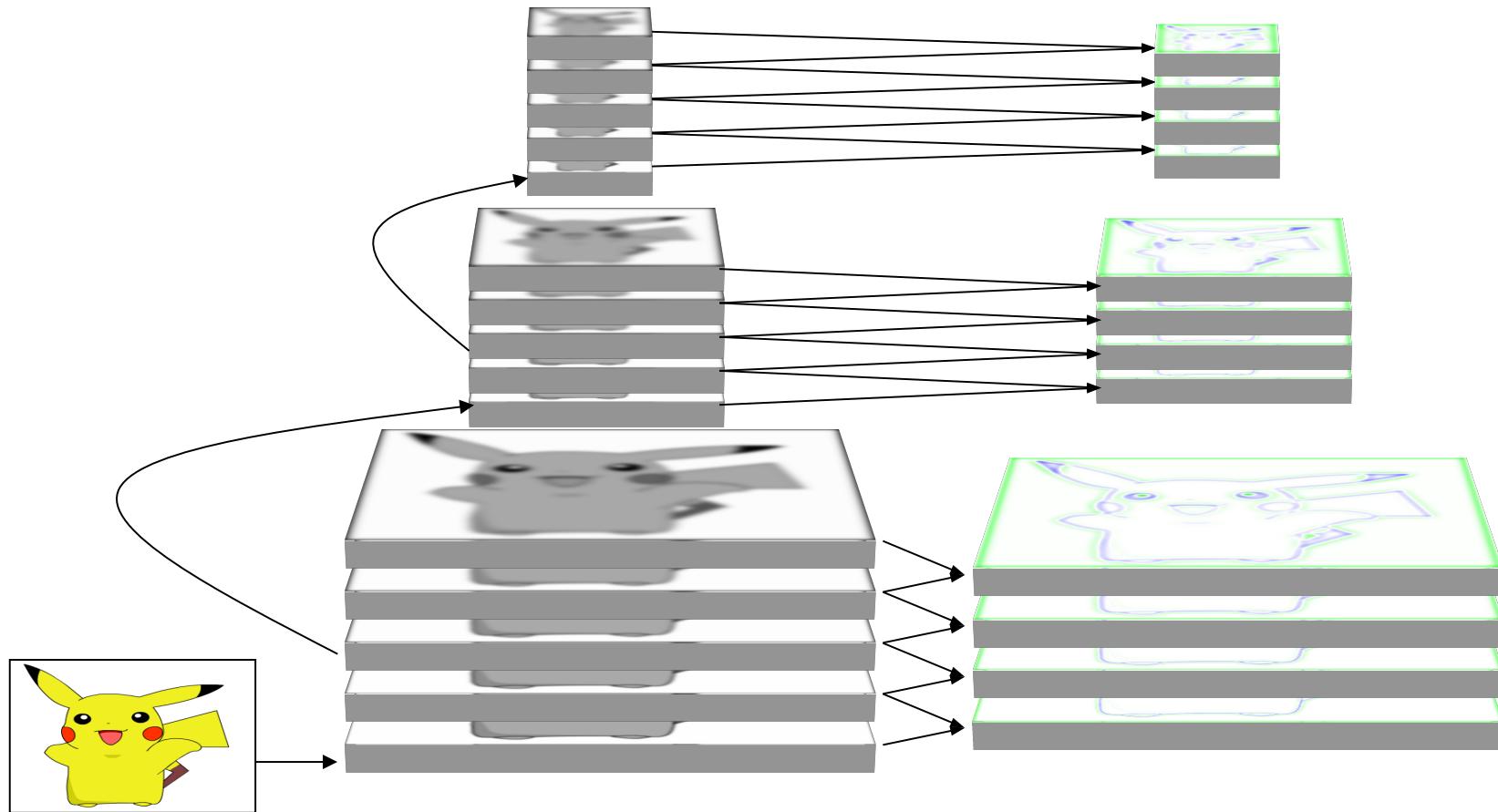
Detectores Multi-Resolución

- Se puede generar otra pirámide: la pirámide de Laplace, si se va guardando la información que se pierde (detalles de la imagen)





Detectores Multi-Resolución





Detectores Multi-Resolución

- Varios tipos de detectores que se basan en esta idea:
 - SDoG (Scale)
 - Harris-Laplace
 - Sift
 - Surf
 - U-Surf



Detector: Sift

- La técnica SIFT (Scale Invariant Feature Transform) es un método para extraer características de imágenes que pueden ser usadas para hacer corresponder diferentes vistas de un mismo objeto o escena.
- Las características son invariantes a la escala y a la rotación; además, también presentan cierta robustez frente a distorsiones afines como el cambio de perspectiva, la adición de ruido o la modificación de la iluminación.
- Las propiedades detectadas por SIFT son muy distintivas lo que permite usar esta técnica para el reconocimiento de objetos.
- El coste computacional de esta metodología se minimiza usando una aproximación en cascada, es decir, las operaciones más caras se aplican únicamente en aquellas localizaciones que pasan un test inicial.



Detector: Sift

SIFT consta de cuatro pasos:

- 1- Detección de características invariantes a la escala.
- 2- Localización de puntos de interés.
 - 2.1- Eliminación de puntos de bajo contraste
 - 2.2- Eliminación de puntos en los bordes
- 3- Asignación de la orientación
- 4- Descriptor de puntos de interés



Sift: Detección de Características

Objetivo: Se detectan los puntos de interés a diferentes escalas.

Se aplica: Diferencia de Gaussianas (DoG) que es una buena aproximación al Laplaciano del Gaussiano (LoG). La Diferencia de Gaussianas $D(x, y, \sigma)$ se define como:

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma)$$

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$$



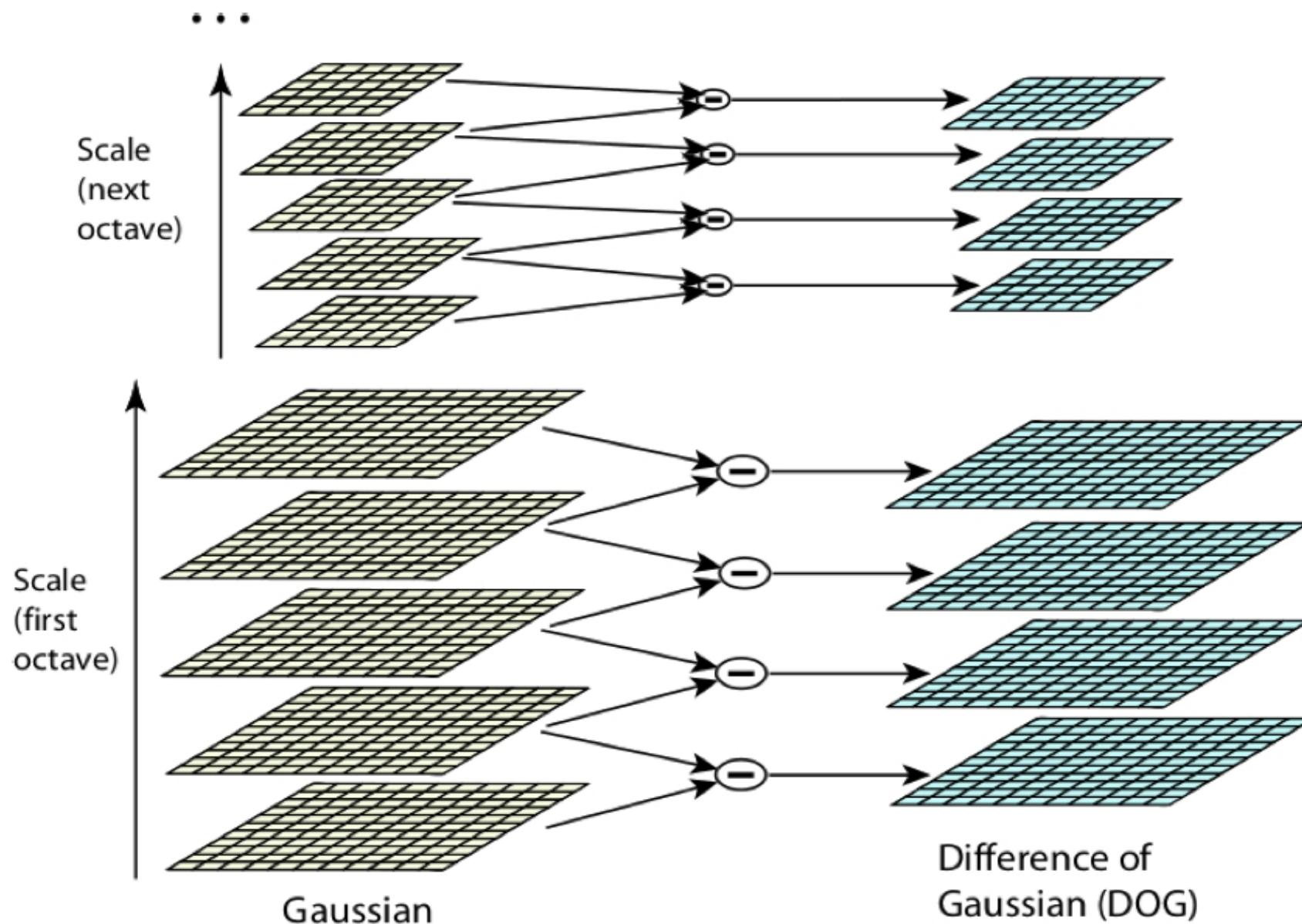
Sift: Detección de Características

El primer paso consiste en convolucionar la imagen con gaussianas. Este procedimiento se realiza con diferentes escalas de la imagen y de la gaussiana; **cada escala utilizada recibe el nombre de octava.**

- Se comienza con la imagen original y una gaussiana de gran tamaño.
- Cuando pasamos a una octava inferior, dividimos por dos el valor de σ y reducimos a la mitad la escala de la imagen.
- A su vez, para cada octava, empleamos diferentes valores de σ de forma que se obtiene un número fijo de imágenes por octava.

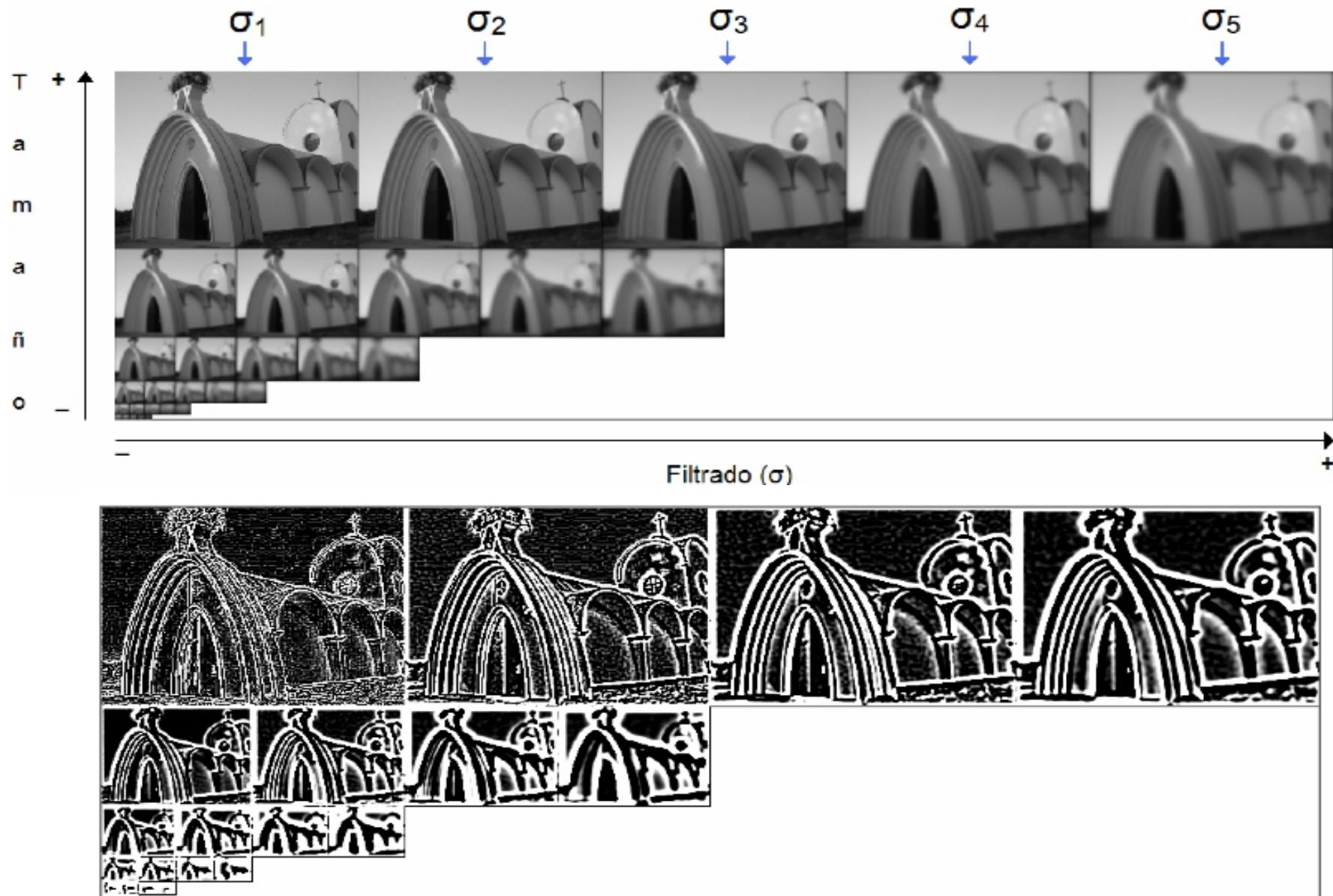


Sift: Detección de Características





Sift: Detección de Características





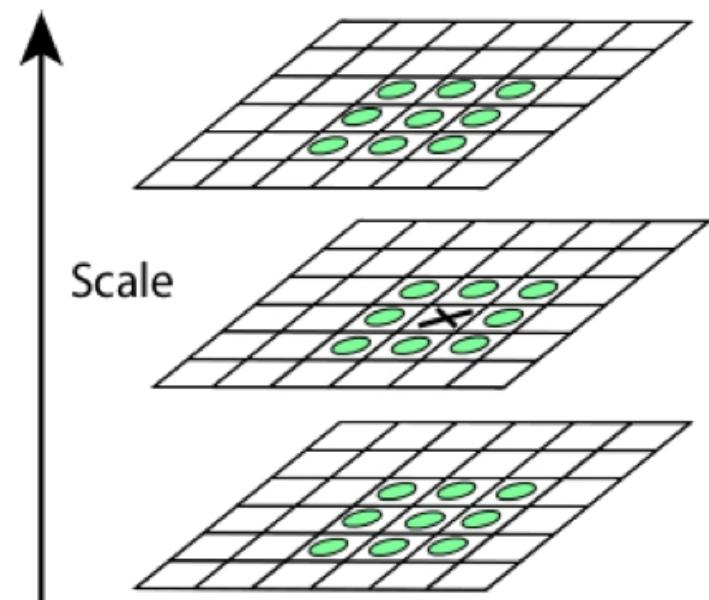
Sift: Detección de Características

Una vez obtenidas las imágenes de la DoG, se realiza una supresión no máxima que estrecha los bordes.

- necesario contar con la orientación de los bordes que podemos obtener usando un detector de bordes de primera derivada como Sobel.

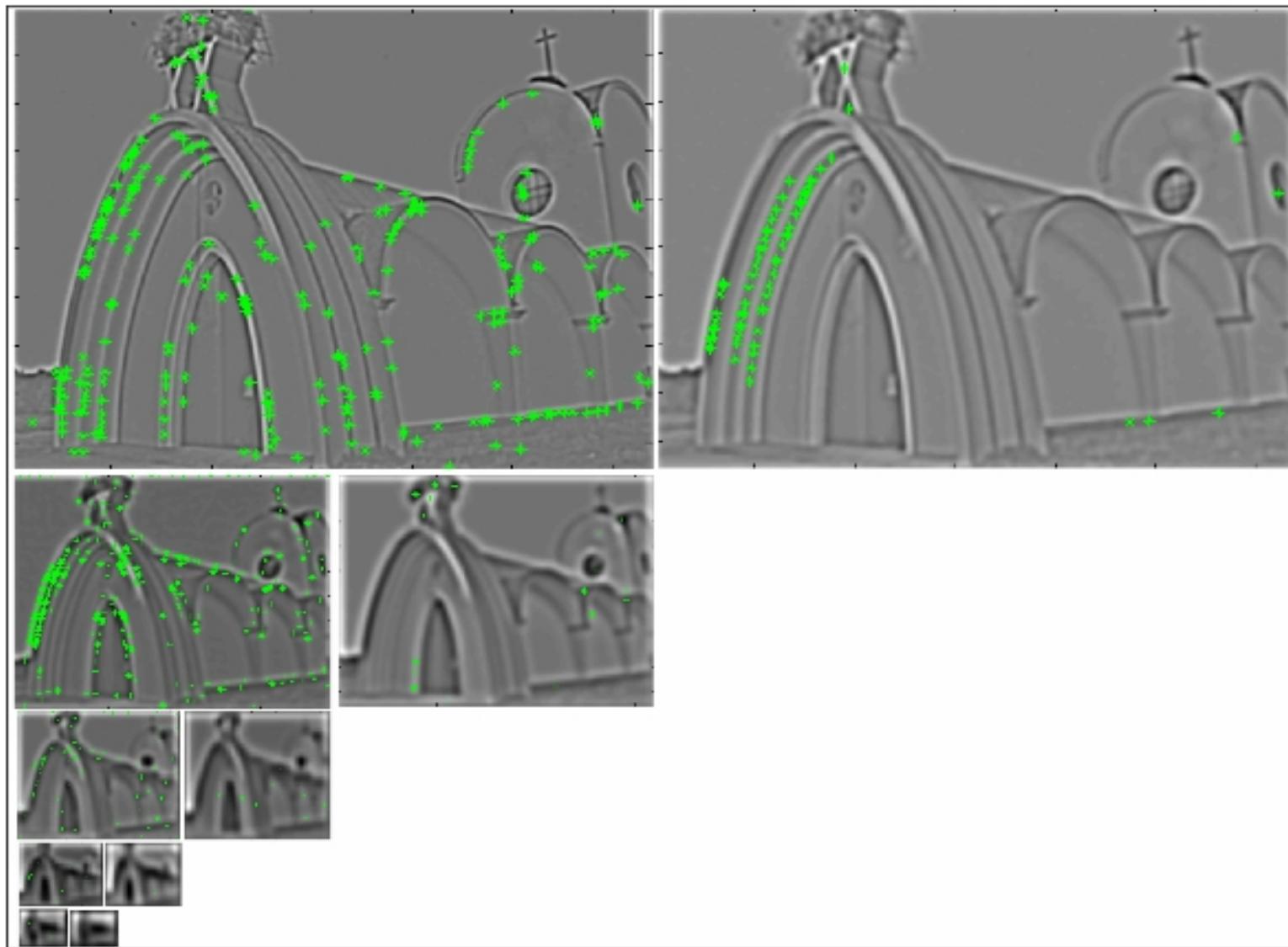
A continuación, los **candidatos a puntos de interés** se identifican como los mínimos y máximos locales buscando entre los 26 vecinos.

- Los 26-vecinos de un conjunto de imágenes son los 8-vecinos de la misma escala y los nueve píxeles vecinos de las escalas posterior y anterior dentro de la misma octava





Sift: Detección de Características





Sift: Localización Puntos de Interés

En el proceso anterior identificamos los extremos en el espacio de escala como posibles candidatos; sin embargo, **estos son demasiados y debemos filtrarlos.**

El siguiente paso del algoritmo consiste en **realizar un ajuste detallado de los datos cercanos** para obtener la localización, la escala y el ratio de las curvas principales.

Esta información **permite rechazar** los puntos:

- que tienen bajo contraste (y, por tanto, son sensibles al ruido).
- que están localizados a lo largo de un borde.



Sift: Localización Puntos de Interés

Eliminación de puntos de bajo contraste

Para descartar los puntos de bajo contraste, primero debemos determinar su localización.

- En un principio se usaba la posición del propio extremo local, pero resulta mucho más eficaz calcular la posición interpolada del extremo. Uso de la expansión de Taylor de orden 2 de la función de la DoG $D(x, y, \sigma)$ con el punto candidato como origen:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$



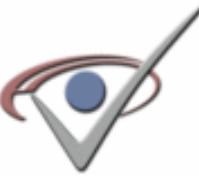
Sift: Localización Puntos de Interés

La localización del extremo \hat{x} se determina tomando la derivada respecto a x de esta función e igualándola a cero:

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$$

- Descartamos aquellos valores de \hat{x} tal que $|D(\hat{x})| < T$. En la literatura se suele utilizar un valor de $T=0,03$ para imágenes cuya intensidad está comprendida en el intervalo $[0, 1]$.

A partir de ahora, la localización de los puntos va a ser siempre interpolada y, por tanto, su escala también.



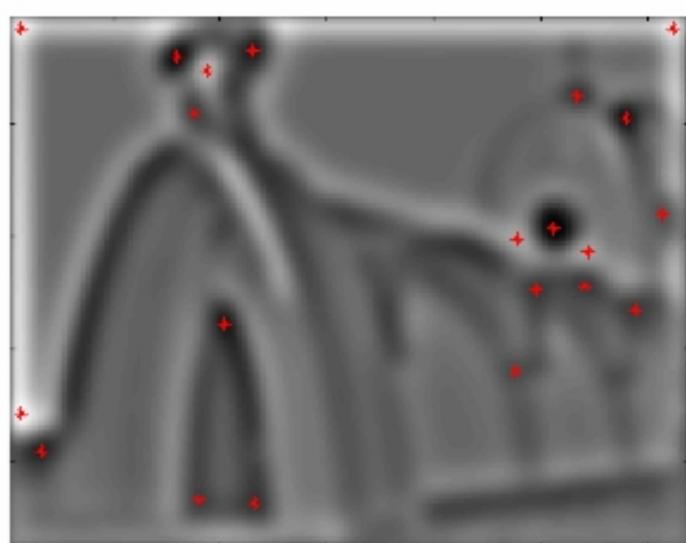
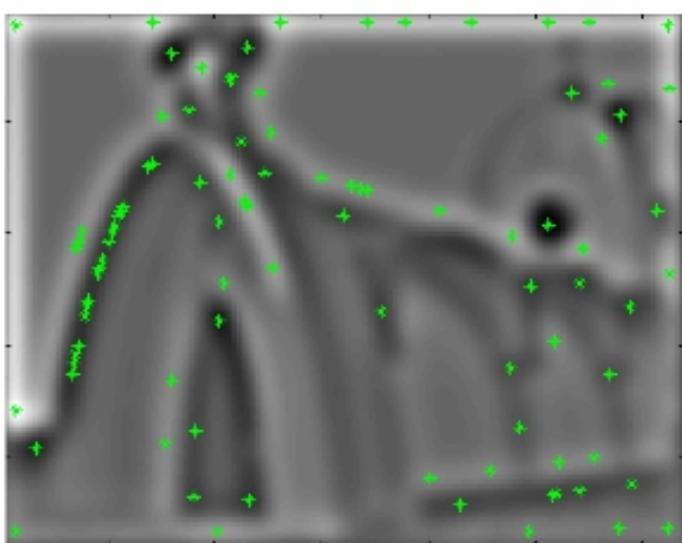
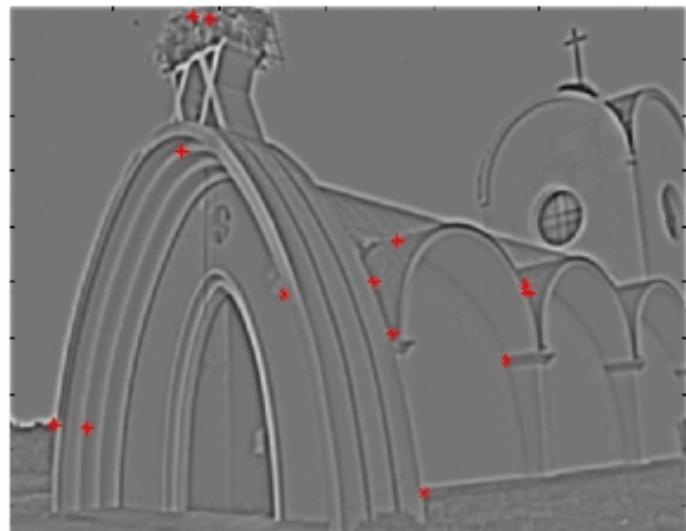
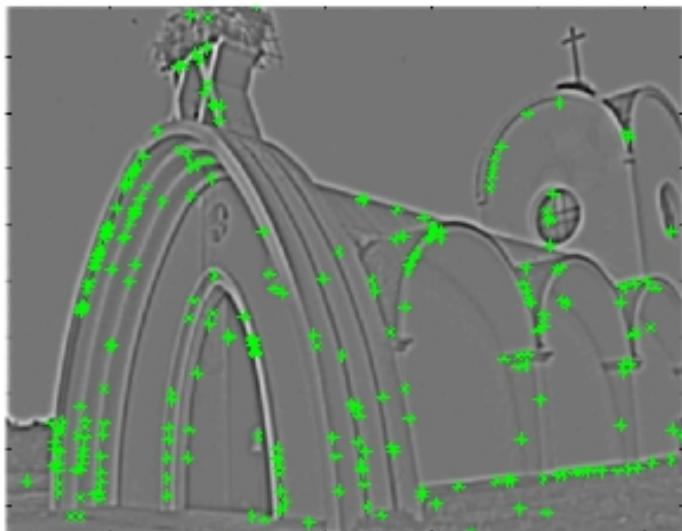
Sift: Localización Puntos de Interés

Eliminación de puntos Borde

No nos interesan los puntos obtenidos que se sitúen en bordes y que suelen ser detectados por DoG. La solución consiste en analizar si estos no pertenecen a una esquina para descartarlos.

- Se pueden utilizar diferentes algoritmos de detección de esquinas.
- El Sift normalmente utiliza Harris.

Sift: Detección de Características





Sift: Localización Puntos de Interés

Asignación de Orientación

El objetivo de este paso del algoritmo es asignar una o más direcciones a cada punto de interés.

- Conseguiremos invarianza rotacional.

Se toma la imagen suavizada $L(x,y, \sigma)$ para el valor de σ que mejor aproxima el valor interpolado de la escala de cada punto de interés. **Operamos independientemente de la escala.**

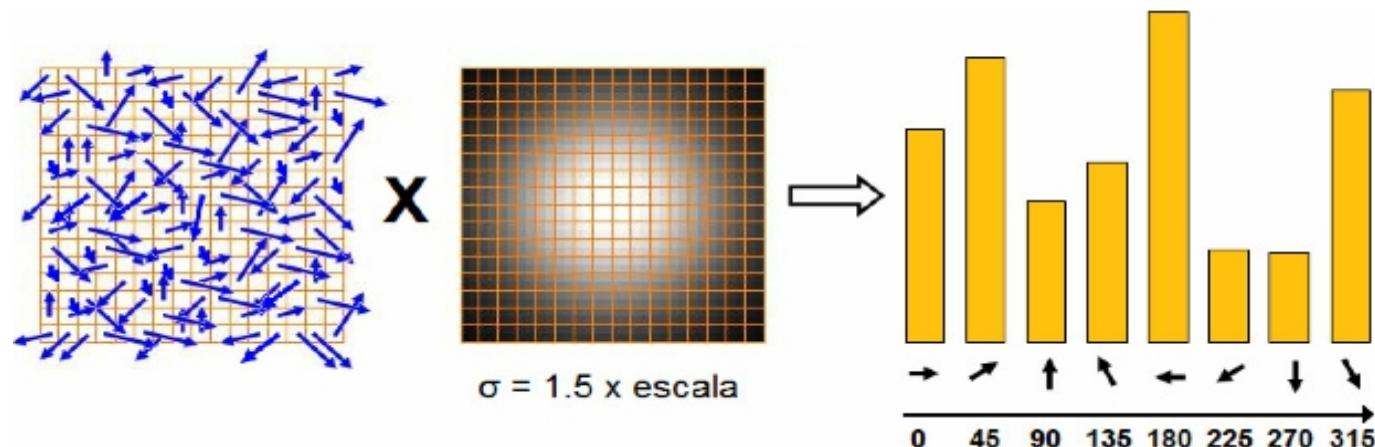
$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}$$



Sift: Localización Puntos de Interés

- Se calcula un histograma formado por el gradiente de las orientaciones de los píxeles cercanos al punto de interés estudiado.
- Las orientaciones se ponderan por la magnitud del gradiente y por una gaussiana circular de parámetro 1,5 veces la escala del punto de interés.
 - Dar mayor peso a las orientaciones con módulos elevados, que por tanto son más importantes.
 - Dar mayor importancia a los puntos cercanos al Keypoint es decir, los puntos centrales de la ventana.



Sift: Localización Puntos de Interés



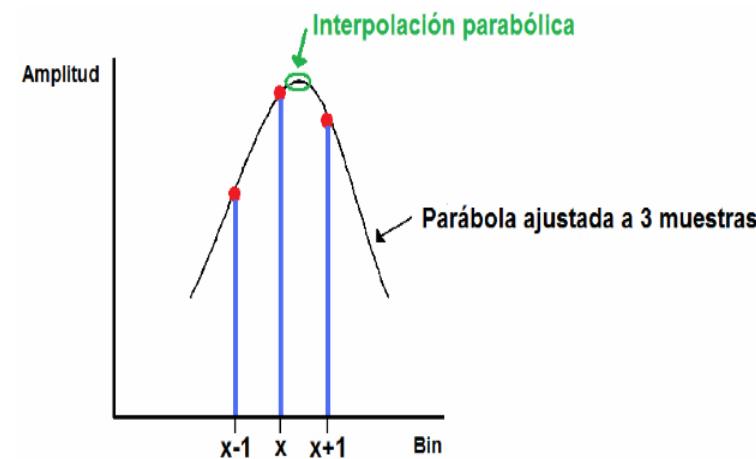
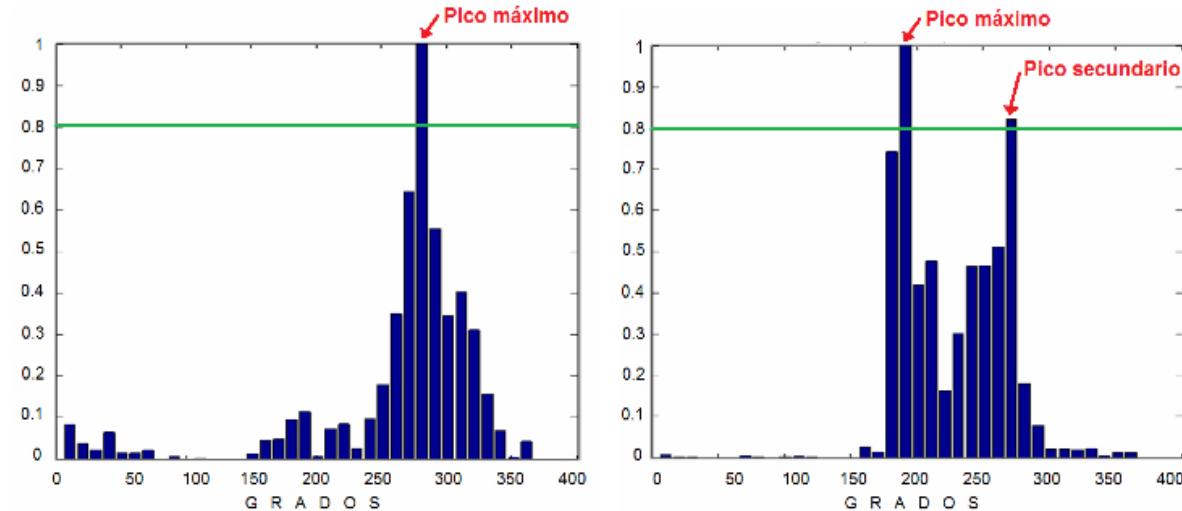
Los picos más altos de cada histograma son las direcciones dominantes de los gradientes locales, y por lo tanto la orientación final del keypoint. Sin embargo, en algunas ocasiones no nos quedaremos sólo con el pico más alto. El proceso será el siguiente:

- Detección del mayor pico.
- Búsqueda de picos secundarios que tengan una altura mayor al 80% del principal. Si no hay ninguno, nos quedamos sólo con el mayor.
- A cada pico seleccionado, se interpola su posición para una mayor precisión. Se lleva a cabo mediante la construcción de una parábola entre él mismo y sus vecinos laterales.

Sift: Localización Puntos de Interés



Para localizaciones con múltiples picos elevados de similar magnitud, se obtendrán varias orientaciones para un mismo punto de la imagen. Es decir, al construir los descriptores en la siguiente etapa del algoritmo, los keypoints con orientaciones múltiples tendrán asignados varios descriptores, que sólo diferirán en su inclinación





Sift: Localización Puntos de Interés

Descriptor de puntos de interés

En esta fase del algoritmo ya tenemos localización, escala y orientación para cada punto.

- **El objetivo** de este último paso es conseguir un descriptor de regiones que sea lo más invariante posible al resto de distorsiones (iluminación y perspectiva).
- La aproximación tomada por SIFT se basa en un origen biológico (en el funcionamiento del córtex visual primario).

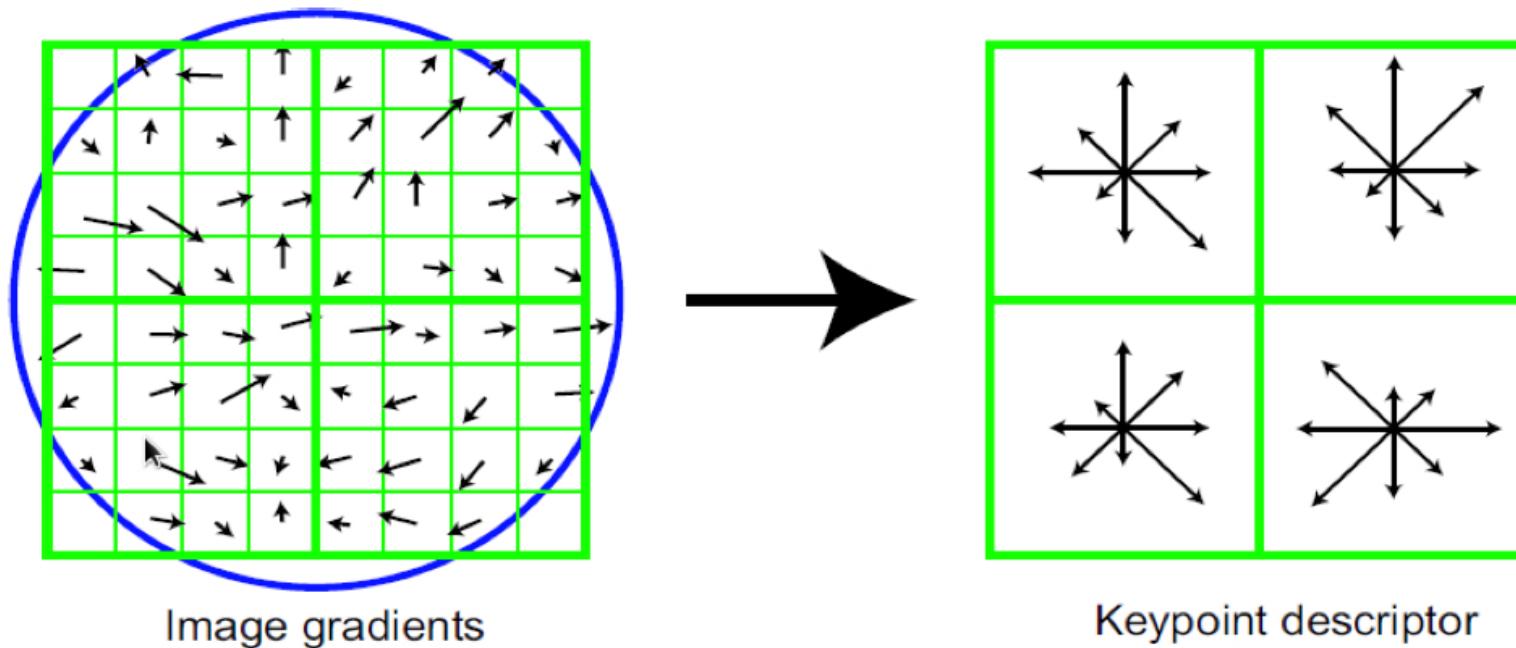
Para constuir el descriptor SIFT debemos crear **histogramas de orientación** para cada vecindario 4×4 con 8 direcciones posibles.



Sift: Localización Puntos de Interés

Estos histogramas se calculan a partir de los valores de orientación y magnitud del gradiente ponderados por una gaussiana de las muestras en una región 16×16 alrededor del punto de interés de tal modo que cada histograma contenga muestras de una subregión 4×4 del vecindario original.

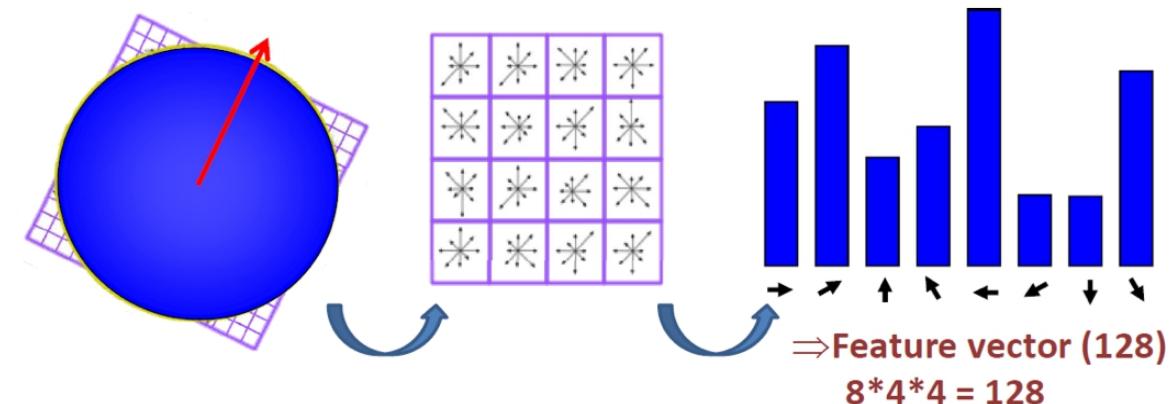
- Ejemplo (regiones 8×8 e histogramas 2×2)



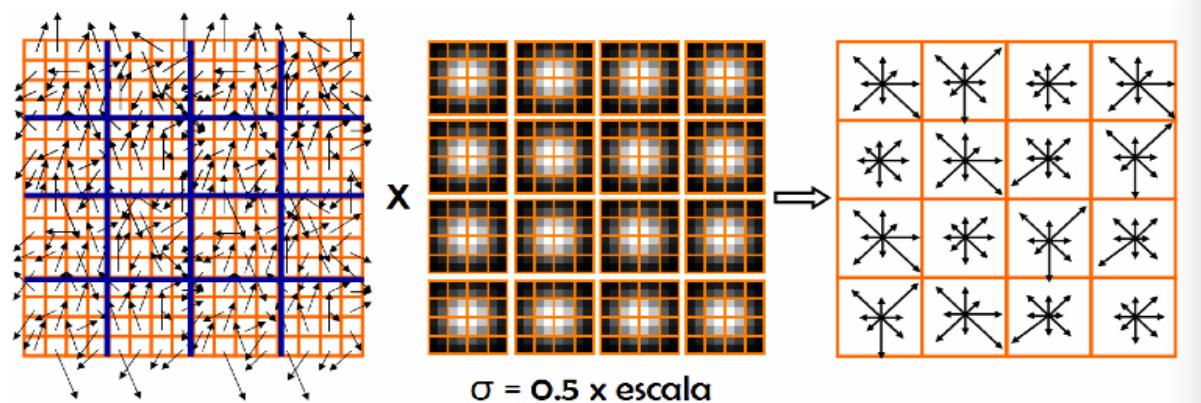


Sift: Localización Puntos de Interés

El descriptor de las características SIFT estará formado por 128 componentes: 8 direcciones por cada uno de los 4×4 histogramas.

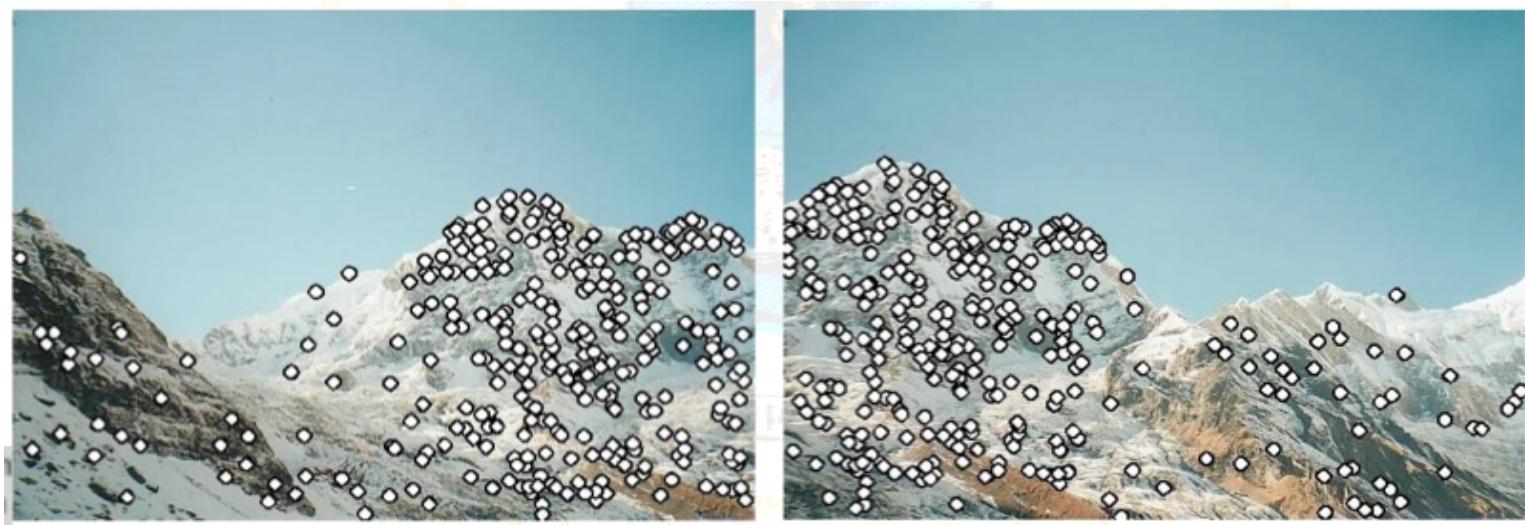


- Podremos utilizar una base de datos de descriptores de puntos de interés para reconocimiento de objetos.





Sift: Alineamiento de Imágenes





Sift: Alineamiento de Imágenes

