

Visión Computacional

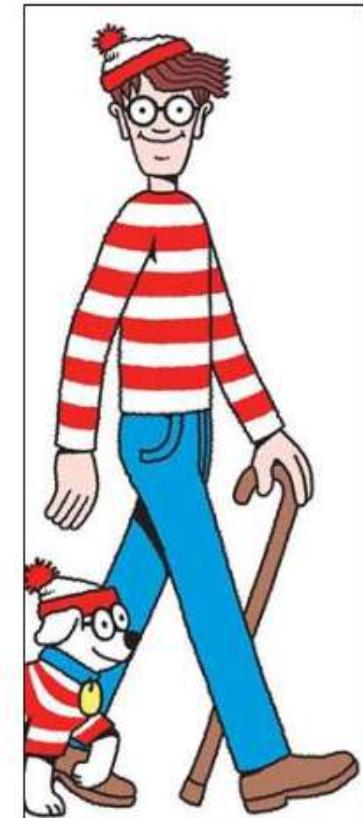
Ivan Sipiran

Edge detection

Queremos encontrar a Waldo



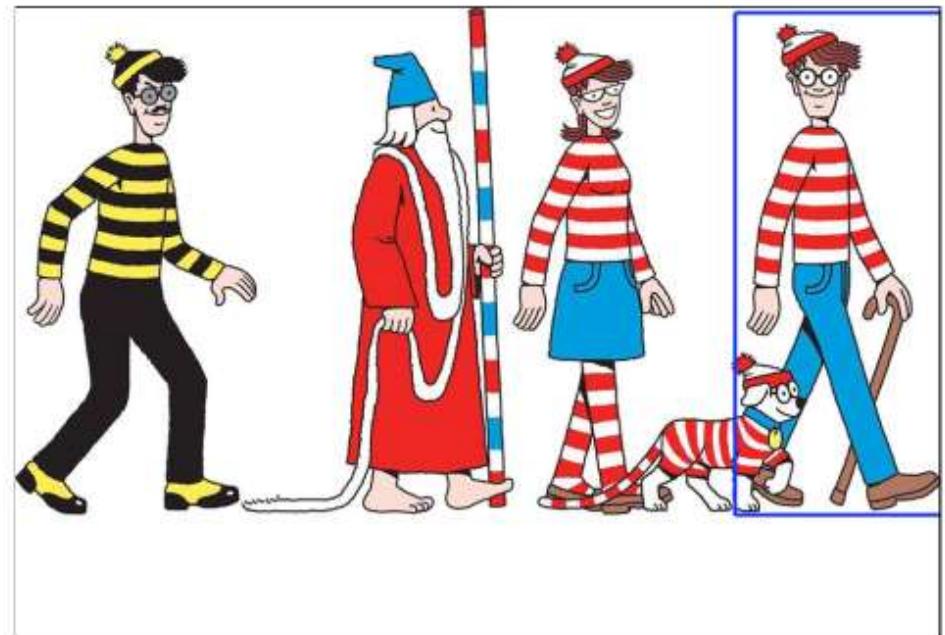
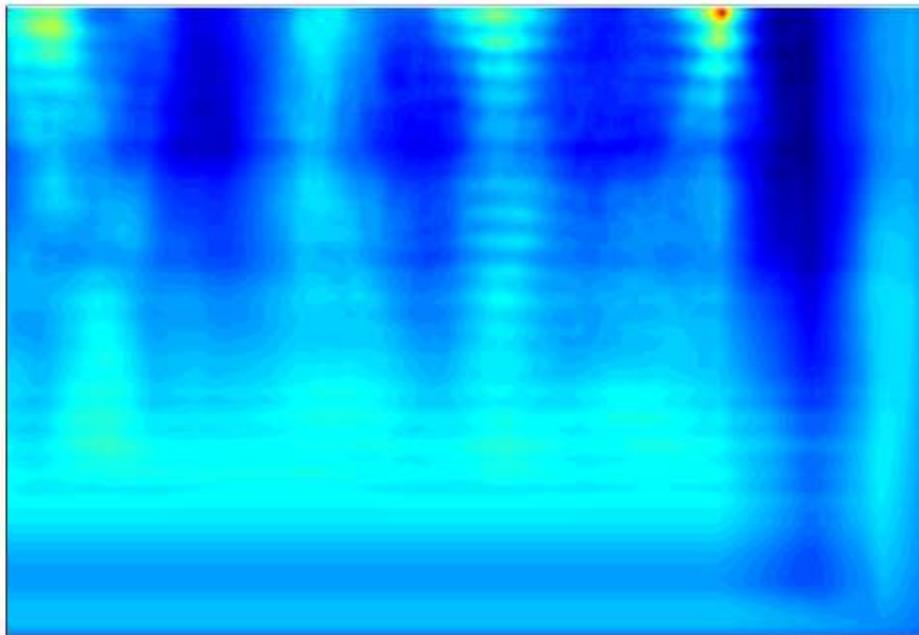
Imagen



filtro

Edge detection

Aplicamos correlación cruzada normalizada



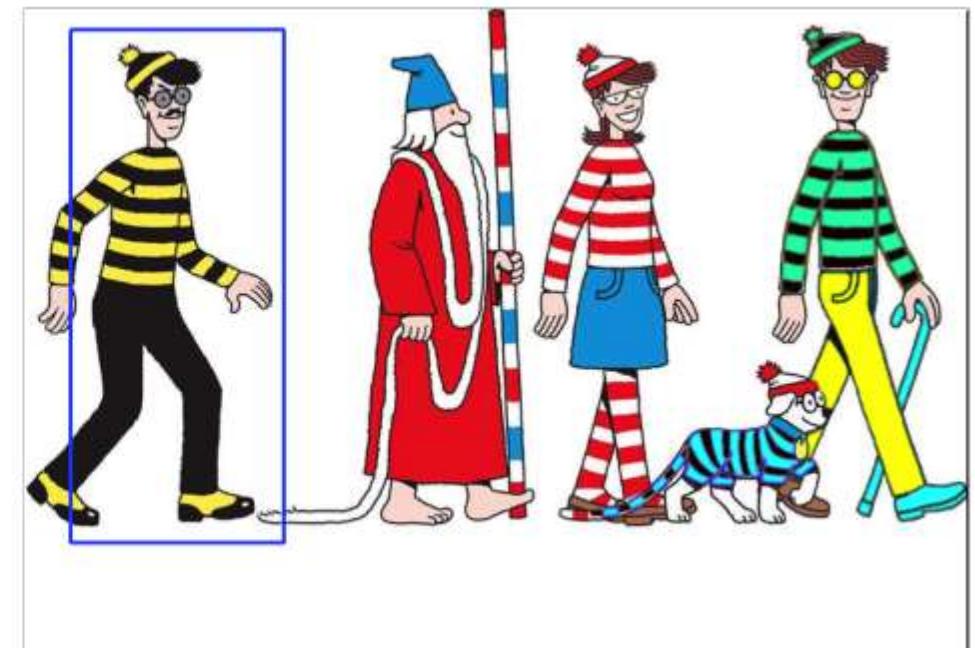
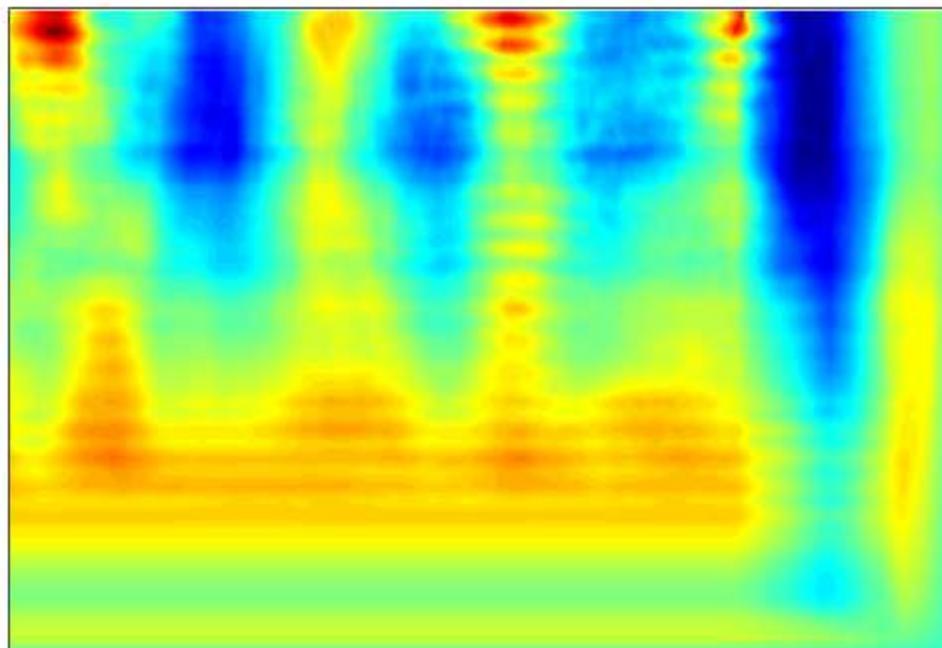
Edge detection

Supongamos que Waldo va de compras.



Edge detection

Supongamos que Waldo va de compras.



Problema: correlación depende de los colores!

Qué tipo de información se preserva?

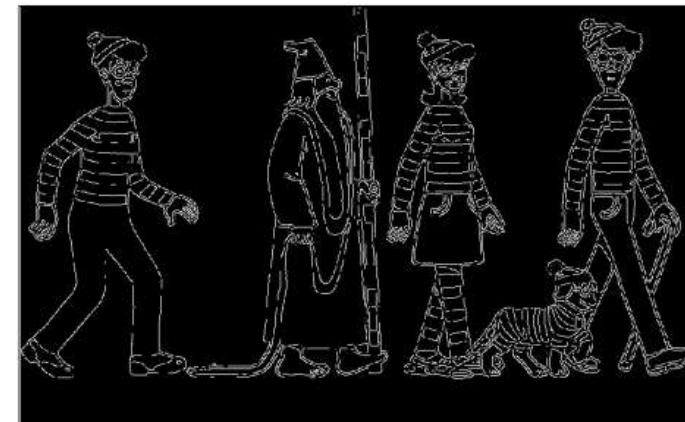
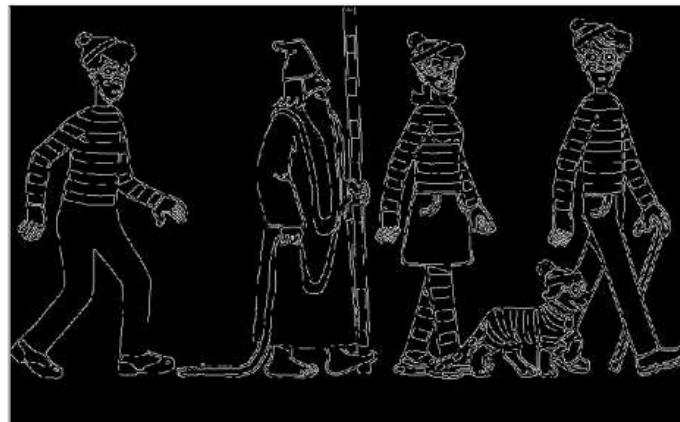
Edge detection

Bordes!



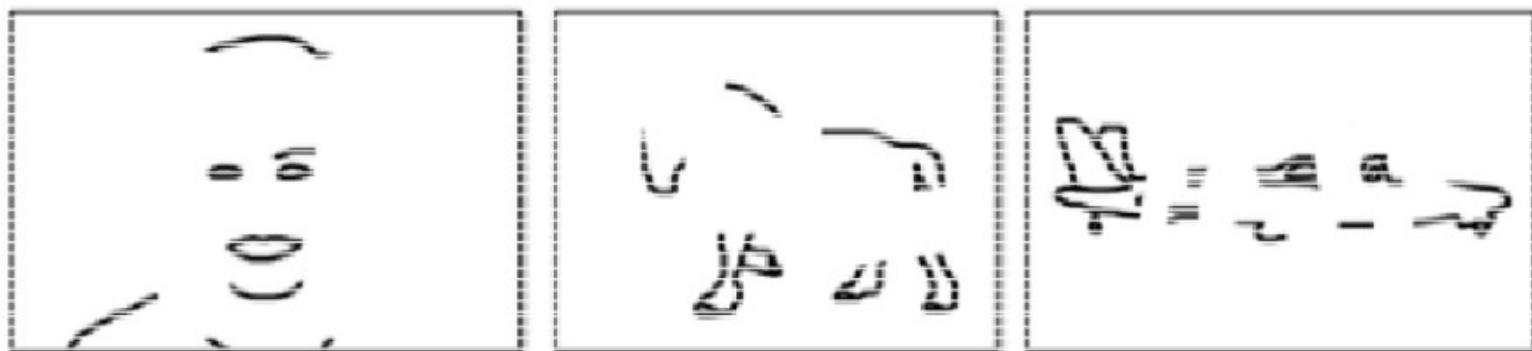
Edge detection

Bordes!



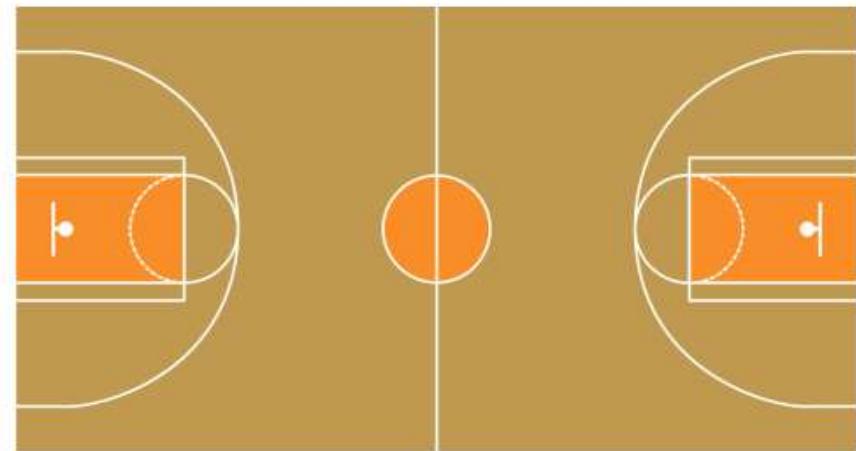
Edge detection

- Mapear imágenes a un conjunto de curvas o segmentos de línea o contornos
- Más compacto que un píxel
- Invariantes a cambios de iluminación
- Importantes para reconocimiento



Edge detection

- Mapear imágenes a un conjunto de curvas o segmentos de línea o contornos
- Más compacto que un píxel
- Invariantes a cambios de iluminación
- Importantes para reconocimiento
- Importante para varias aplicaciones



Qué tan lejos está la persona a la canasta?

Edge detection

- Mapear imágenes a un conjunto de curvas o segmentos de línea o contornos
- Más compacto que un píxel
- Invariantes a cambios de iluminación
- Importantes para reconocimiento
- Importante para varias aplicaciones



Cómo puede un robot tomar la moneda?

Edge detection

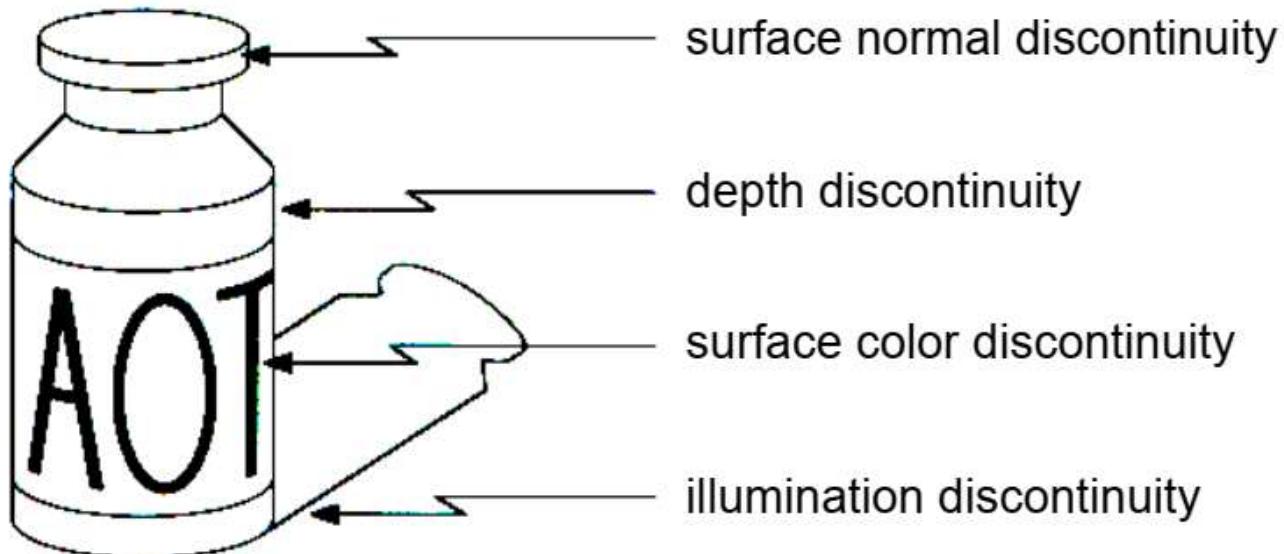
- Mapear imágenes a un conjunto de curvas o segmentos de línea o contornos
- Más compacto que un píxel
- Invariantes a cambios de iluminación
- Importantes para reconocimiento
- Importante para varias aplicaciones



Cómo puede un robot tomar la moneda?

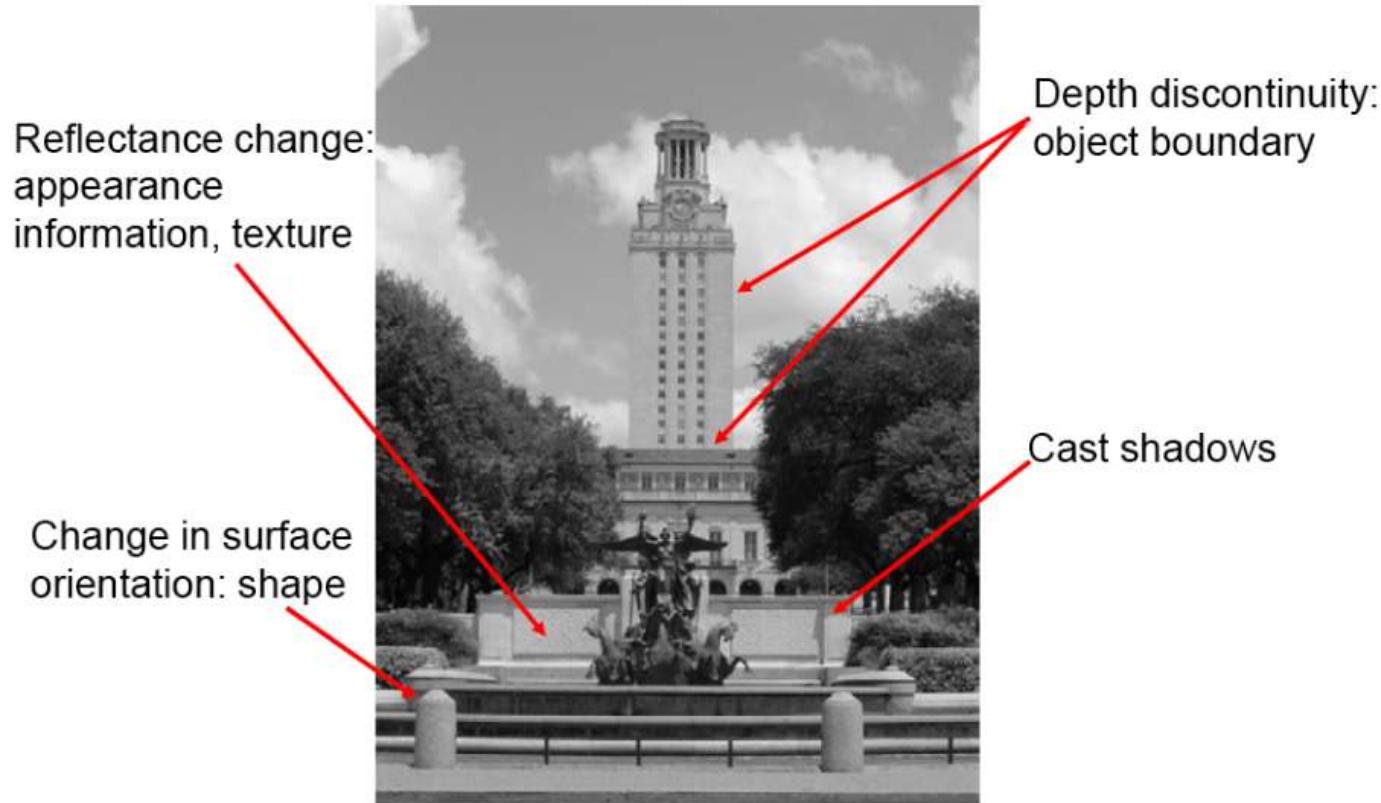
El origen de los bordes

- Bordes son causados por una variedad de factores



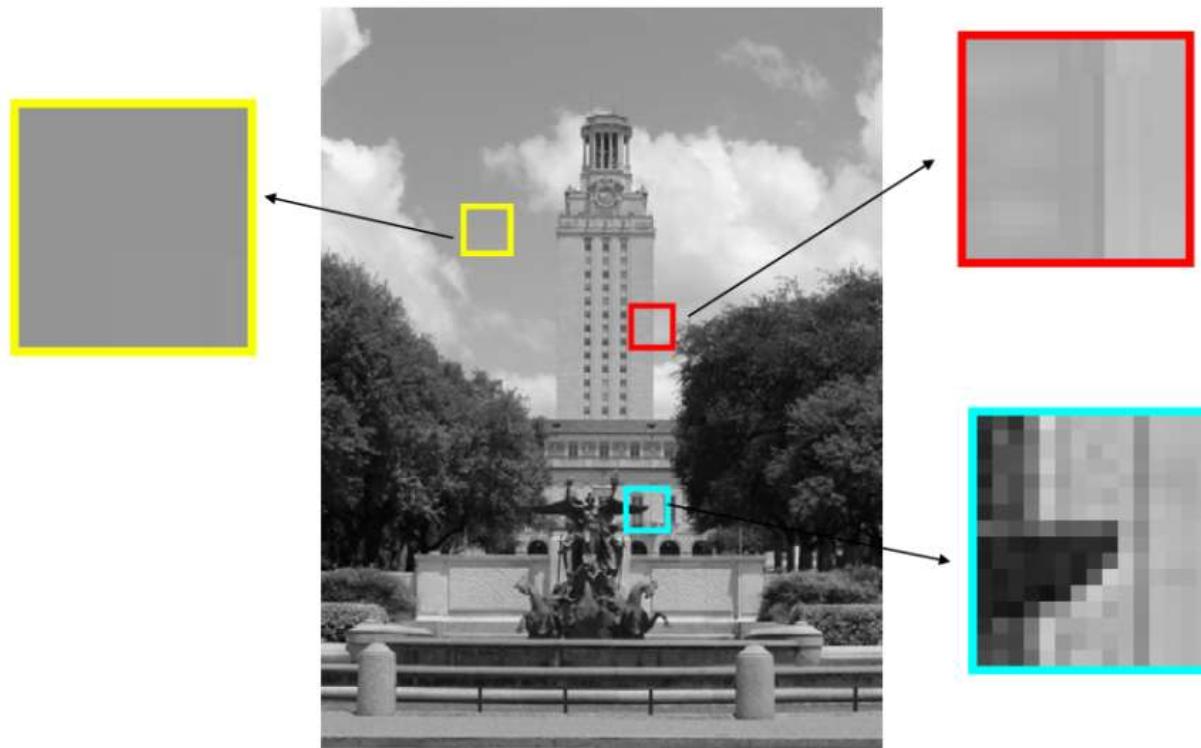
El origen de los bordes

- Bordes son causados por una variedad de factores



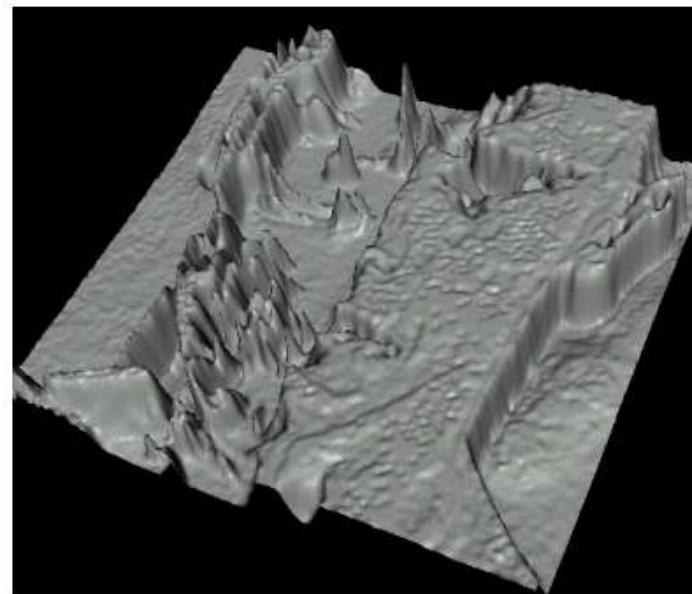
El origen de los bordes

- Bordes son causados por una variedad de factores



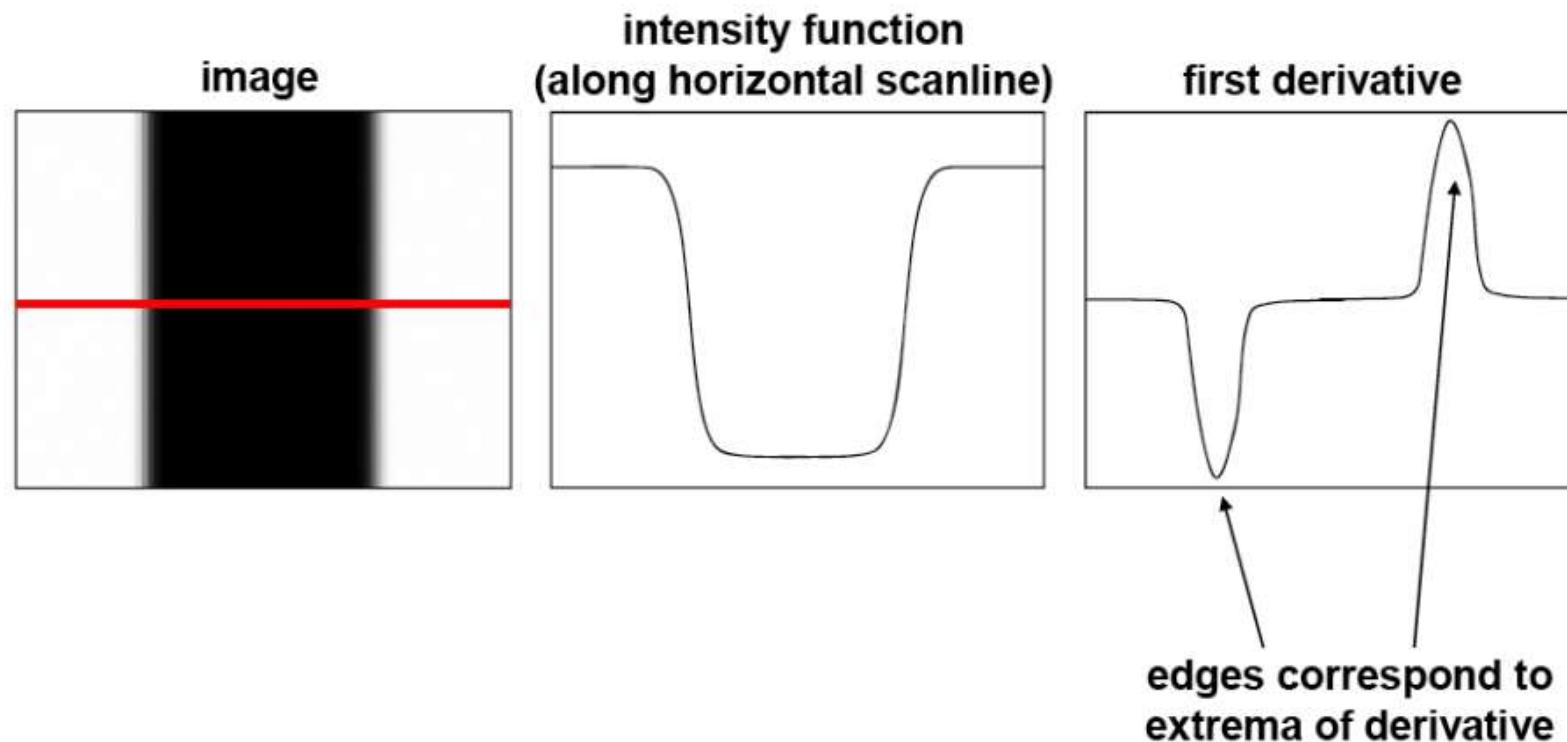
El origen de los bordes

- Si miramos la imagen como una función
- Bordes son como acantilados



Caracterización de bordes

- Lugar de rápido cambio en la intensidad de la imagen



Caracterización de bordes

- Cómo calculamos derivadas en una imagen $f(x, y)$?
 - Si la imagen fuera continua

$$\frac{\partial f(x, y)}{\partial x} = \lim_{h \rightarrow 0} \frac{f(x + h, y) - f(x, y)}{h}$$

- En discreto

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

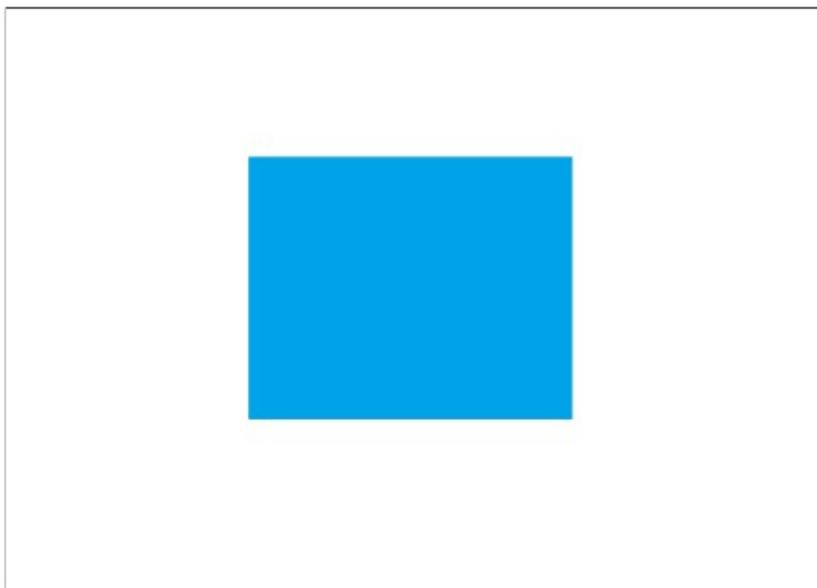
- Se puede aplicar como un filtro!

$$\frac{\partial f}{\partial x}:$$

$$\frac{\partial f}{\partial y}:$$

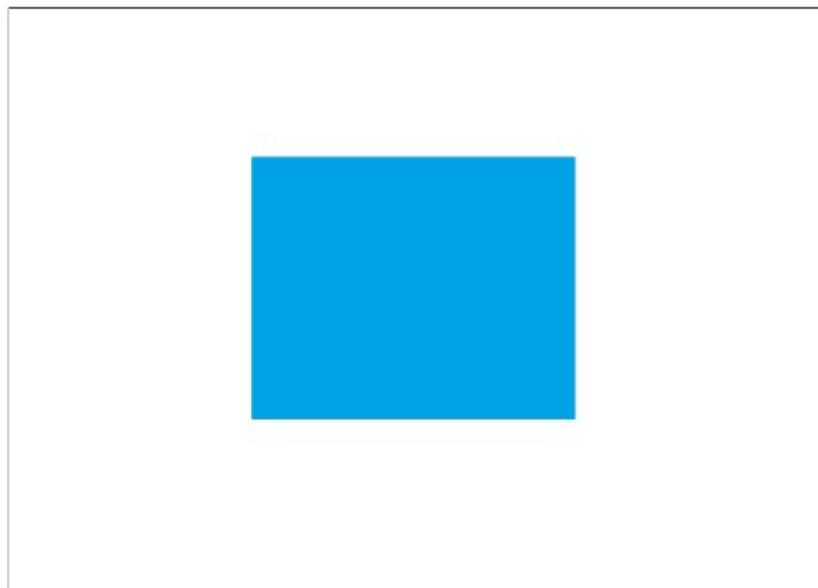
Caracterización de bordes

- Cómo sería la derivada horizontal si el filtro es $[-1, 1]$?



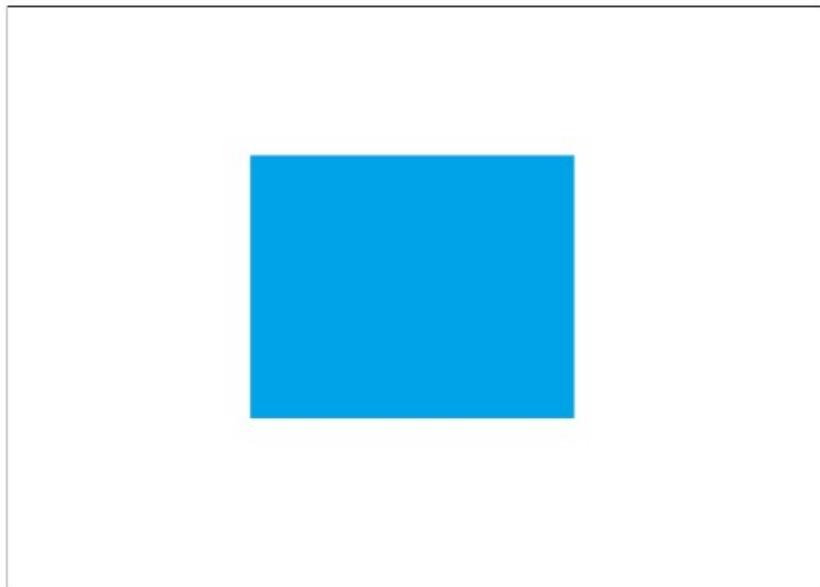
Caracterización de bordes

- Cómo sería la derivada horizontal si el filtro es $[-1, 1]$?



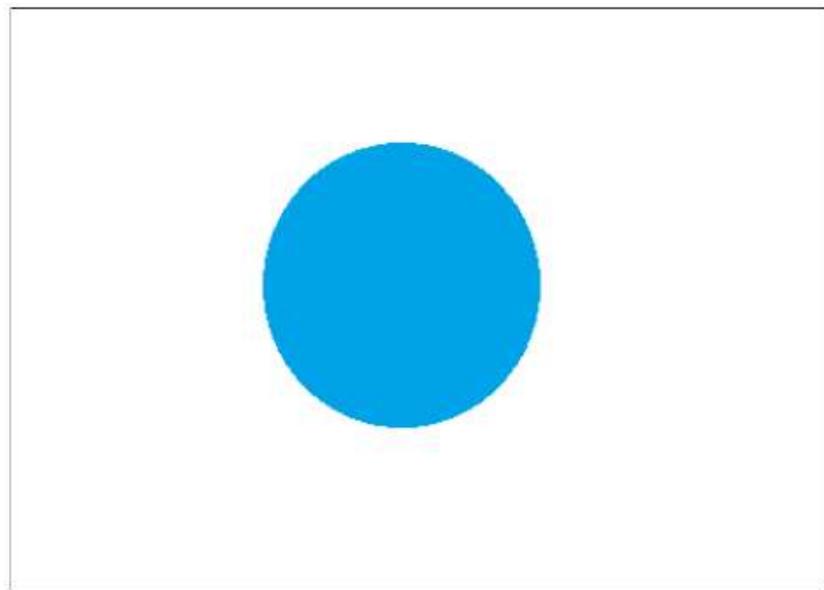
Caracterización de bordes

- Cómo sería la derivada vertical si el filtro es $[-1,1]^T$?



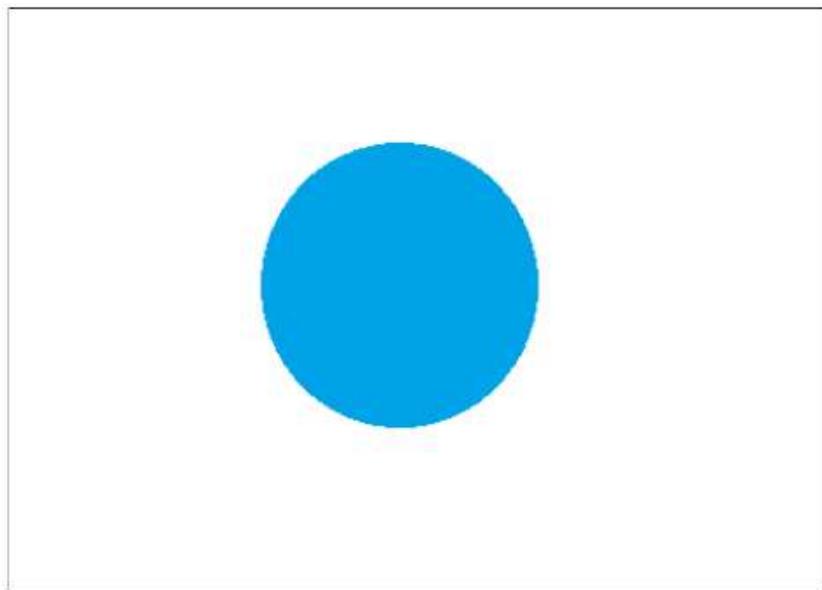
Caracterización de bordes

- Cómo sería la derivada horizontal si el filtro es $[-1, 1]$?



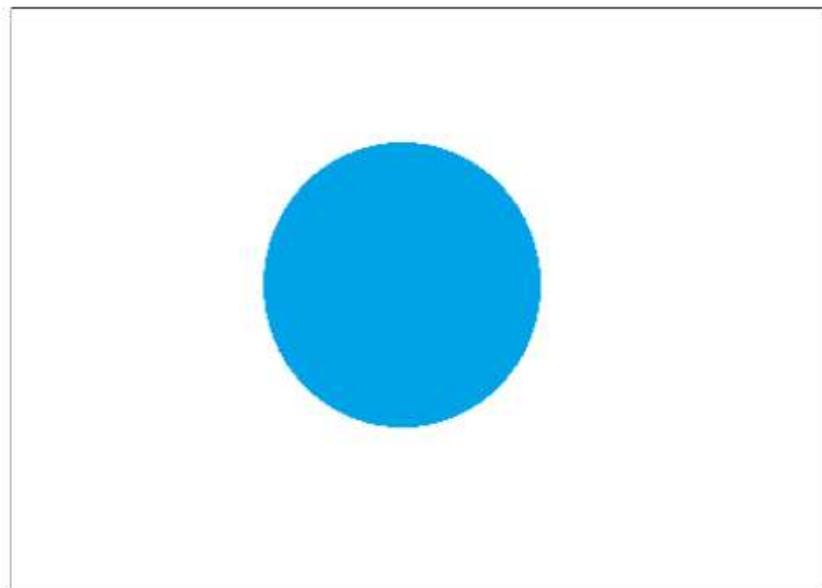
Caracterización de bordes

- Cómo sería la derivada horizontal si el filtro es $[-1, 1]$?



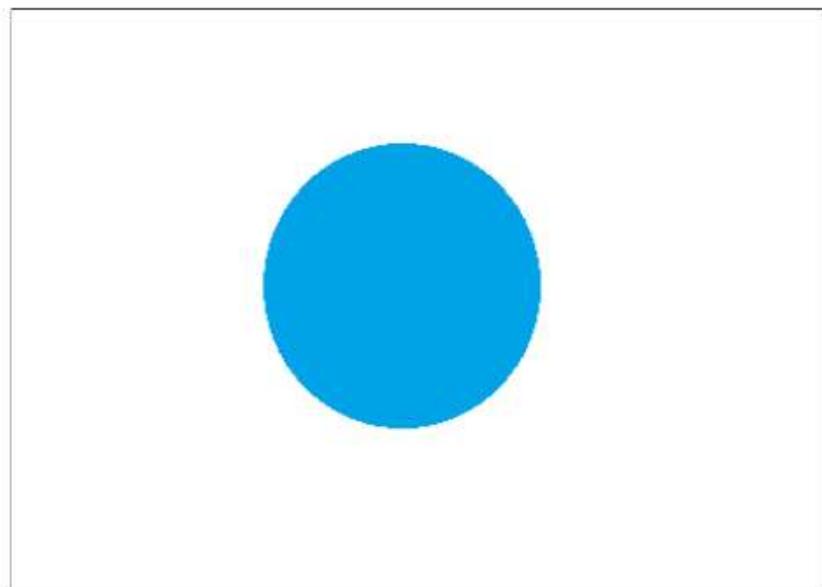
Caracterización de bordes

- Cómo sería la derivada vertical si el filtro es $[-1,1]^T$?



Caracterización de bordes

- Cómo sería la derivada vertical si el filtro es $[-1,1]^T$?

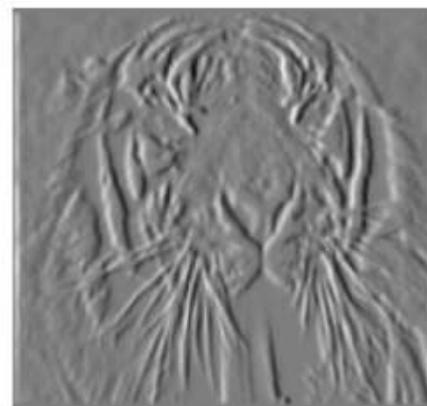


Caracterización de bordes



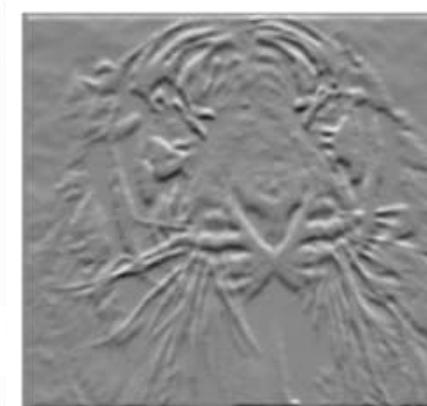
$$\frac{\partial f(x, y)}{\partial x}$$

-1	1
----	---



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

-1	?	1
1	or	-1



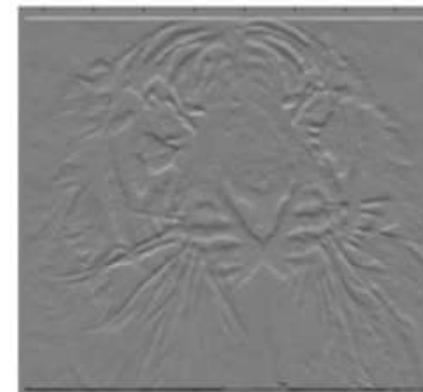
Filtros de diferencia finita

Prewitt: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

Sobel: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

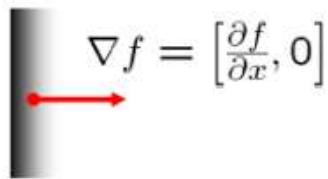
Roberts: $M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

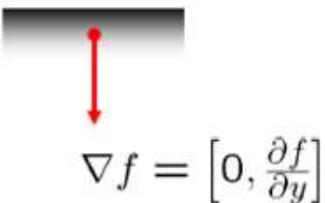
```
>> My = fspecial('sobel');  
>> outim = imfilter(double(im), My);  
>> imagesc(outim);  
>> colormap gray;
```

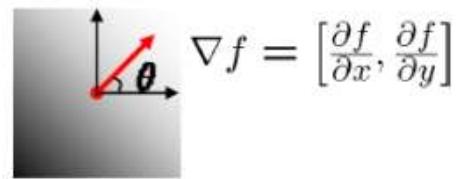


Gradiente de una imagen

- El gradiente de una imagen $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$
- Gradiente apunta en la dirección de cambio más rápido en intensidad


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


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

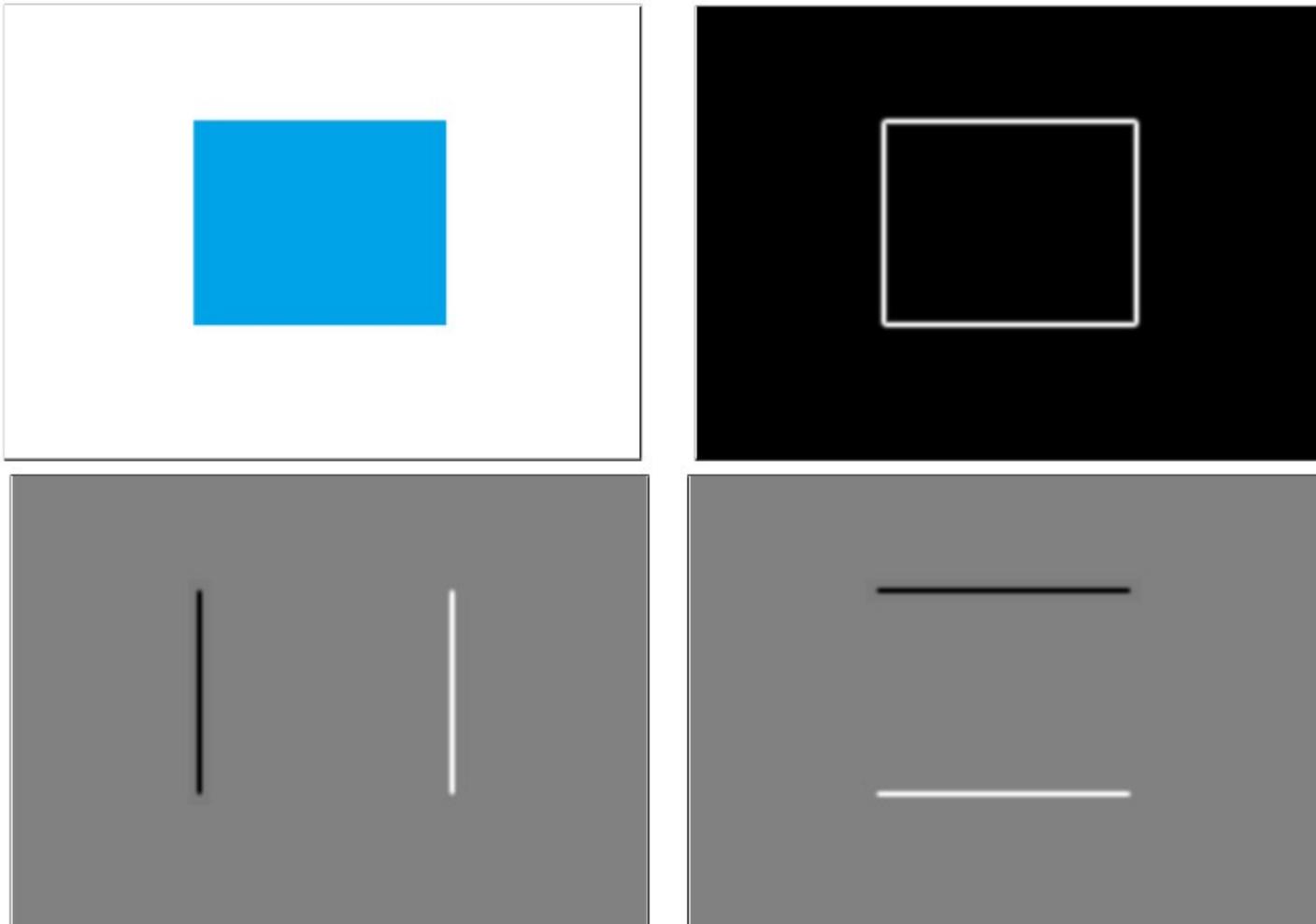

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

- Dirección de gradiente Magnitud

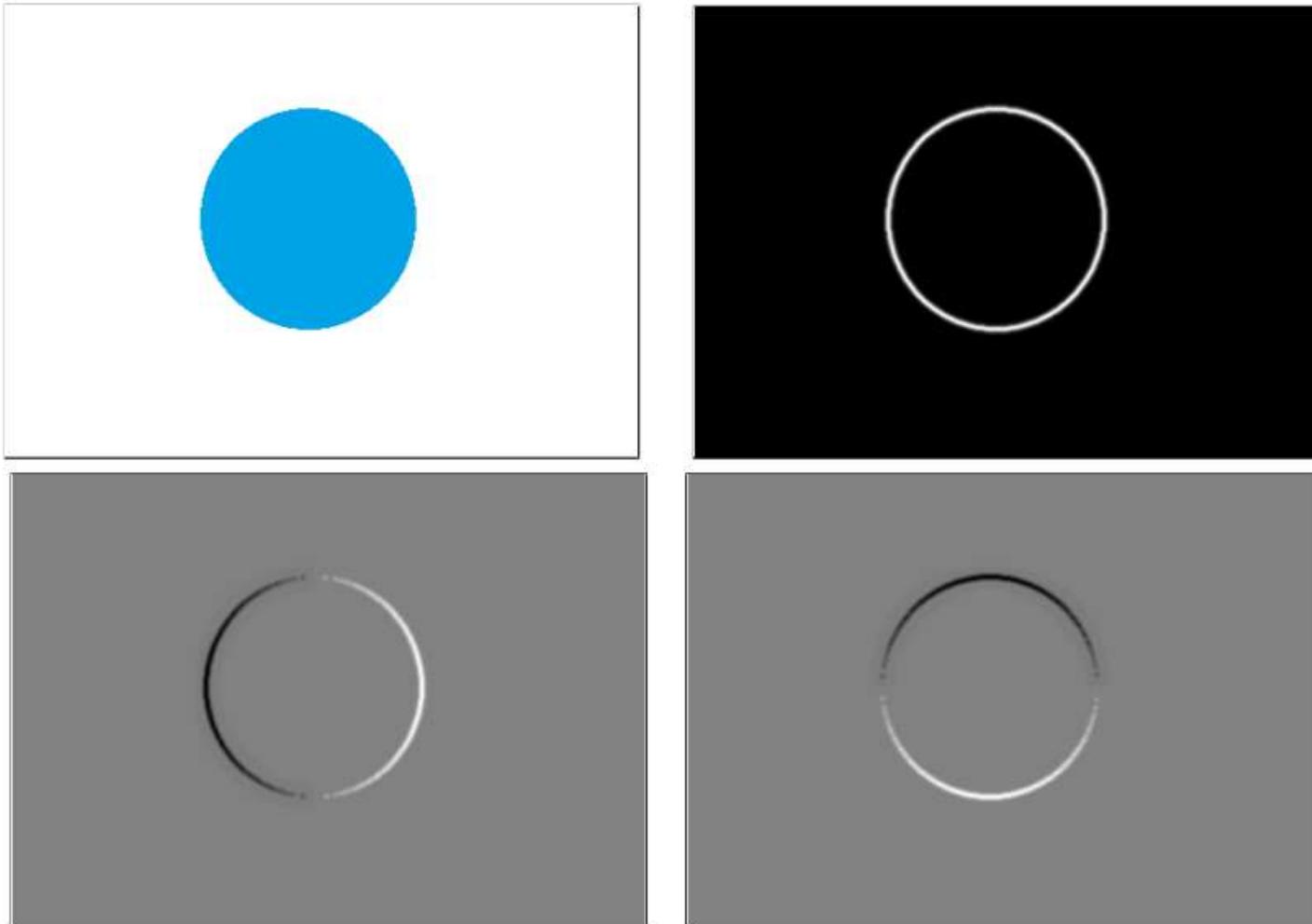
$$\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$

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

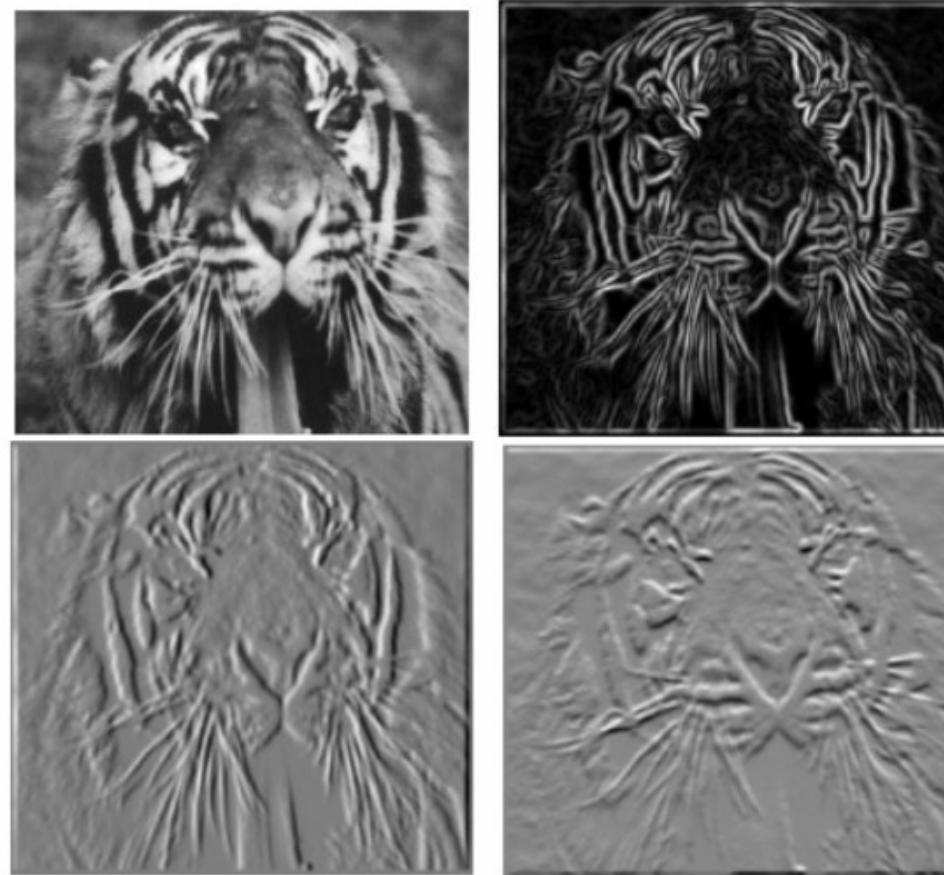
Gradiente de una imagen



Gradiente de una imagen

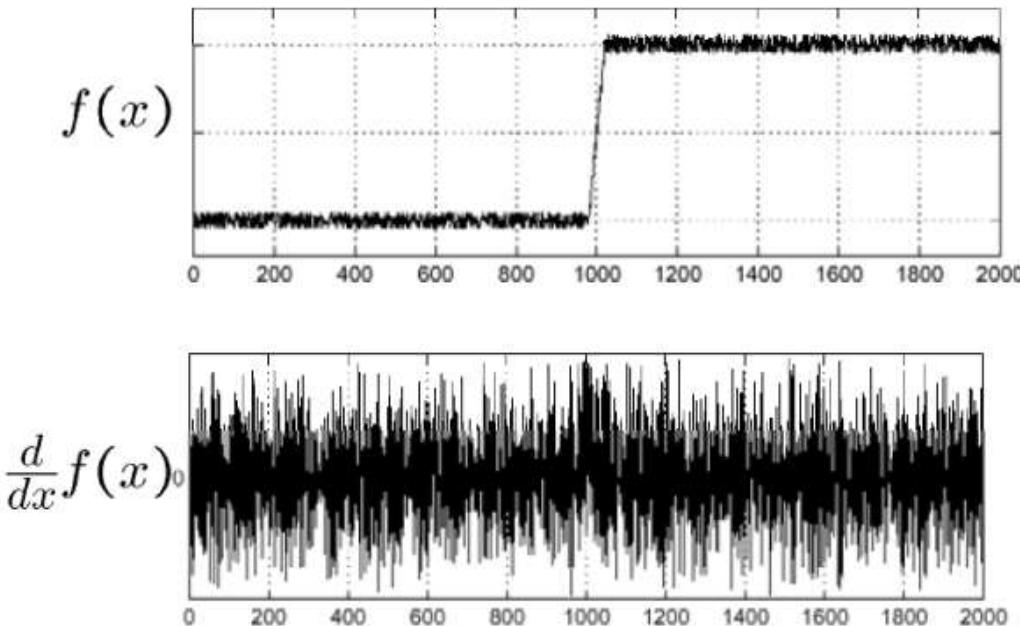
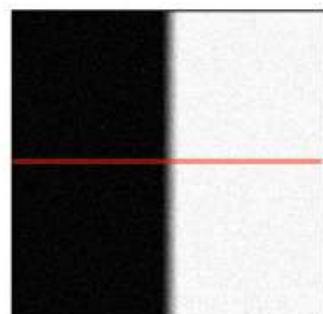


Gradiente de una imagen



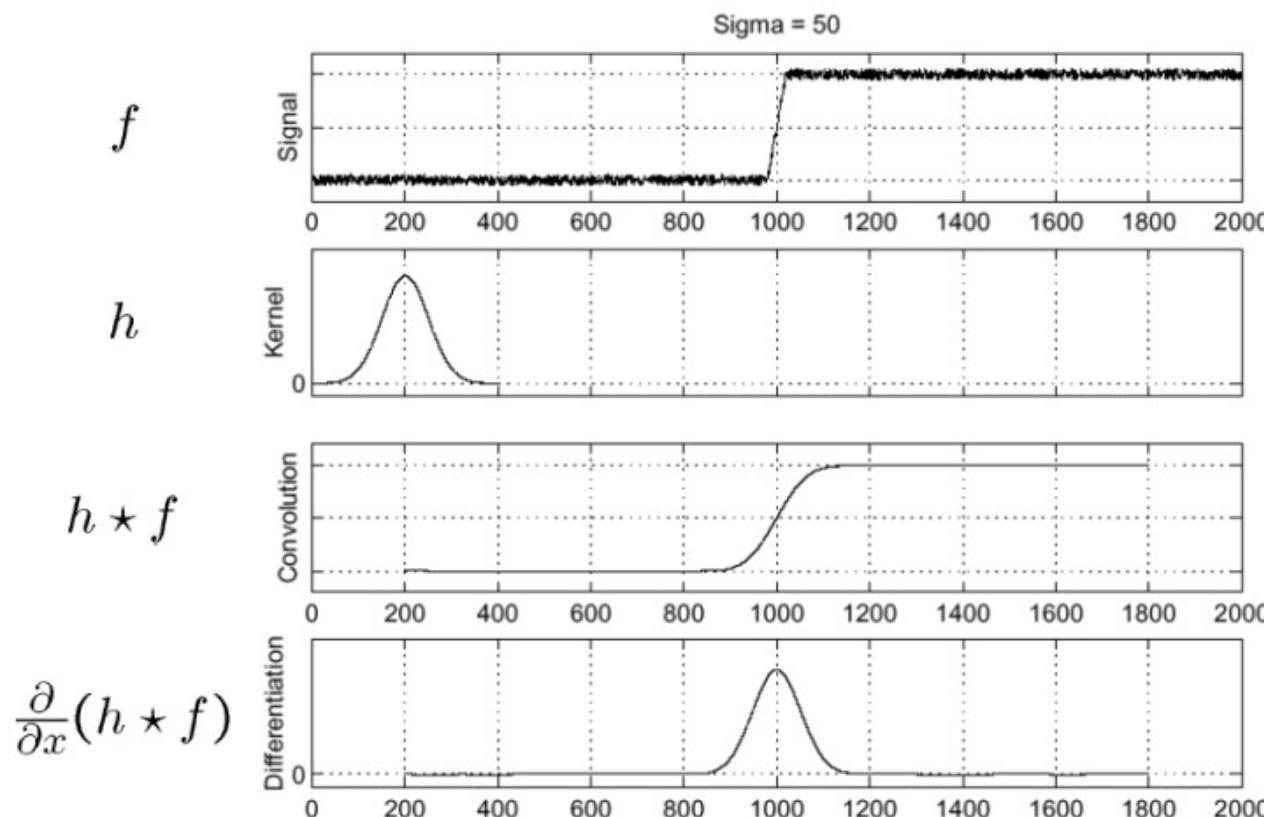
Efectos del ruido

- Qué pasa si la imagen es ruidosa?



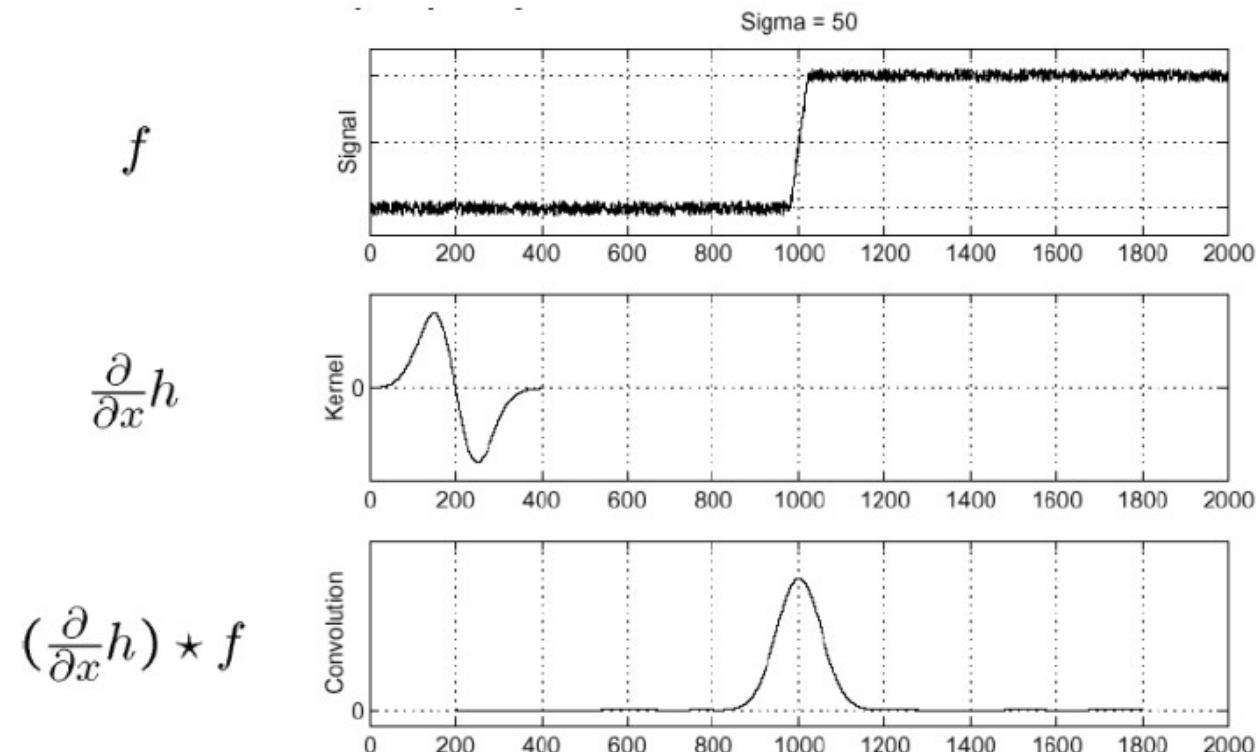
Efectos del ruido

- Suavizar con algún filtro h (ej. Gaussiano) y evaluar el gradiente

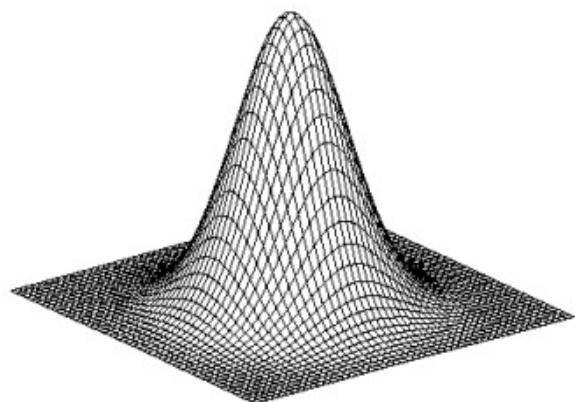


Efectos del ruido

- Propiedad derivada de convolución $\frac{\partial}{\partial x}(h * f) = \left(\frac{\partial h}{\partial x}\right) * f = h * \left(\frac{\partial f}{\partial x}\right)$
- Nos ahorraremos una operación

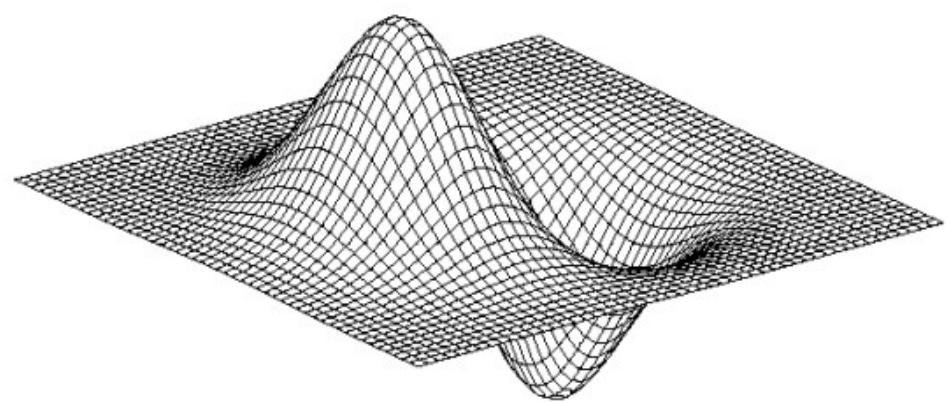


Filtros de detección 2D



Gaussian

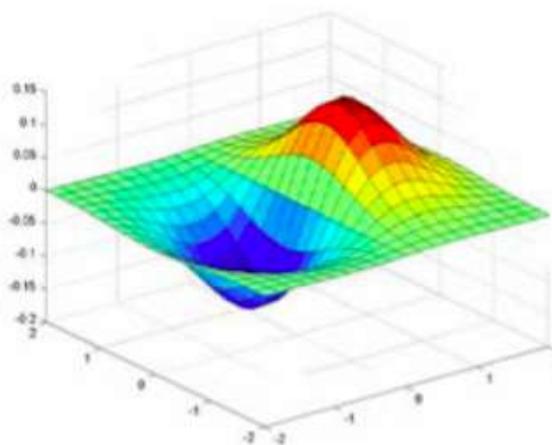
$$h_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{u^2+v^2}{2\sigma^2}}$$



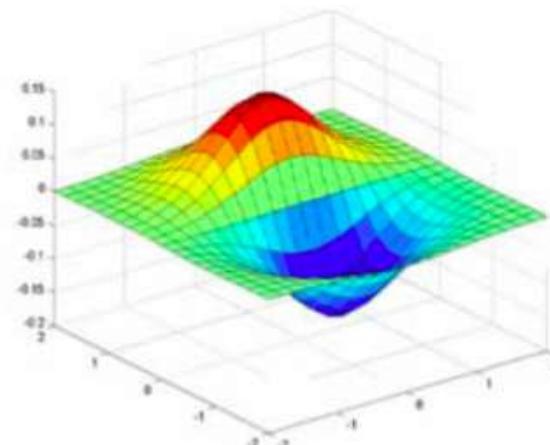
Derivative of Gaussian (x)

$$\frac{\partial}{\partial x} h_\sigma(u, v)$$

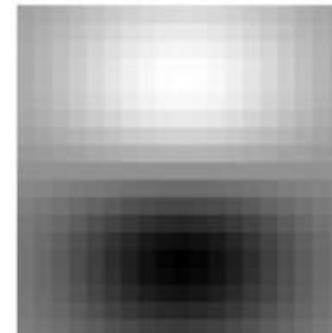
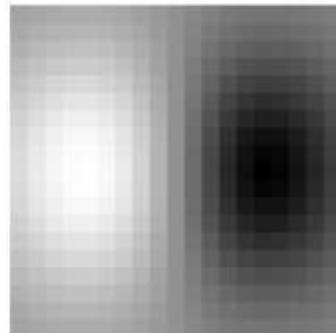
Derivada del Gaussiano



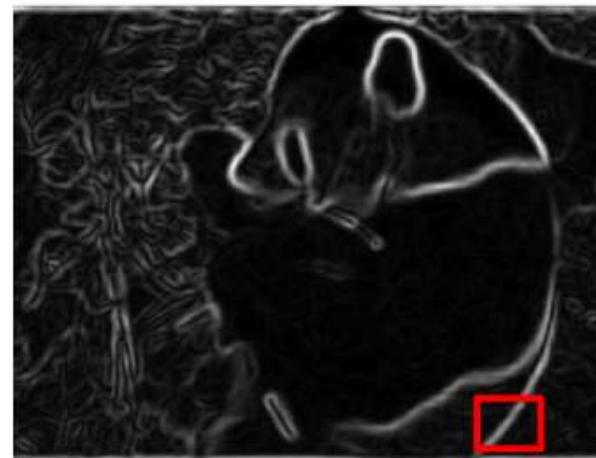
x-direction



y-direction

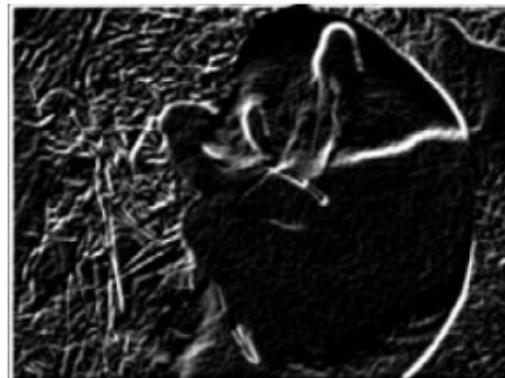


Derivada del Gaussiano

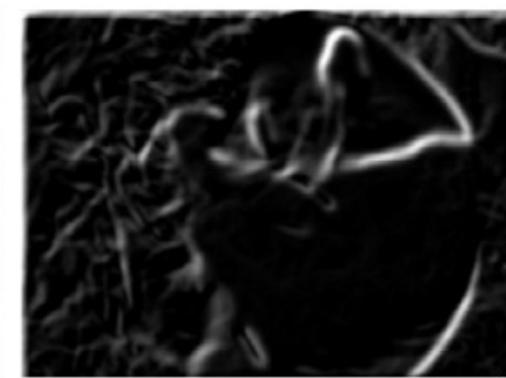


Derivada del Gaussiano

- Efecto de σ en el parámetro del Gaussiano
 - Valores grandes: detecta bordes de escala grande
 - Valores pequeños: detecta bordes finos



$\sigma = 1$ pixel



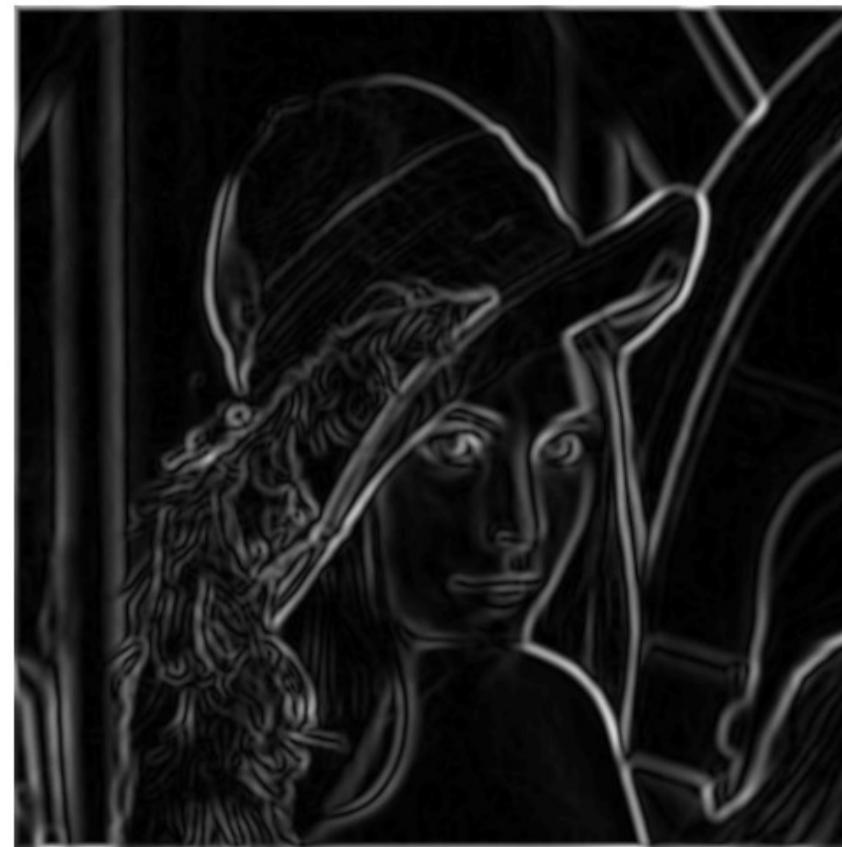
$\sigma = 3$ pixels

Detector de bordes Canny



Detector de bordes Canny

Tomamos la magnitud del gradiente



Detector de bordes Canny

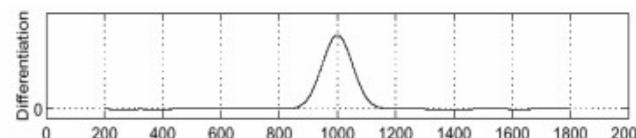
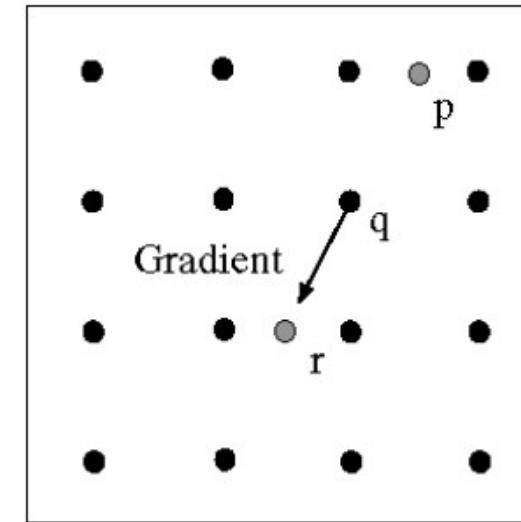
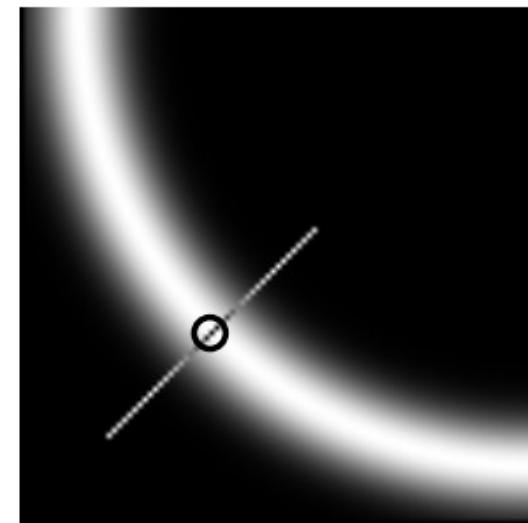
Aplicamos binarización



Detector de bordes Canny



Supresión de no máximos



Chequear si un pixel es máximo local en la dirección del gradiente

Encontrando bordes

Hay píxeles que no sobreviven a la binarización



Hysteresis Thresholding

Usar un umbral alto para empezar las curvas de bordes y usar un umbral bajo para continuarlas



Hysteresis Thresholding



original image



high threshold
(strong edges)



low threshold
(weak edges)



hysteresis threshold

Canny - Resultados



Image Features

Image Features

- Descripciones útiles de propiedades locales o globales diseñadas (o aprendidas!) para lograr cierta tarea
- Diferentes features para diferentes tareas
- Tres preguntas
 - Dónde extraer características en una imagen?
 - Qué extraer (contenido)?
 - Cómo usarlas para resolver tarea?

Image Features

- Qué lugar es este?



Image Features

- Qué lugar es este?

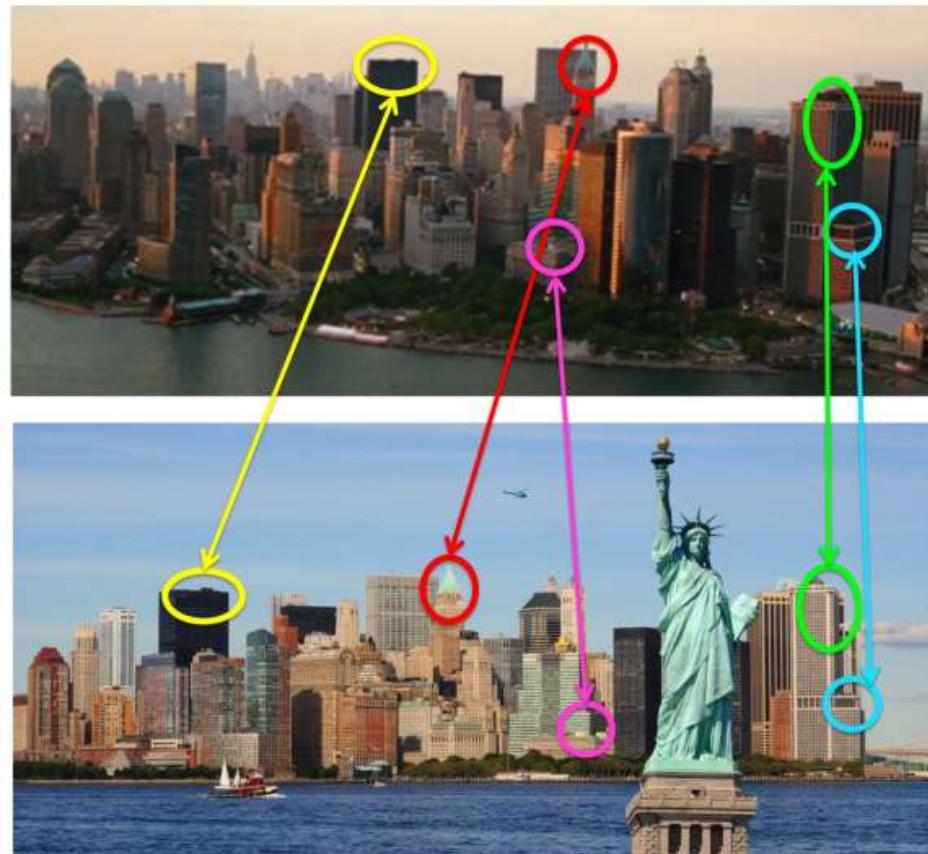
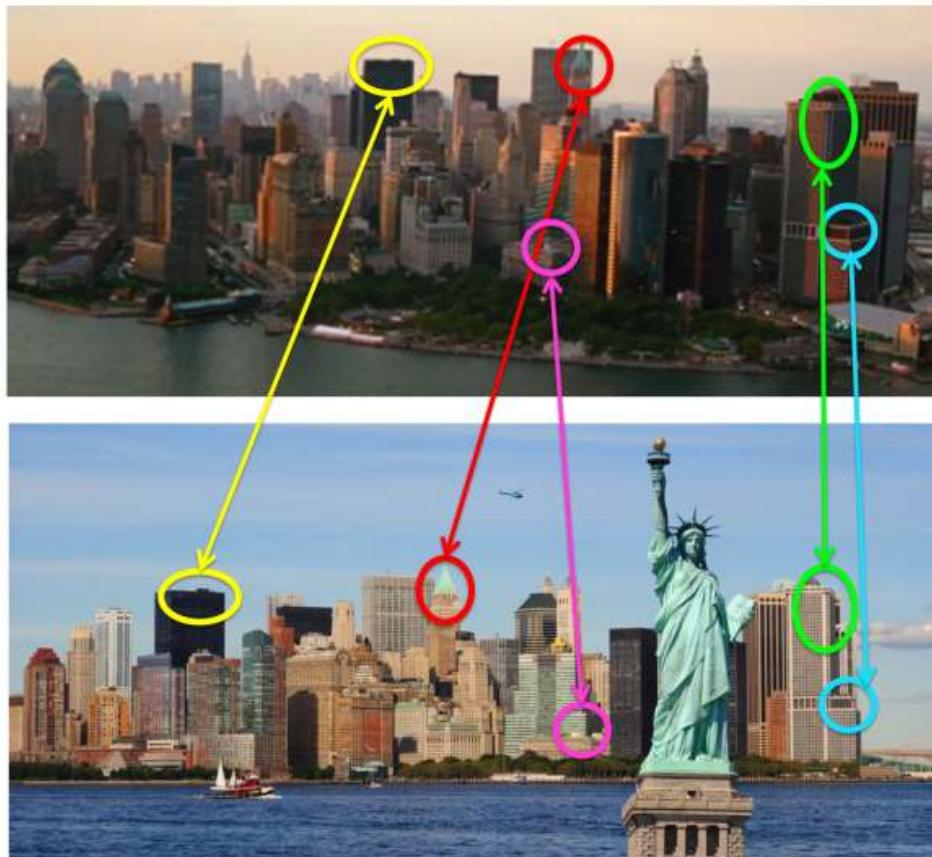


Image Features

- Qué lugar es este?



Necesitamos puntos distintivos

Necesitamos representaciones distintivas

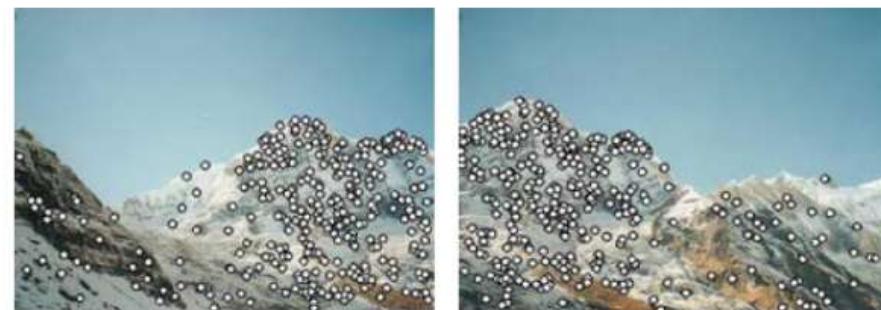
Image Features

- **Detección:** Dónde extraer características?
 - Keypoints, o regiones de interés
 - Cada píxel (denso)
- **Descripción:** Qué extraer?
 - Alcance espacial del feature
 - Contenido del feature
- **Matching:** cómo encontramos correspondencias

Detección de keypoints

Ejemplo

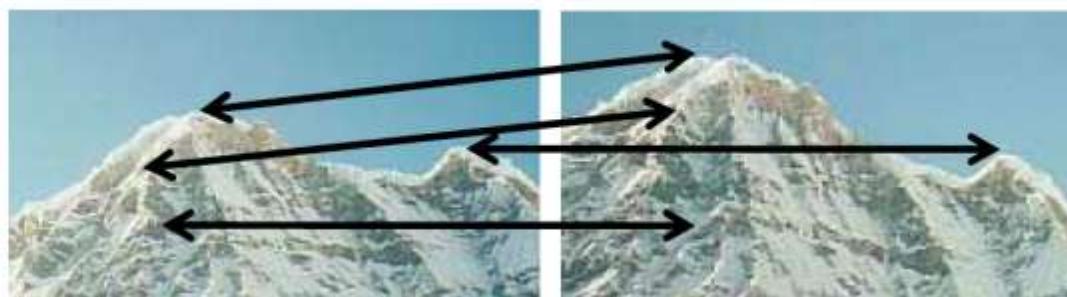
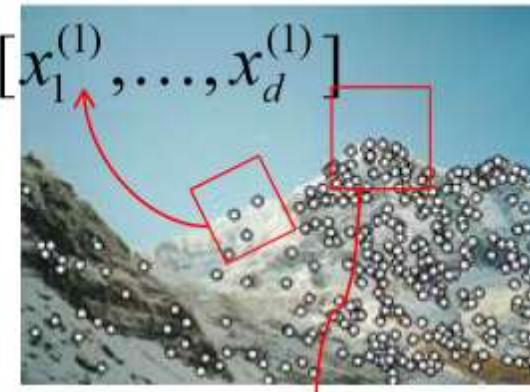
- Construcción de imágenes panorámicas



Ejemplo



$$\mathbf{x}_l = [x_1^{(l)}, \dots, x_d^{(l)}]$$



Keypoints: repetibilidad

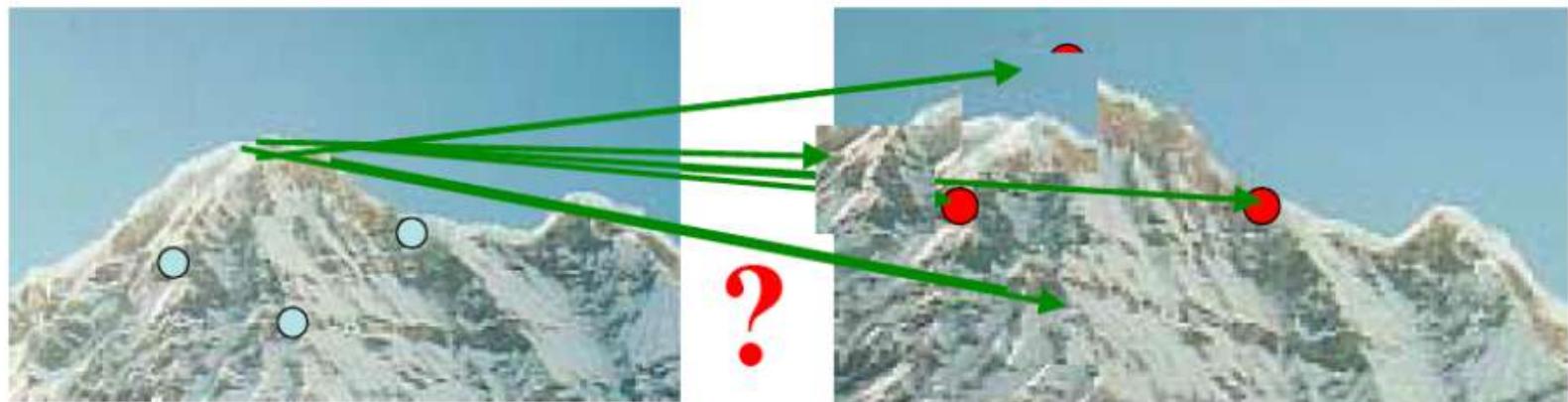
- Detectar los mismos puntos en varias imágenes
- Detectar puntos suficientes, pero no muchos



- Con pocos puntos, hay pocas chances de encontrar matches

Keypoints: distintividad

- Poder determinar qué punto es cual



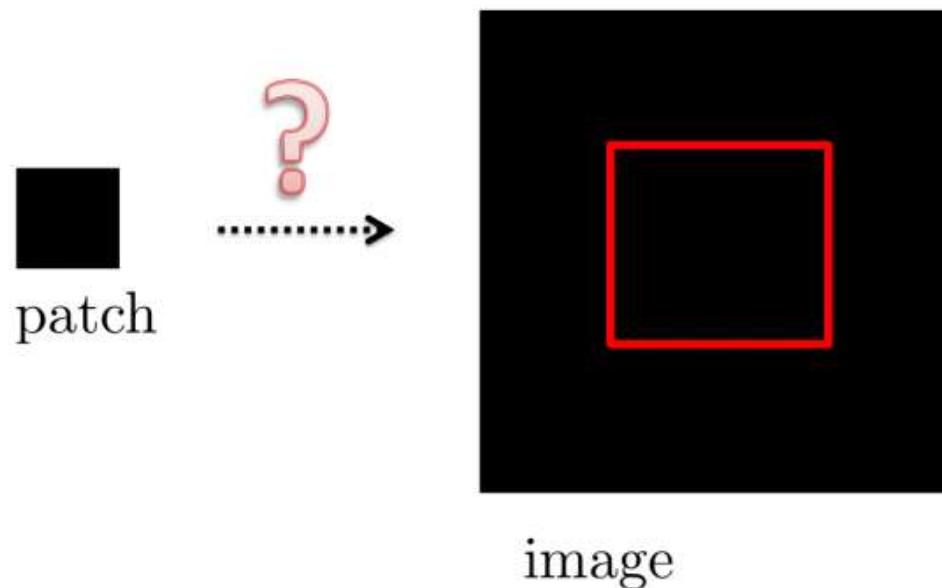
Keypoints

- Qué puntos escoger?



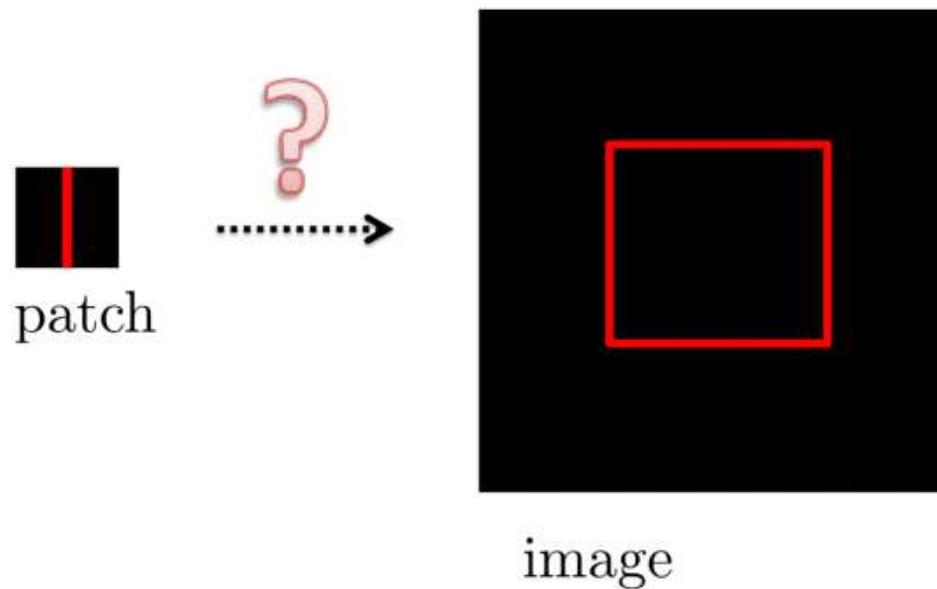
Keypoints

- Puntos sin textura son imposibles de localizar



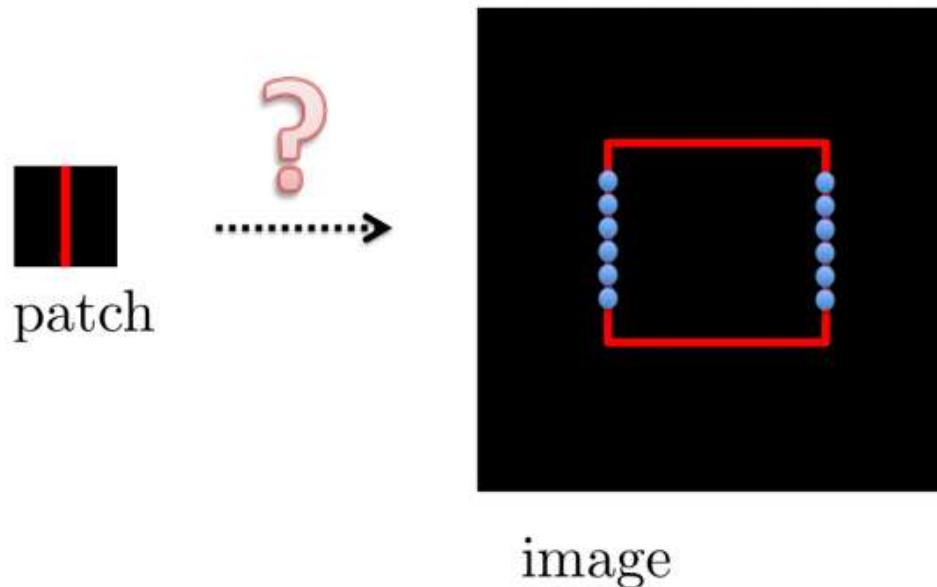
Keypoints

- Puntos con alto contraste son fáciles de localizar



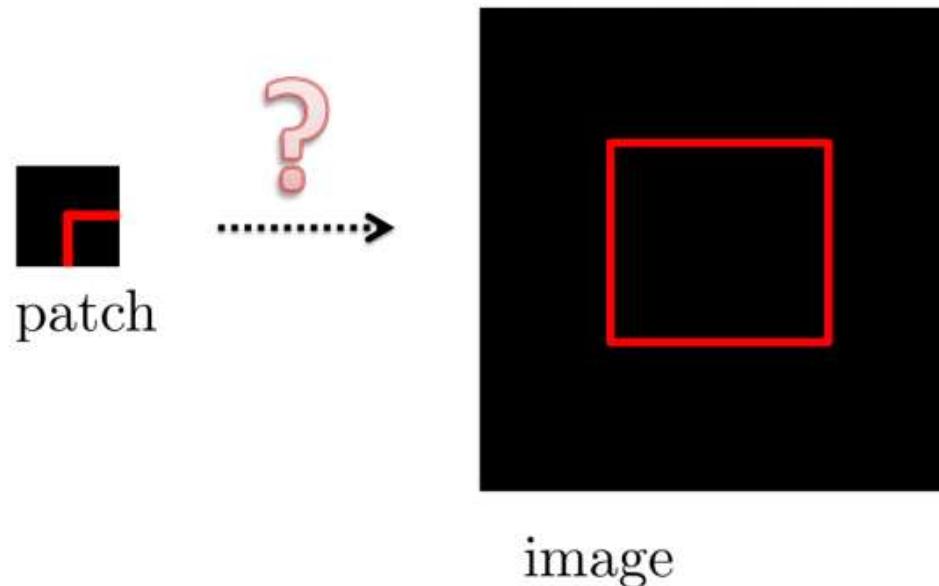
Keypoints

- Puntos con alto contraste son fáciles de localizar
- Problemas de consistencia espacial



Keypoints

- Puntos con alto contraste son fáciles de localizar
- Problemas de consistencia espacial
- Esquinas!



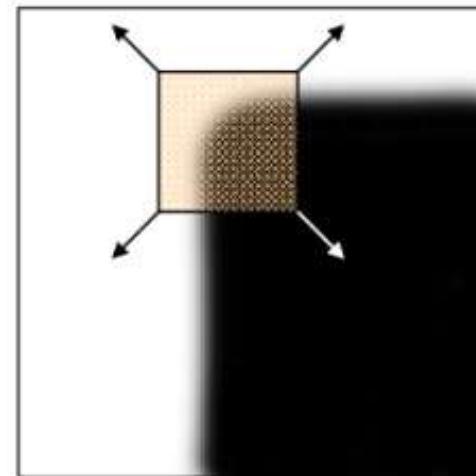
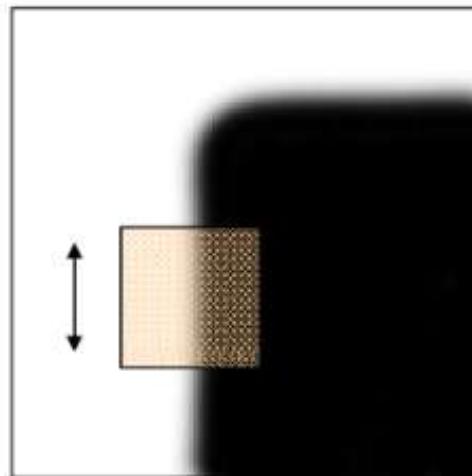
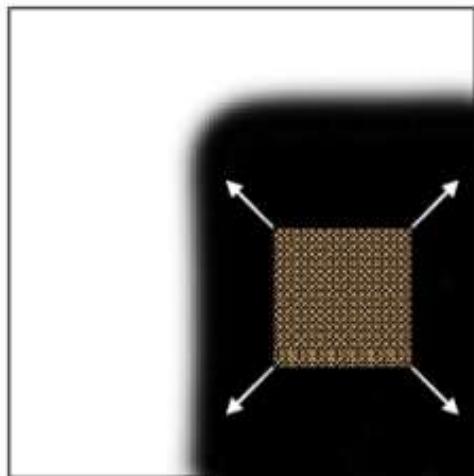
Keypoints

- Cómo podemos encontrar esquinas?



Keypoints

- Medir la cantidad de cambio en una ventana deslizante



Keypoints

- Comparar dos parches de imágenes usando diferencias al cuadrado

$$E_{\text{WSSD}}(u, v) = \sum_x \sum_y w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

Diagram illustrating the components of the Windowed Sum of Squared Differences (WSSD) metric:

- window function**: A blue box with an arrow pointing to the term $w(x, y)$ in the formula.
- shifted intensity**: A blue box with an arrow pointing to the term $I(x + u, y + v)$ in the formula.
- intensity**: A blue box with an arrow pointing to the term $I(x, y)$ in the formula.

The diagram shows two images side-by-side:

- Left Image:** A grayscale patch of an image with a red solid rectangle labeled $I(x, y)$ and a green dashed rectangle labeled $w(x, y)$ indicating the window function applied to the patch.
- Right Image:** A grayscale patch of the error function $E(u, v)$, which shows the squared difference between the original and shifted intensities. A green square highlights the minimum value at $E(3,2)$.

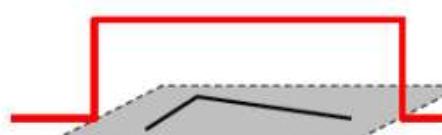
Keypoints

- Comparar dos parches de imágenes usando diferencias al cuadrado

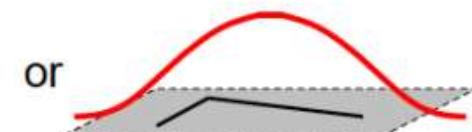
$$E_{\text{WSSD}}(u, v) = \sum_x \sum_y w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

↑
x y
window function shifted intensity intensity

Window function $w(x, y) =$



1 in window, 0 outside

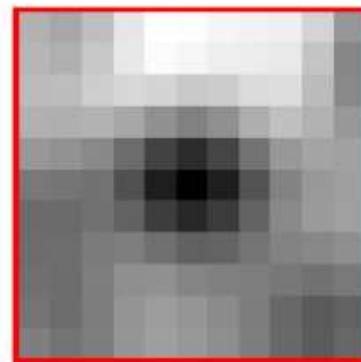


Gaussian

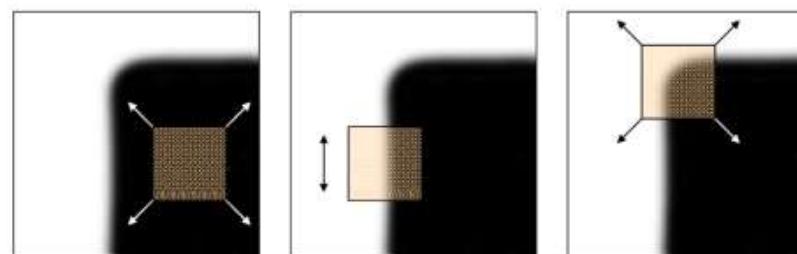
Keypoints

- Observamos la función de error

$$E(u, v)$$



- Recordemos lo que queremos lograr



Keypoints

- Usando expansión de Taylor simple de primer orden

$$I(x + u, y + v) \approx I(x, y) + u \cdot \frac{\partial I}{\partial x}(x, y) + v \cdot \frac{\partial I}{\partial y}(x, y)$$

$$\begin{aligned} E_{WSSD}(u, v) &= \sum_x \sum_y w(x, y) \left(I(x + u, y + v) - I(x, y) \right)^2 \\ &\approx \sum_x \sum_y w(x, y) \left(I(x, y) + u \cdot I_x + v \cdot I_y - I(x, y) \right)^2 \\ &= \sum_x \sum_y w(x, y) \left(u^2 I_x^2 + 2u \cdot v \cdot I_x \cdot I_y + v^2 I_y^2 \right) \\ &= \sum_x \sum_y w(x, y) \cdot \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_x \cdot I_y \\ I_x \cdot I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \end{aligned}$$

Keypoints

- Como (u, v) no dependen de (x, y) , podemos reescribir

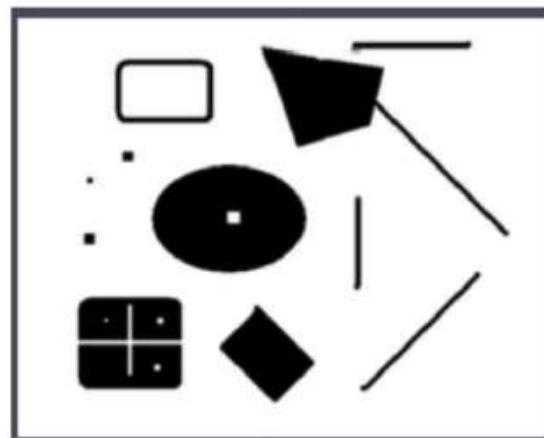
$$\begin{aligned} E_{\text{WSSD}}(u, v) &= \sum_x \sum_y w(x, y) \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_x \cdot I_y \\ I_x \cdot I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \\ &= \begin{bmatrix} u & v \end{bmatrix} \underbrace{\left(\sum_x \sum_y w(x, y) \begin{bmatrix} I_x^2 & I_x \cdot I_y \\ I_x \cdot I_y & I_y^2 \end{bmatrix} \right)}_{\text{Let's denotes this with } M} \begin{bmatrix} u \\ v \end{bmatrix} \\ &= \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \end{aligned}$$

- M es una matriz de 2×2 que contiene los gradientes

$$M = \sum_x \sum_y w(x, y) \begin{bmatrix} I_x^2 & I_x \cdot I_y \\ I_x \cdot I_y & I_y^2 \end{bmatrix}$$

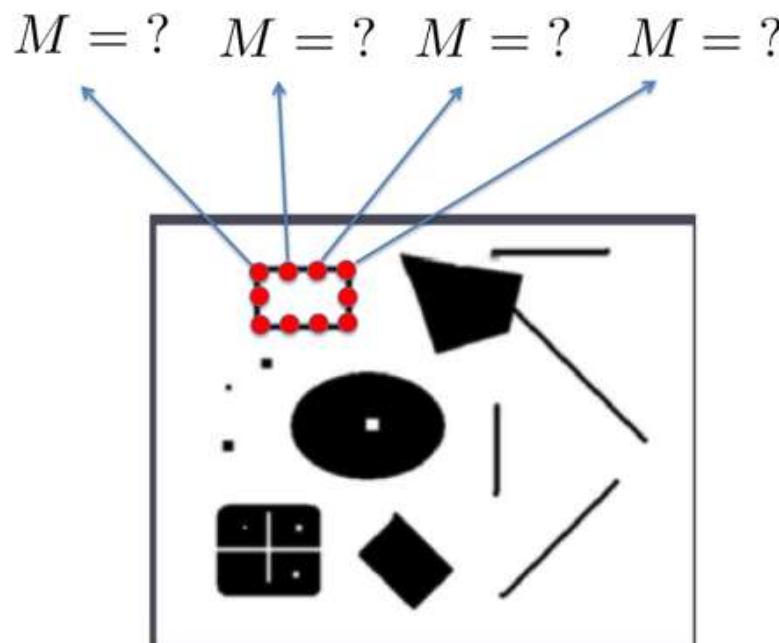
Keypoints

- Cómo se computa M



Keypoints

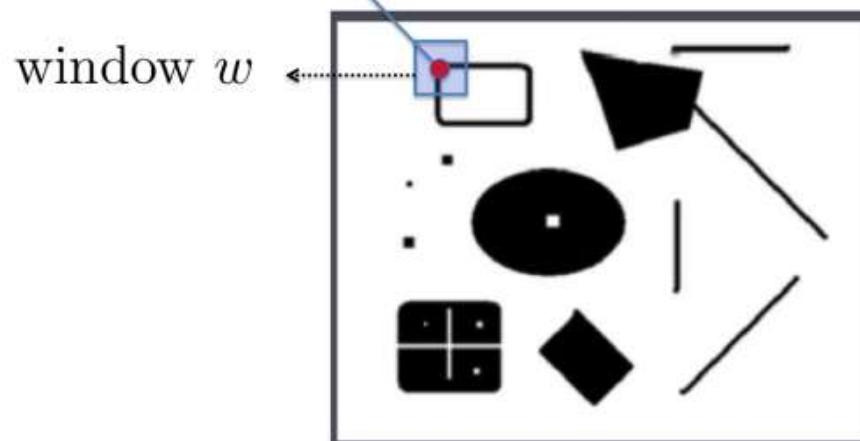
- Cómo se computa M
 - Necesitamos calcular M en cada posición de la imagen



Keypoints

- Cómo se computa M
 - Calcular promedio ponderado de gradientes en ventana

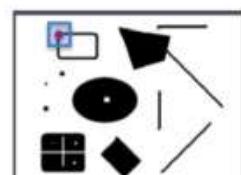
$$M = \sum_x \sum_y w(x, y) \begin{bmatrix} I_x^2 & I_x \cdot I_y \\ I_x \cdot I_y & I_y^2 \end{bmatrix}$$



Keypoints

- Cómo se computa M
 - Ya sabemos cómo calcular gradientes

$$M = \sum_x \sum_y w(x, y) \begin{bmatrix} I_x^2 & I_x \cdot I_y \\ I_x \cdot I_y & I_y^2 \end{bmatrix}$$



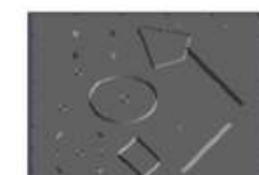
image



$$I_x = \frac{\partial I}{\partial x}$$



$$I_y = \frac{\partial I}{\partial y}$$



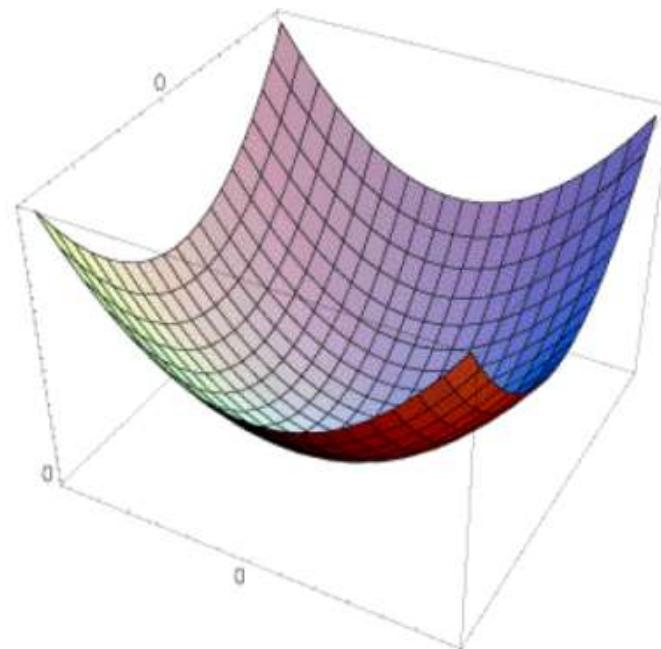
$$I_x \cdot I_y$$

Keypoints

- Función E es una función cuadrática

$$E_{\text{WSSD}}(u, v) = [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \sum_x \sum_y w(x, y) \begin{bmatrix} I_x^2 & I_x \cdot I_y \\ I_x \cdot I_y & I_y^2 \end{bmatrix}$$



Keypoints

- Tomemos cortes horizontales de E

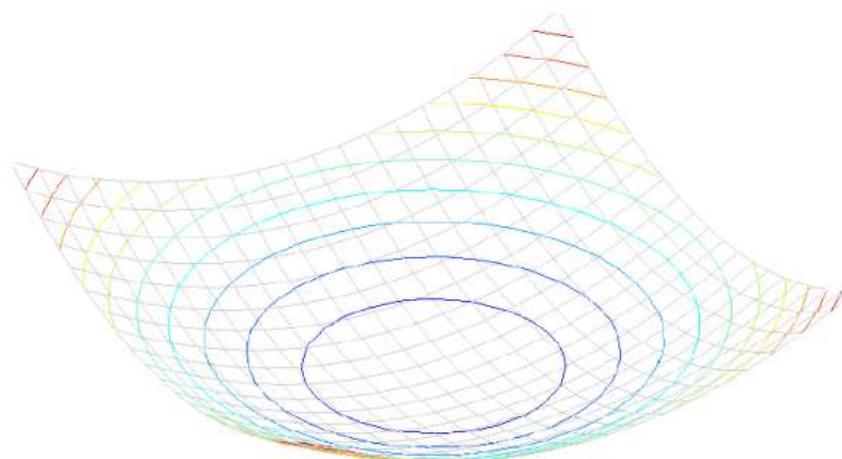
$$[u \ v] M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$

Keypoints

- Tomemos cortes horizontales de E

$$[u \ v] M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$

- Es la ecuación del elipse



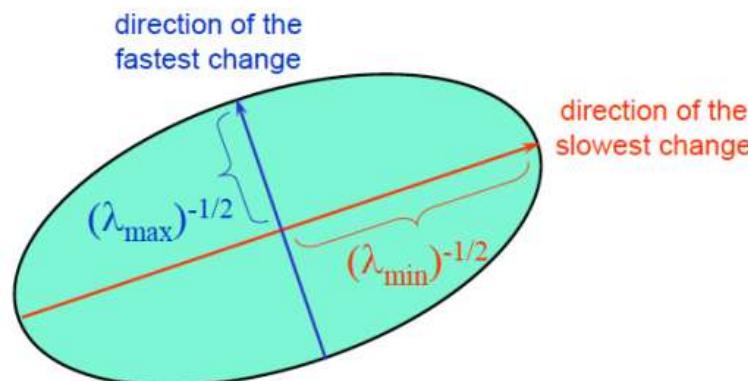
Keypoints

- Matriz M es simétrica
- Se puede diagonalizar

$$M = \sum_x \sum_y w(x, y) \begin{bmatrix} I_x^2 & I_x \cdot I_y \\ I_x \cdot I_y & I_y^2 \end{bmatrix}$$

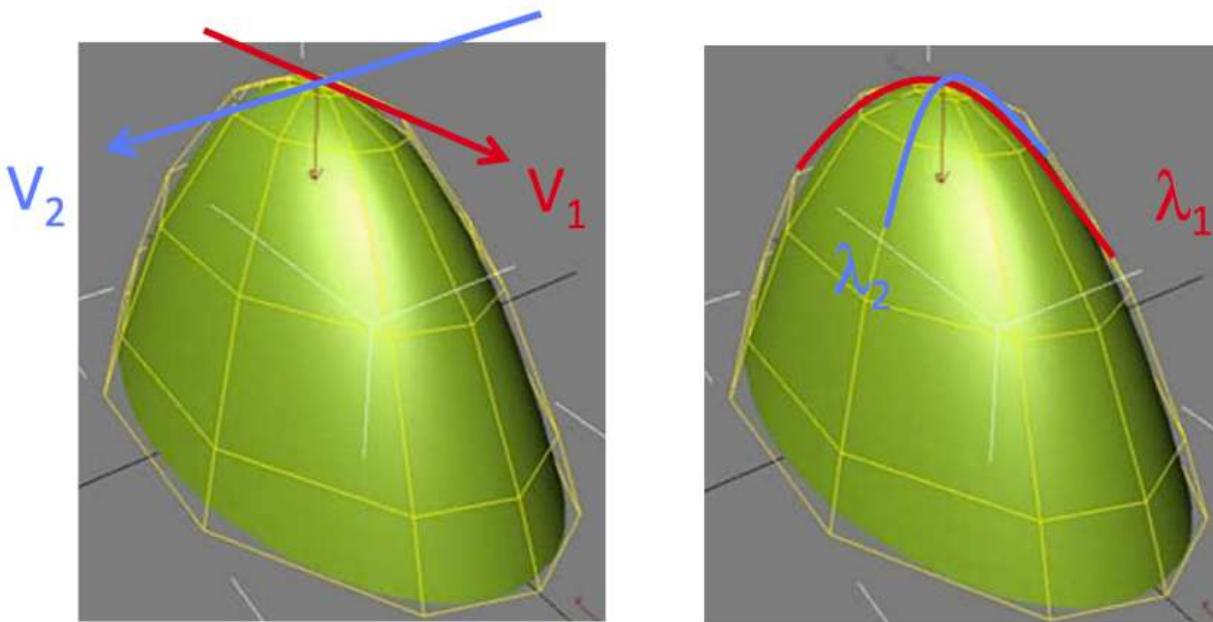
$$M = V \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} V^{-1}$$

- Columnas de V son ejes mayor y menor del elipse



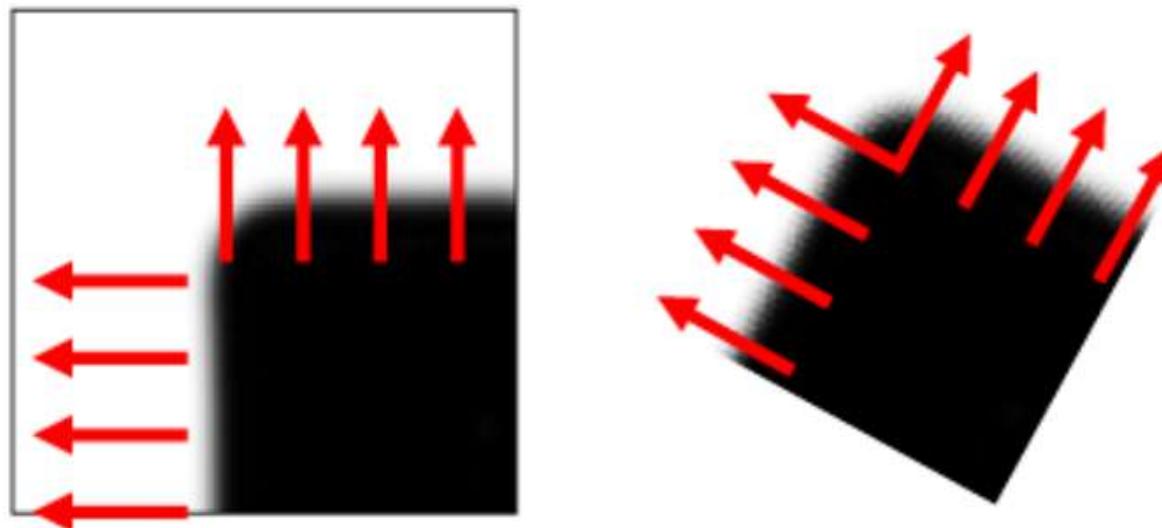
Keypoints

- Columnas de V son direcciones principales
- λ_1, λ_2 son curvaturas principales

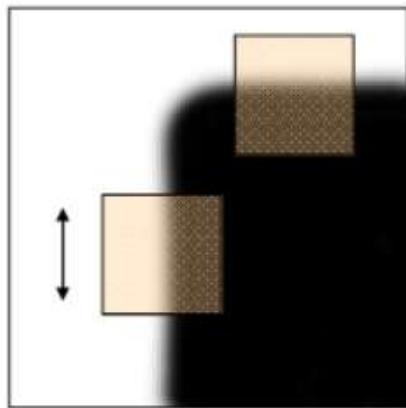


Keypoints

- Los valores propios de M revelan la cantidad de intensidad de cambio en las dos direcciones de gradiente principales



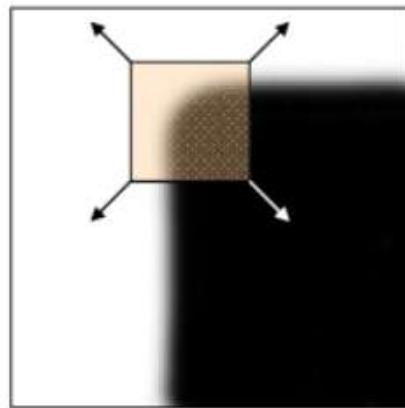
Keypoints



“edge”:

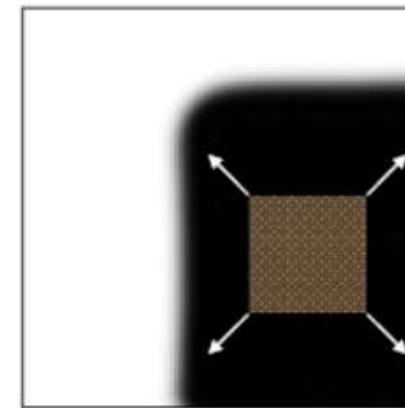
$$\lambda_1 \gg \lambda_2$$

$$\lambda_2 \gg \lambda_1$$



“corner”:

λ_1 and λ_2 are large,
 $\lambda_1 \sim \lambda_2$;



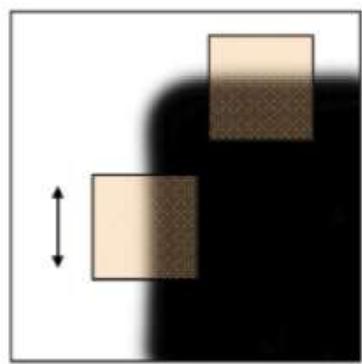
“flat” region

λ_1 and λ_2 are
small;

Keypoints - Harris

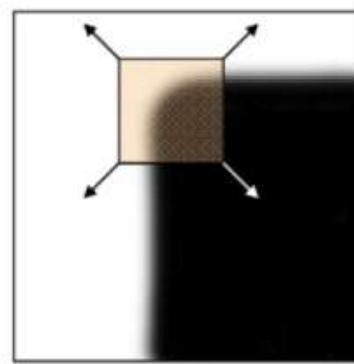
- Algoritmo de Harris

$$R = \det(M) - \alpha \cdot \text{trace}(M)^2 = \lambda_0\lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$$



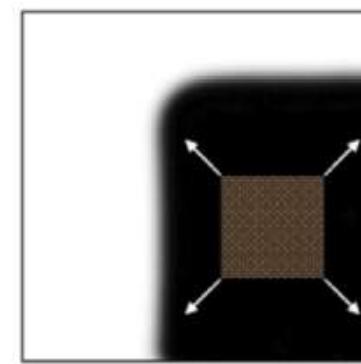
“edge”:

$$R < 0$$



“corner”:

$$R > 0$$

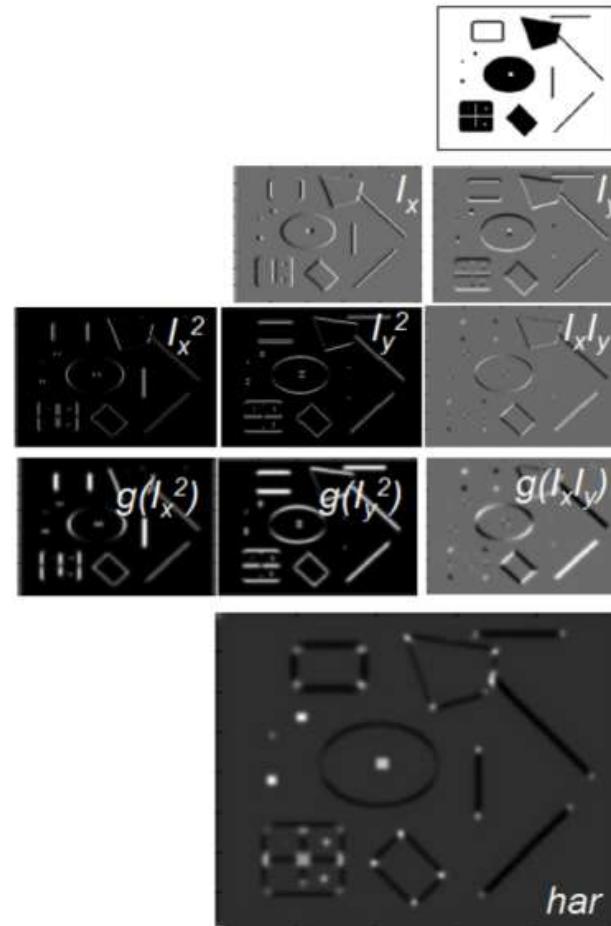


“flat” region

$$|R| \text{ small}$$

Keypoints - Harris

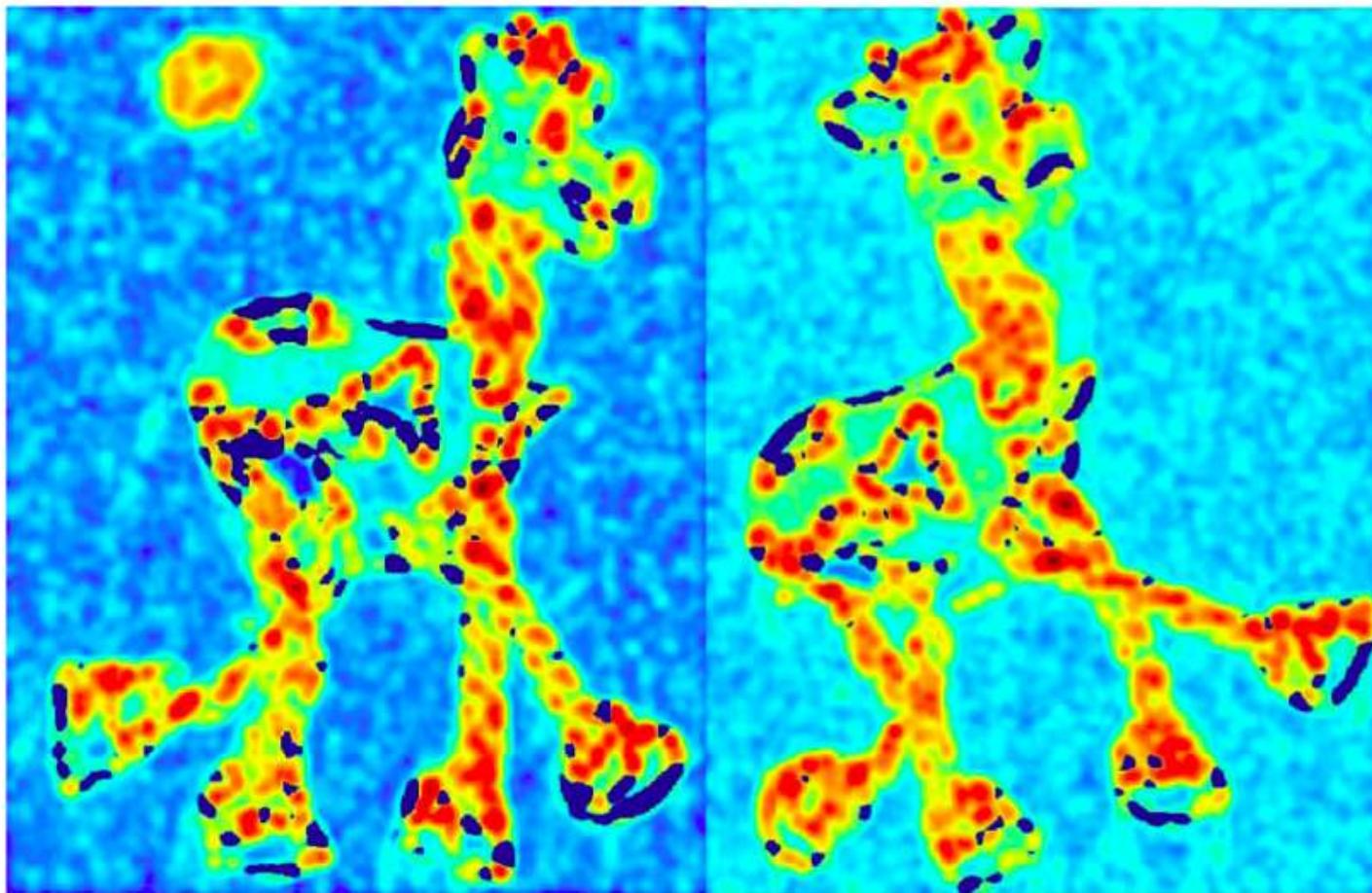
- ① Compute gradients I_x and I_y
- ② Compute I_x^2 , I_y^2 , $I_x \cdot I_y$
- ③ Average (Gaussian) \rightarrow gives M
- ④ Compute
 $R = \det(M) - \alpha \text{trace}(M)^2$ for each
image window (*cornerness score*)
- ⑤ Find points with large R ($R >$
threshold).
- ⑥ Take only points of local maxima,
i.e., perform non-maximum
suppression



Keypoints – Harris - Ejemplo



Keypoints – Harris - Ejemplo



Keypoints – Harris - Ejemplo



Keypoints – Harris - Ejemplo



Keypoints – Harris - Ejemplo



Keypoints – Harris - Ejemplo



Keypoints – Harris - Ejemplo



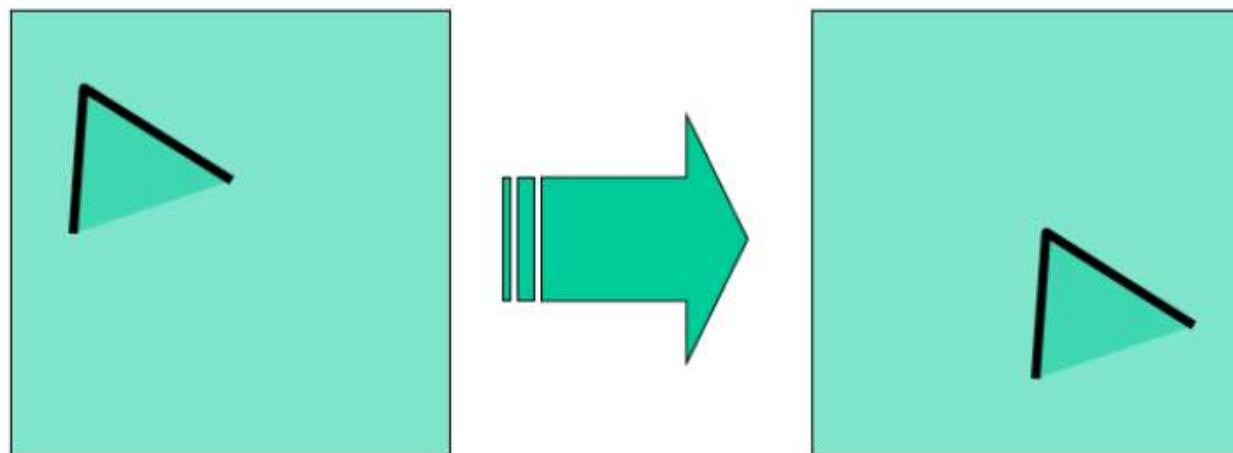
Keypoints - Propiedades

- Puntos deberían ser invariantes a transformaciones



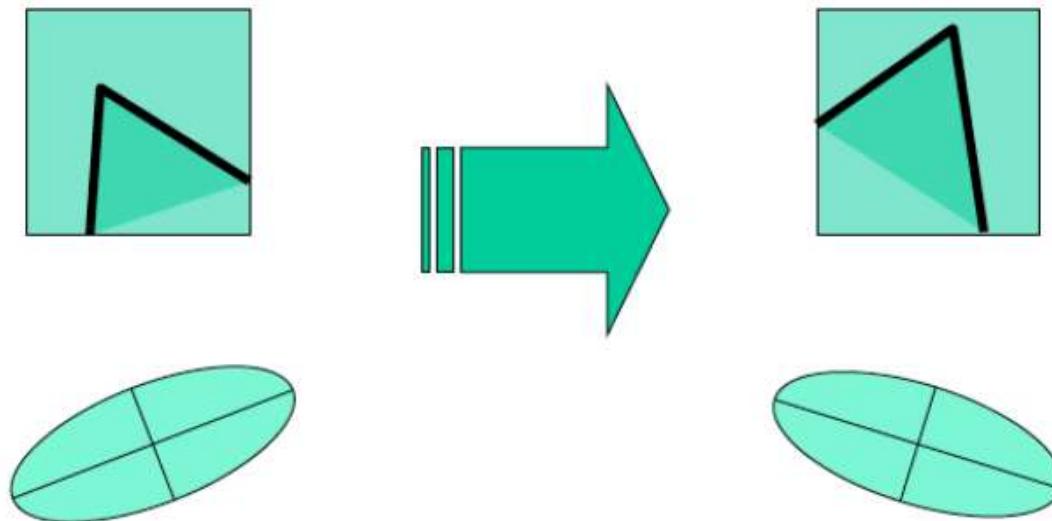
Keypoints - Propiedades

- Harris es invariante a translaciones



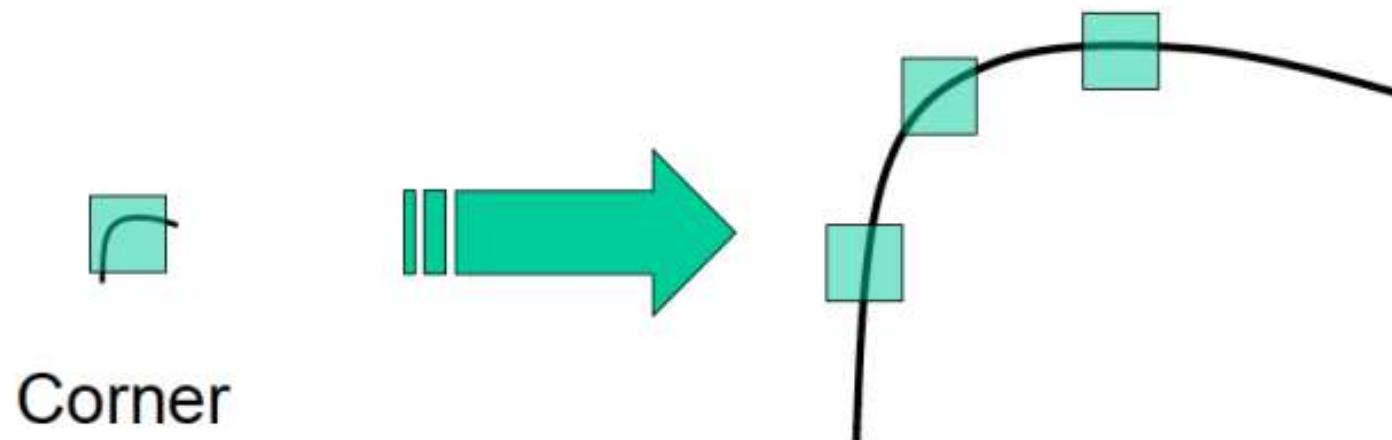
Keypoints - Propiedades

- Harris es invariante a rotaciones



Keypoints - Propiedades

- Harris no es invariante a escala



Detección de keypoints invariantes a escala

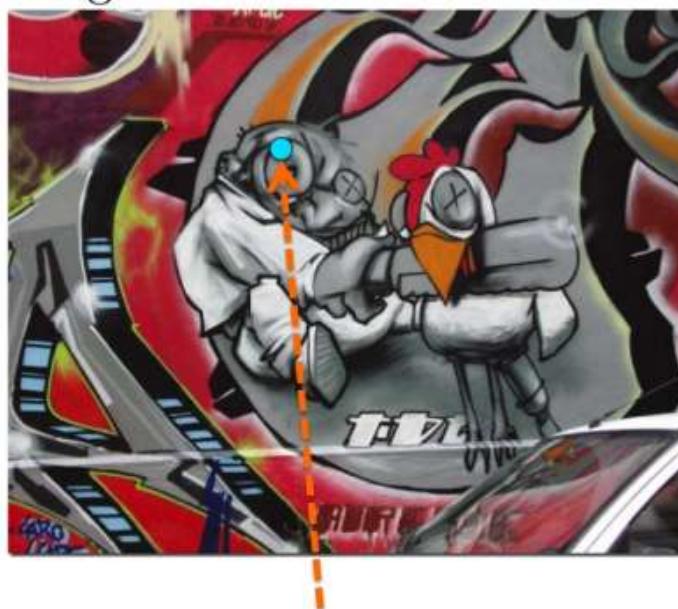
Invarianza a la escala

- Cómo podemos seleccionar puntos en cada imagen, tal que las detecciones sean repetibles a diferentes escalas?

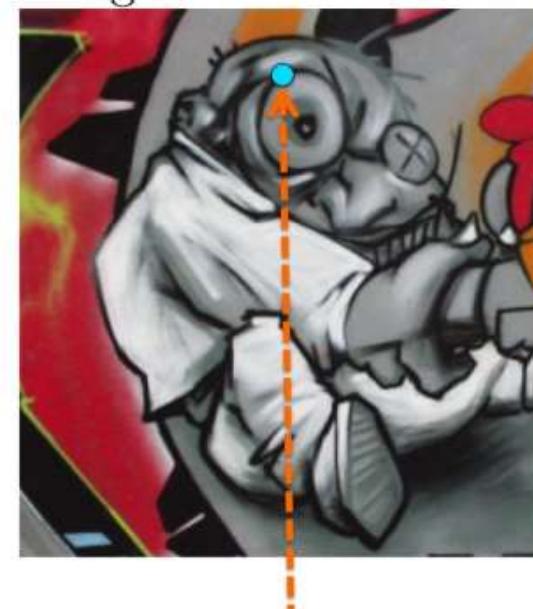


Invarianza a la escala

- Cómo podemos seleccionar puntos en cada imagen, tal que las detecciones sean repetibles a diferentes escalas?



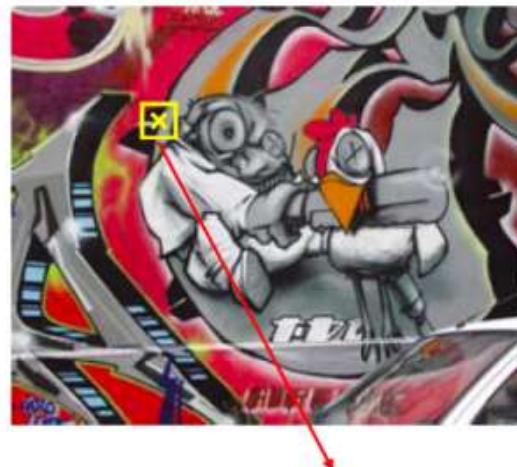
Si detecto este punto aquí



También quiero detectar este punto aquí

Invarianza a la escala

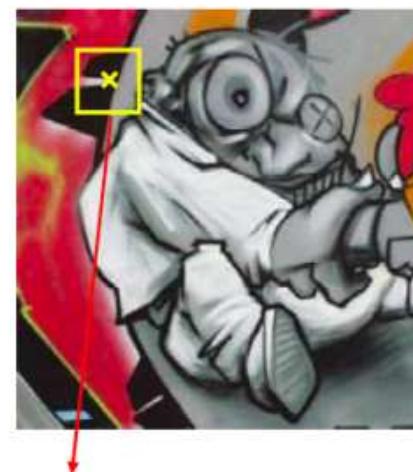
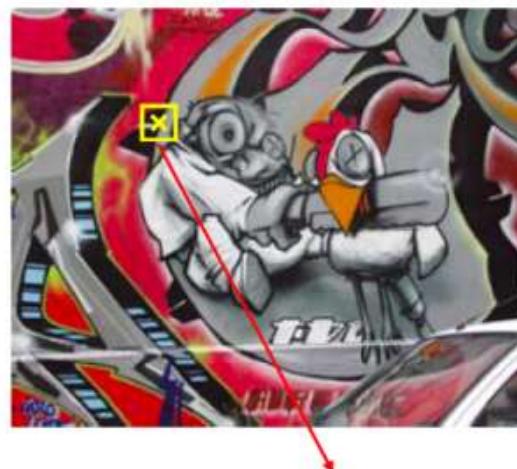
- Cómo podemos seleccionar puntos en cada imagen, tal que las detecciones sean repetibles a diferentes escalas?
- Extraer información estable en posición y escala



$$f(I_{i_1 \dots i_m}(x, \sigma)) = f(I_{i_1 \dots i_m}(x', \sigma'))$$

Invarianza a la escala

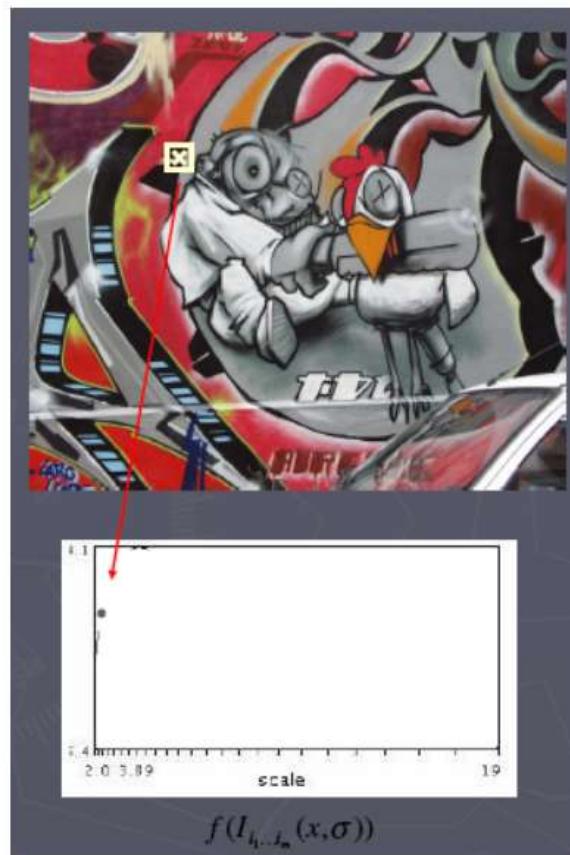
- Cómo podemos seleccionar puntos en cada imagen, tal que las detecciones sean repetibles a diferentes escalas?
- Encontrar la escala que da el máximo local de una función en posición y escala



$$f(I_{i_1 \dots i_m}(x, \sigma)) = f(I_{i_1 \dots i_m}(x', \sigma'))$$

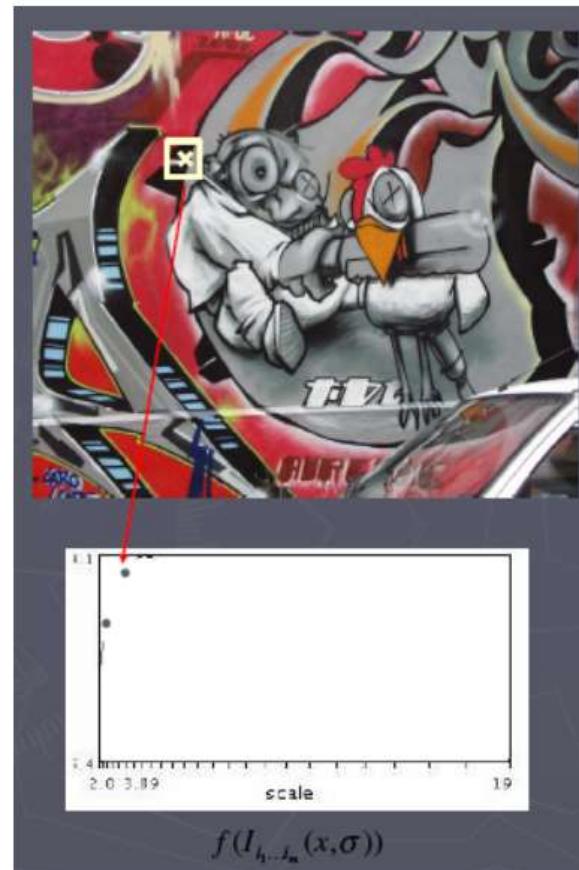
Detección de escala automática

- Una función que depende de la escala



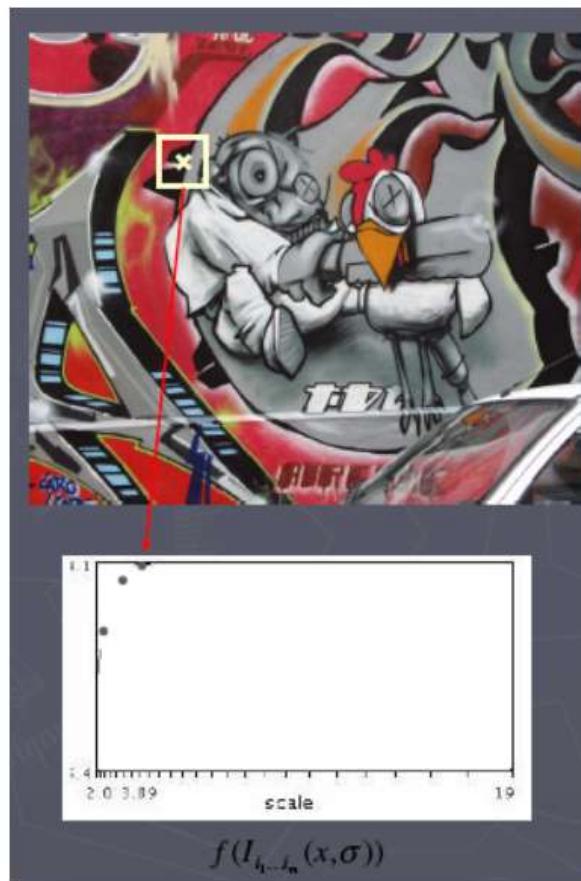
Detección de escala automática

- Una función que depende de la escala



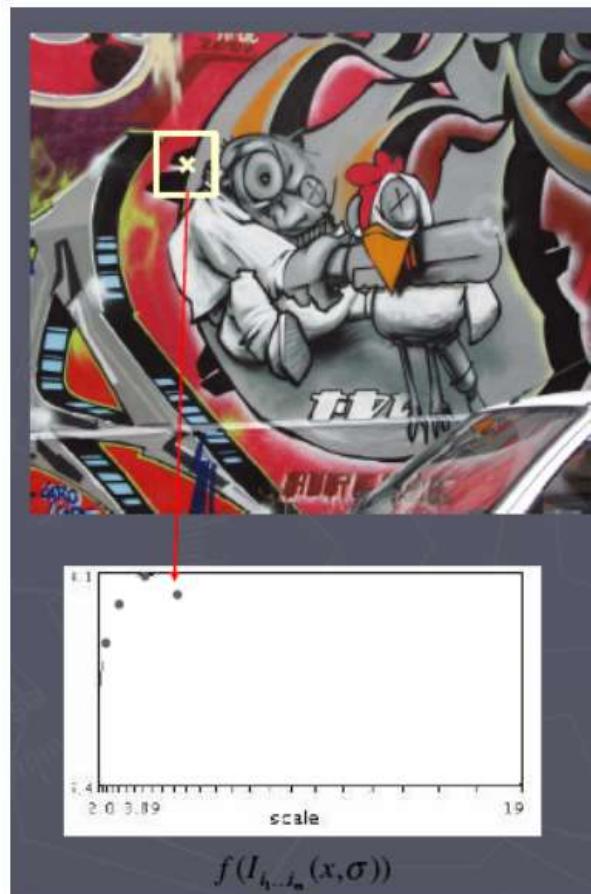
Detección de escala automática

- Una función que depende de la escala



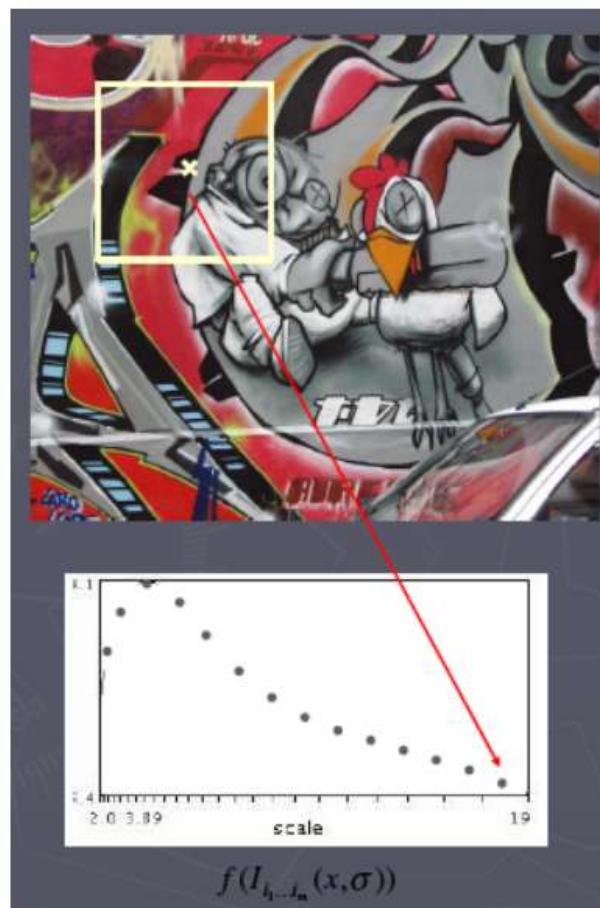
Detección de escala automática

- Una función que depende de la escala



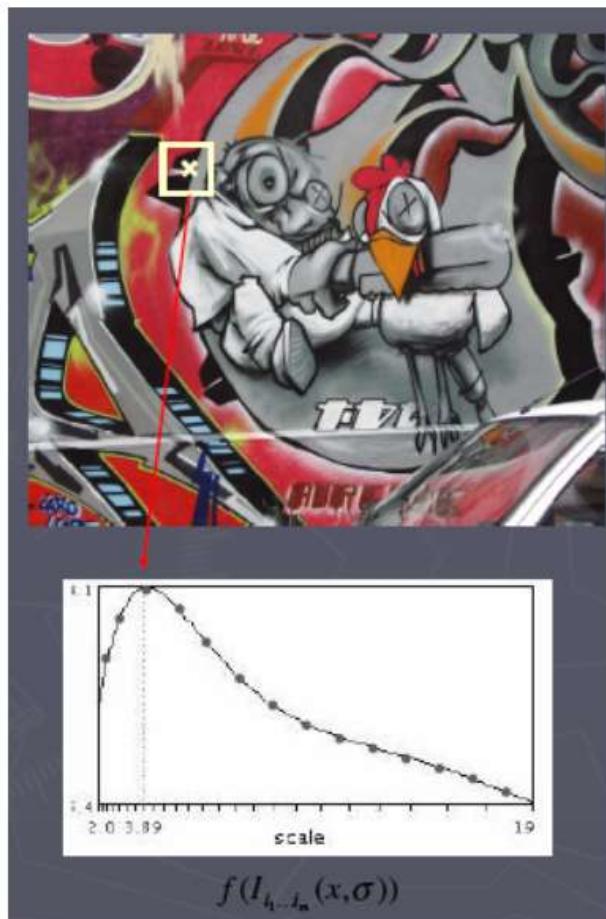
Detección de escala automática

- Una función que depende de la escala



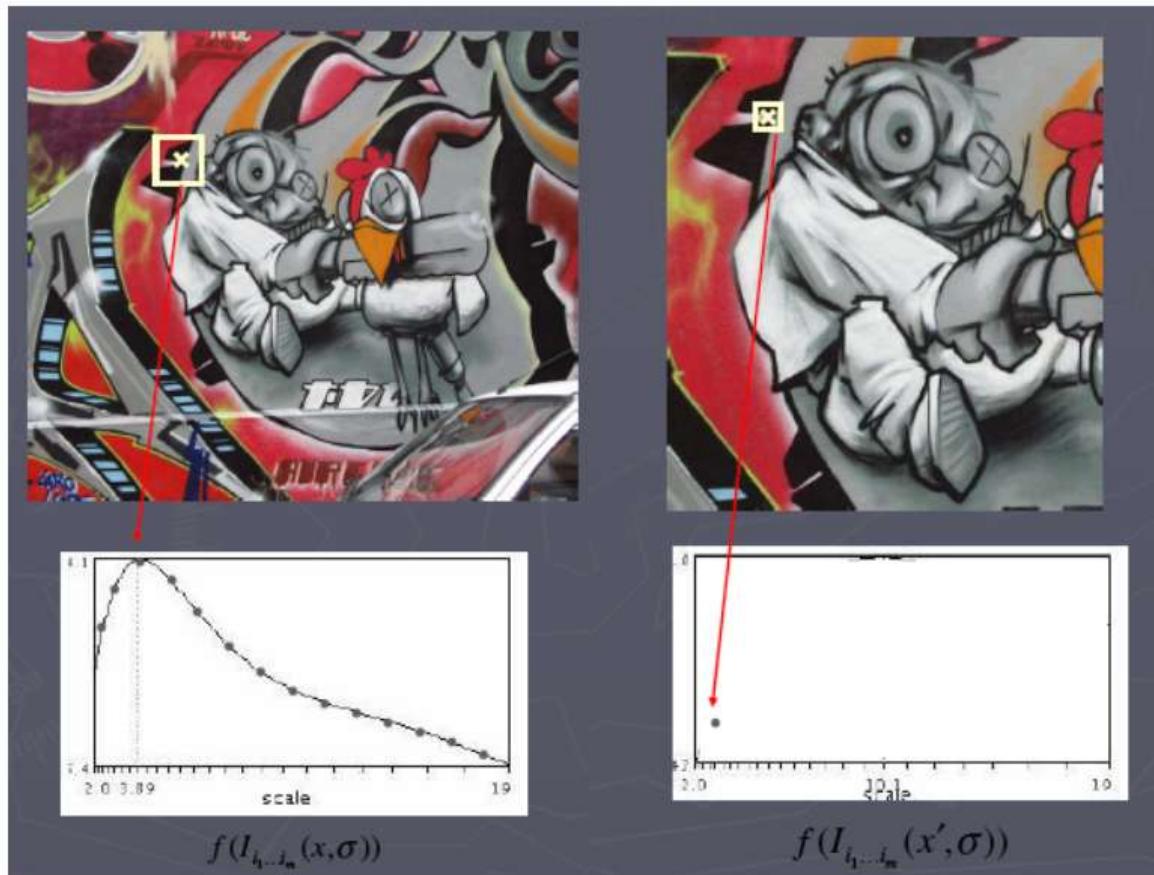
Detección de escala automática

- Una función que depende de la escala



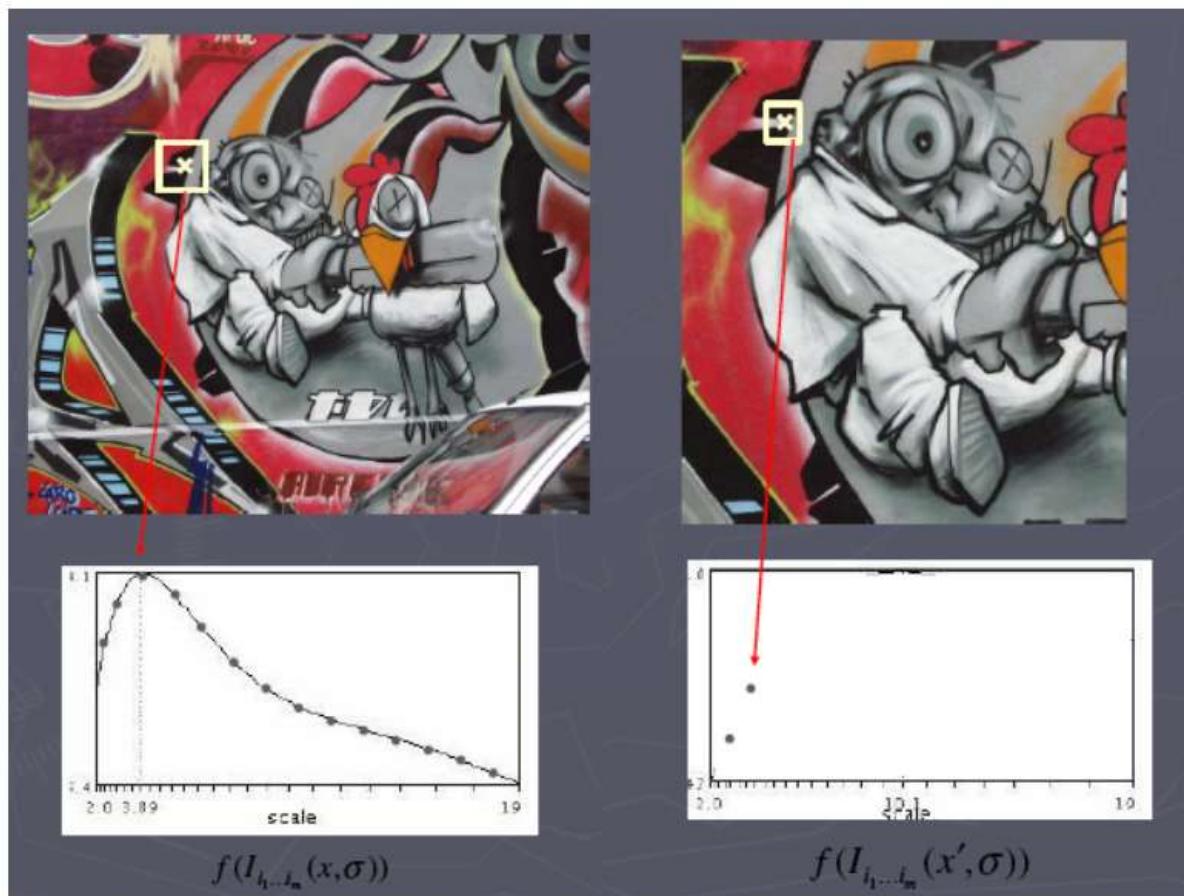
Detección de escala automática

- Una función que depende de la escala



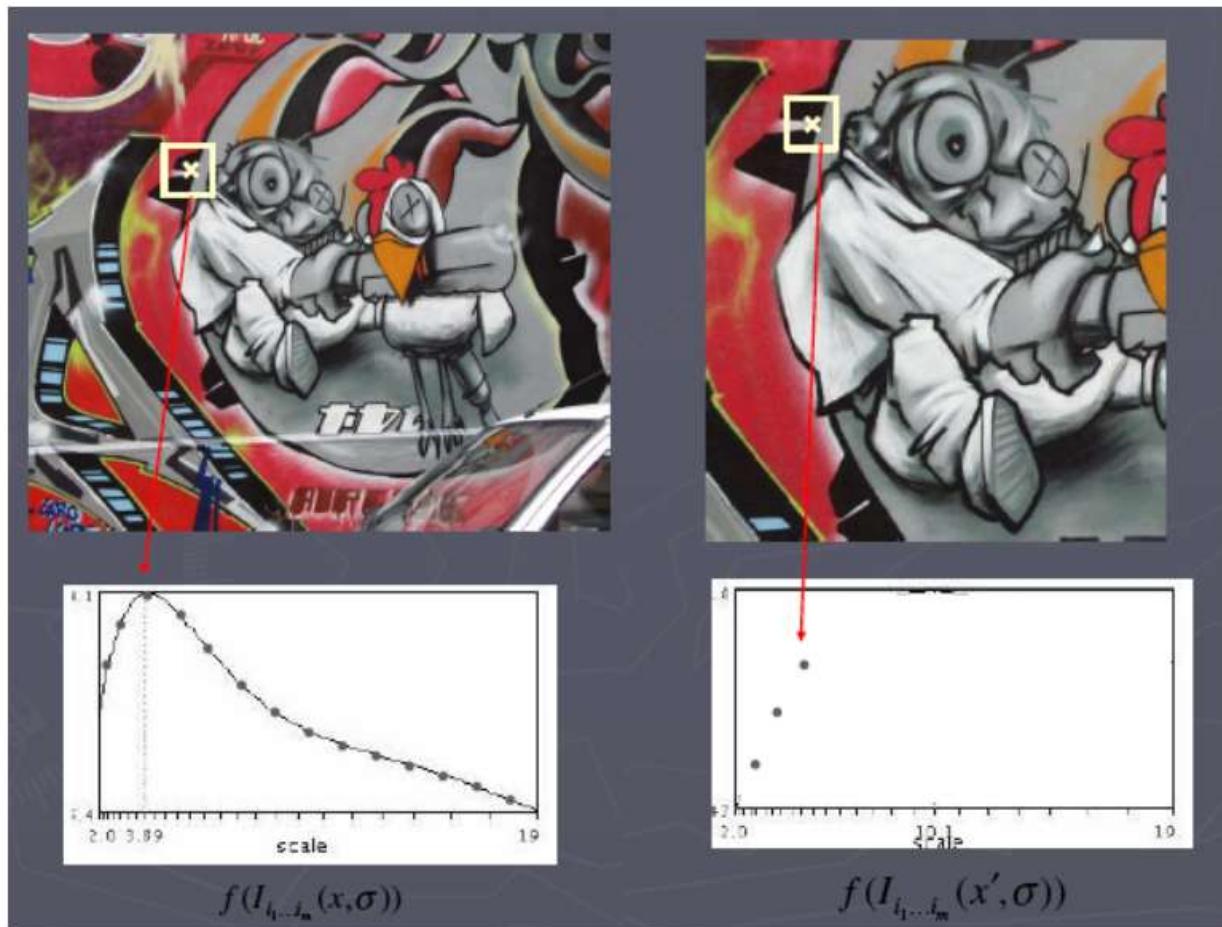
Detección de escala automática

- Una función que depende de la escala



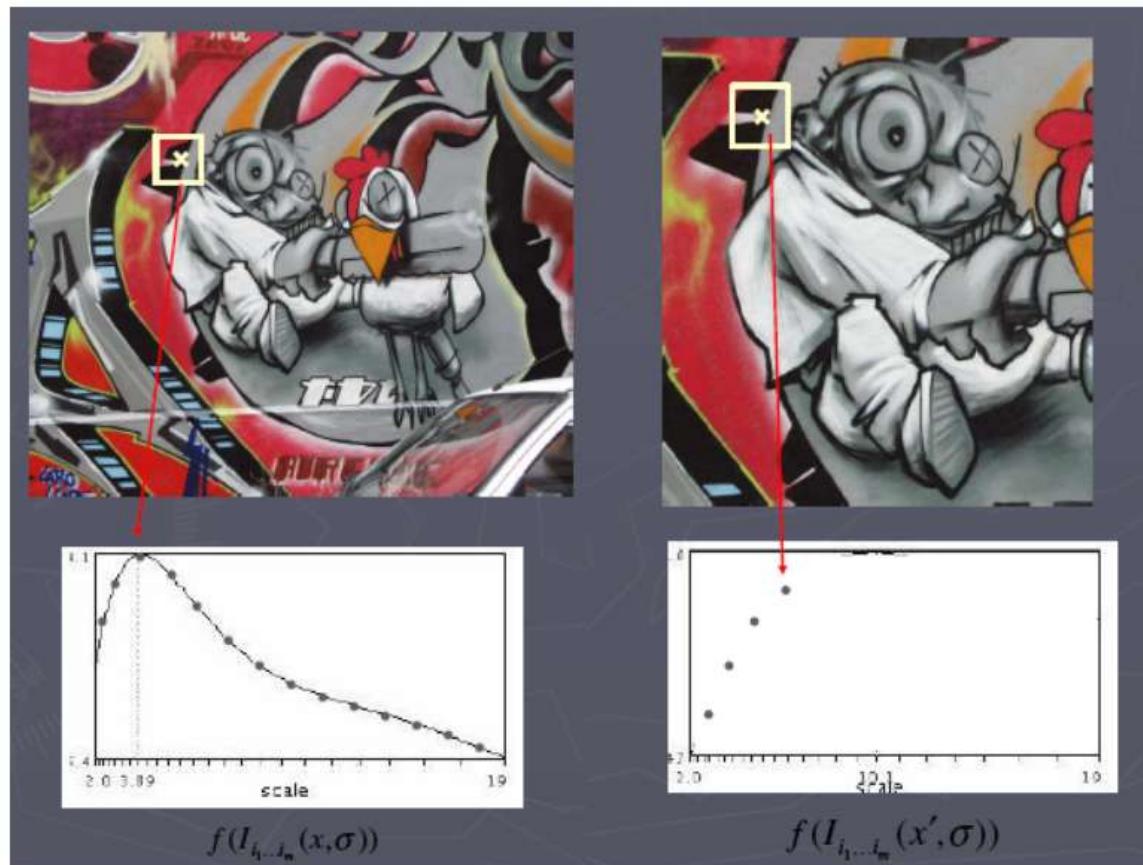
Detección de escala automática

- Una función que depende de la escala



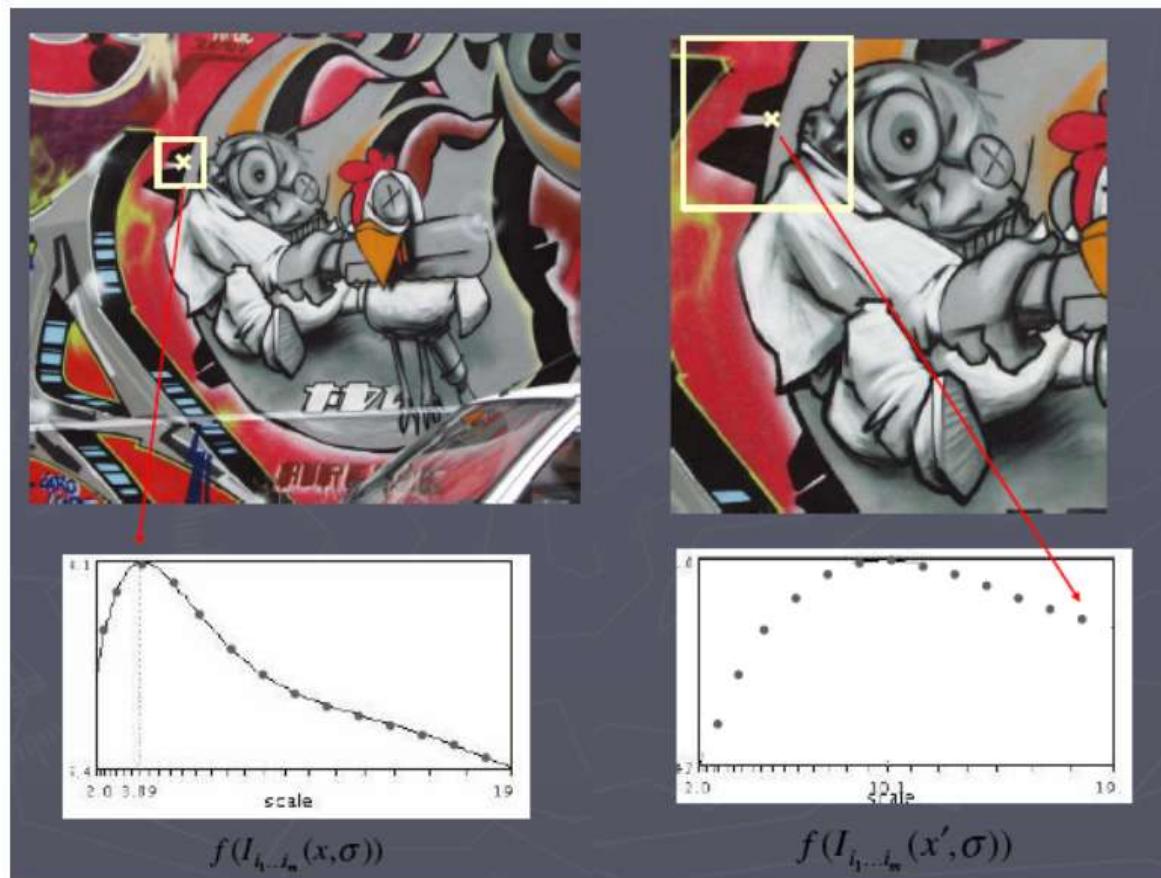
Detección de escala automática

- Una función que depende de la escala



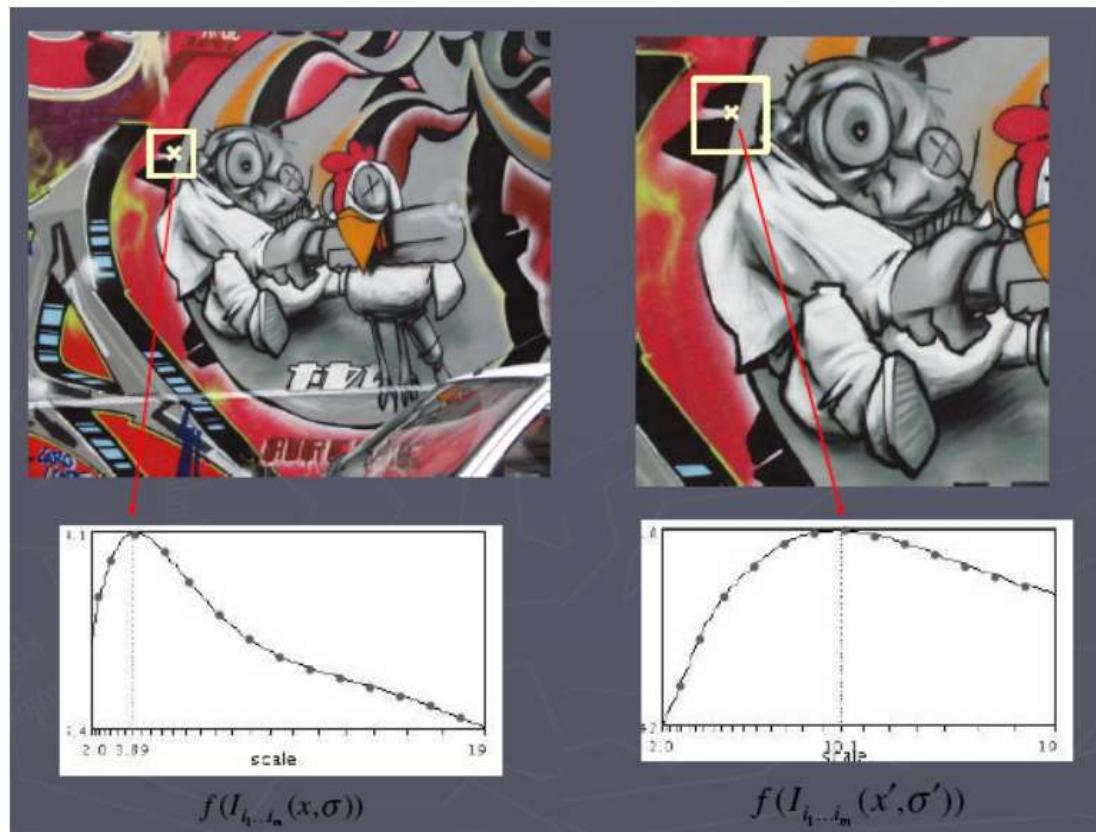
Detección de escala automática

- Una función que depende de la escala



Detección de escala automática

- Una función que depende de la escala



Cómo calculamos esa función?

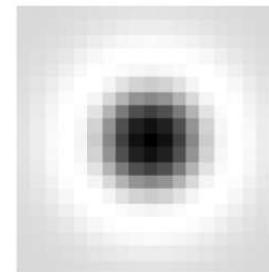
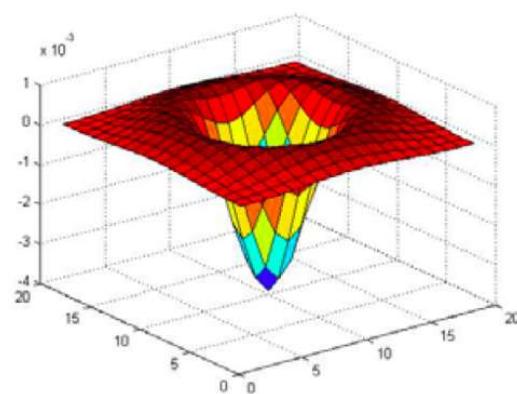
- Dos opciones
 - Laplaciano del Gaussiano
 - Diferencia de Gaussianos

Laplaciano del Gaussiano

- Se puede usar para detectar bordes

$$\nabla^2 g(x, y, \sigma) = \frac{\partial^2 g(x, y, \sigma)}{\partial x^2} + \frac{\partial^2 g(x, y, \sigma)}{\partial y^2}, \quad g \text{ es un Gaussiano}$$

- Es un operador simétrico circular
 - Diferencias en todas las direcciones



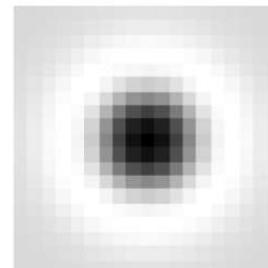
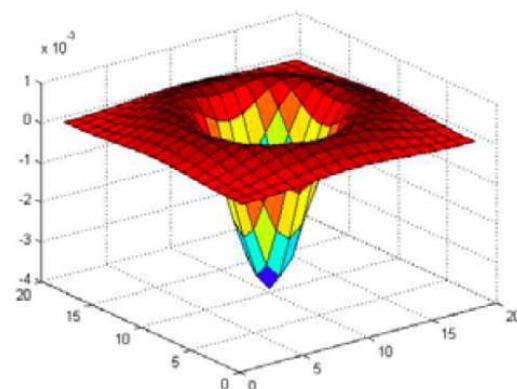
Laplaciano del Gaussiano

- Se puede usar para detectar bordes

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

g es un Gaussiano

- Es un operador simétrico circular
 - Diferencias en todas las direcciones

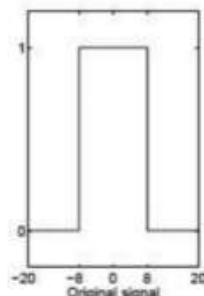


$$\nabla^2 g(x, y, \sigma) = -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) \exp^{-\frac{x^2+y^2}{2\sigma^2}}$$

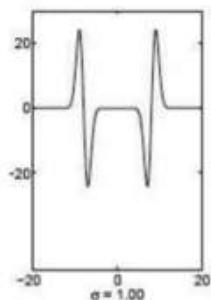
Laplaciano del Gaussiano

- Podemos detectar “cosas” de distintos tamaños

Original signal

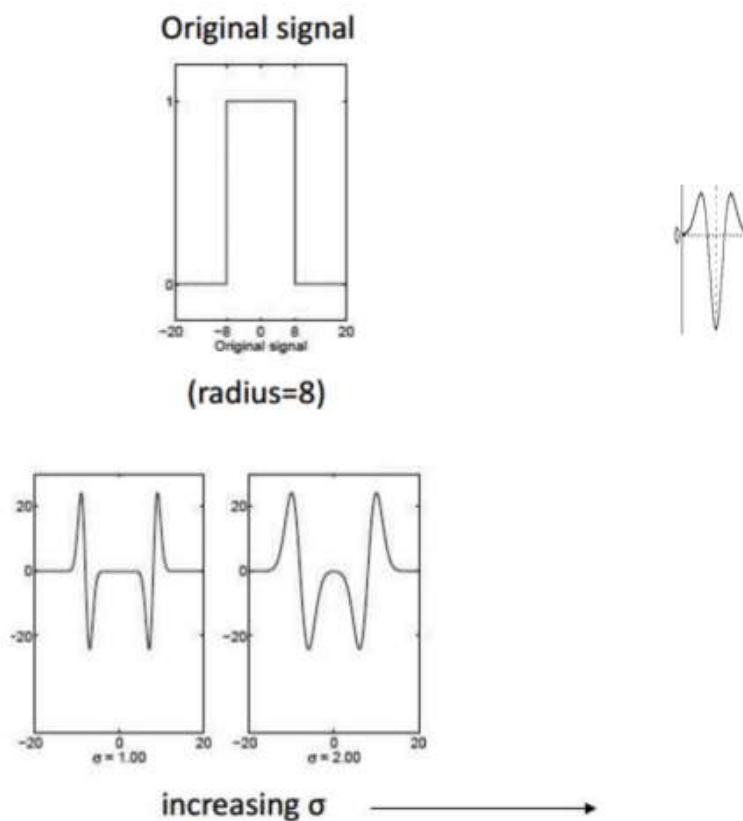


(radius=8)



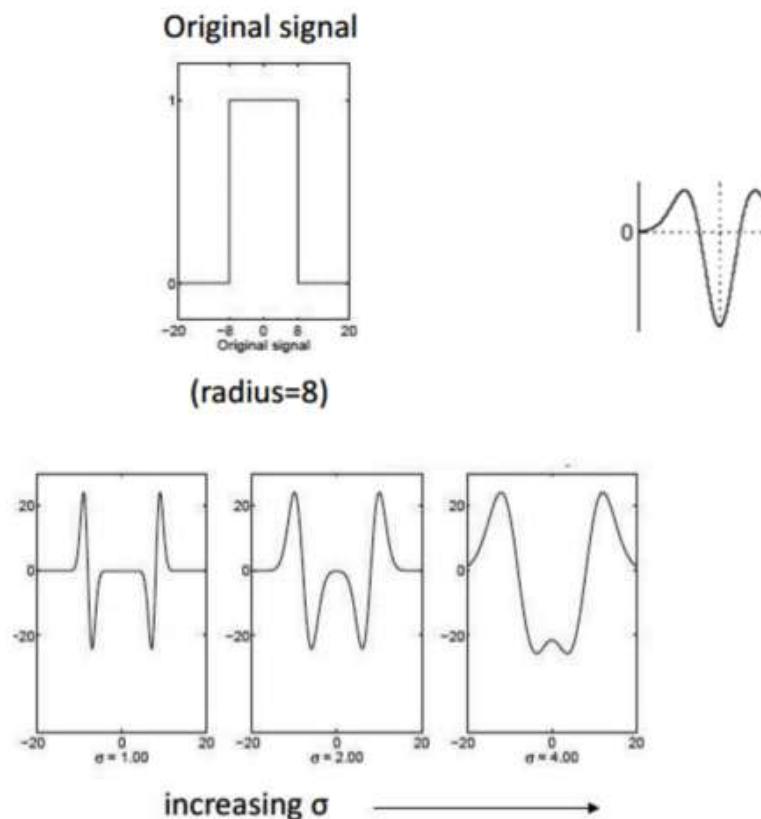
Laplaciano del Gaussiano

- Podemos detectar “cosas” de distintos tamaños



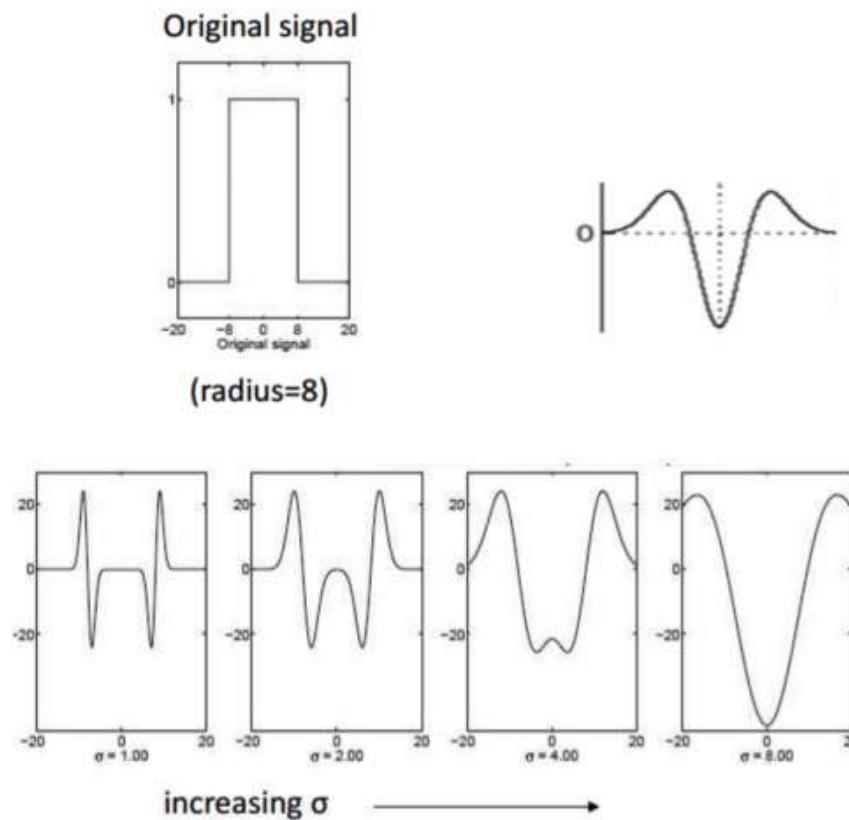
Laplaciano del Gaussiano

- Podemos detectar “cosas” de distintos tamaños



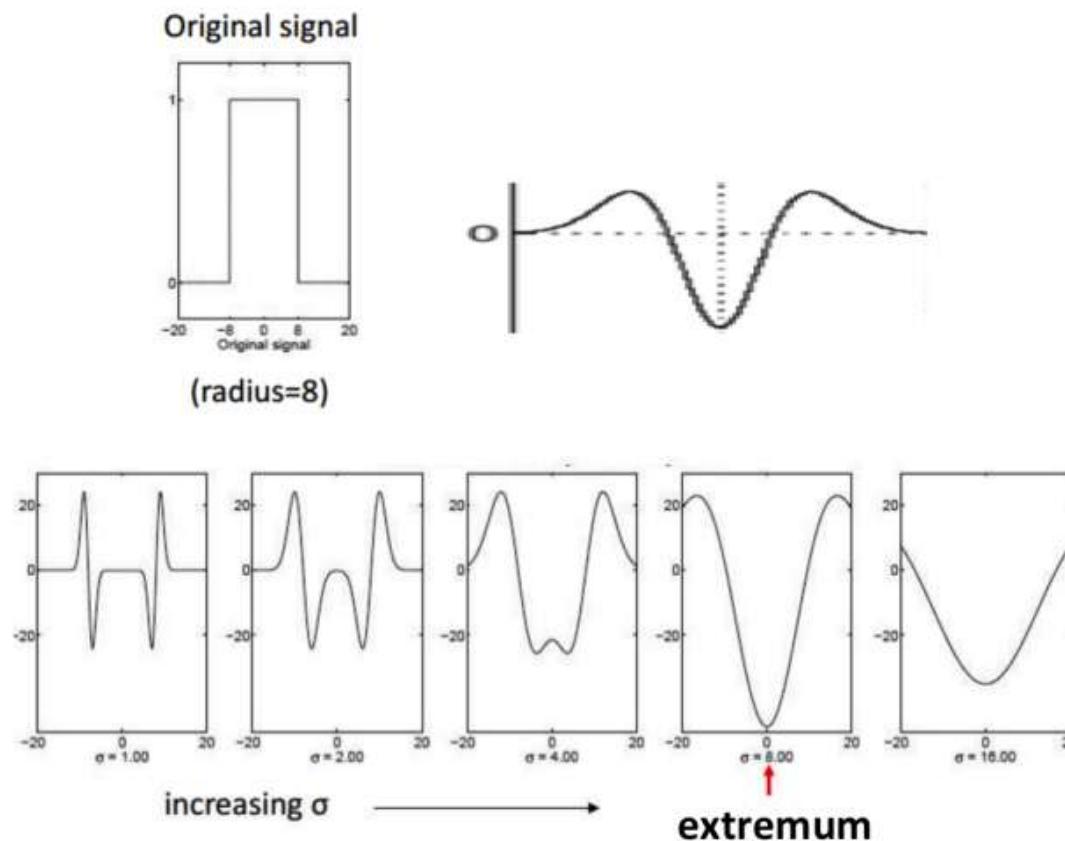
Laplaciano del Gaussiano

- Podemos detectar “cosas” de distintos tamaños



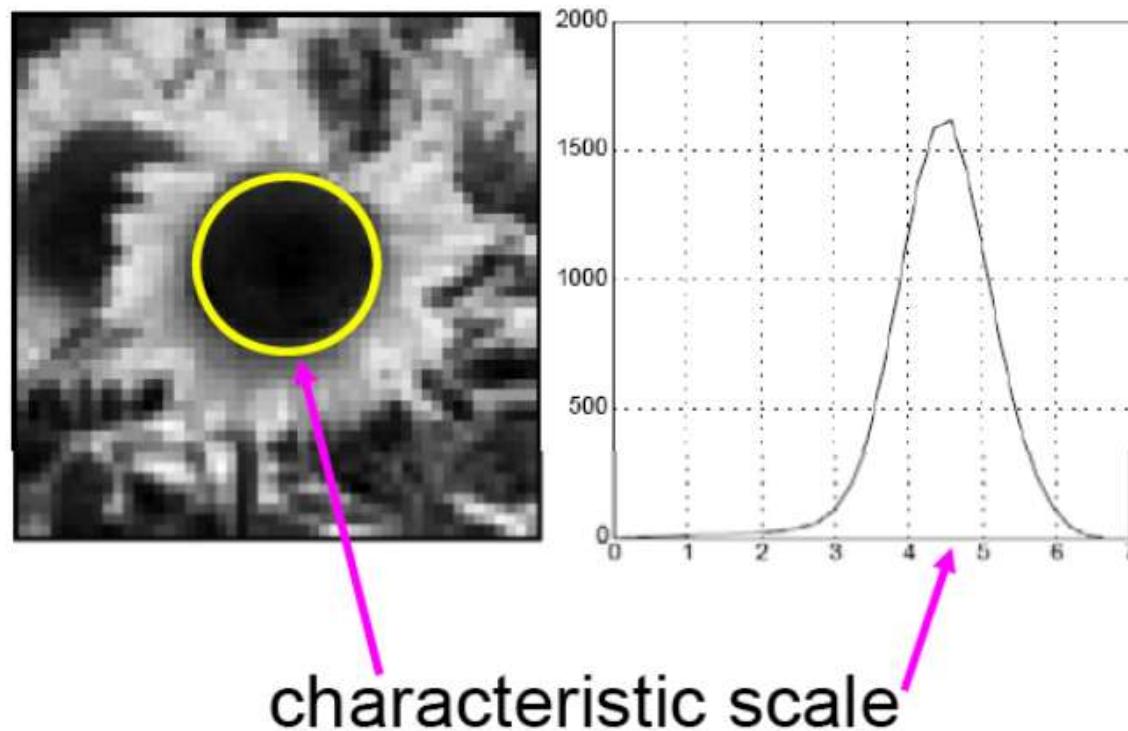
Laplaciano del Gaussiano

- Podemos detectar “cosas” de distintos tamaños



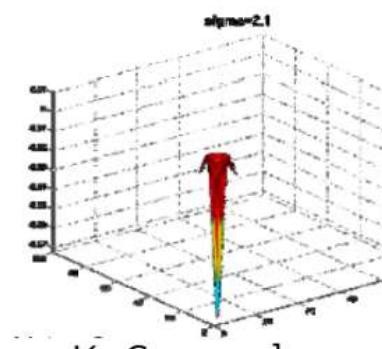
Laplaciano del Gaussiano

- La escala característica es la escala que produce máximos o mínimos del Laplaciano



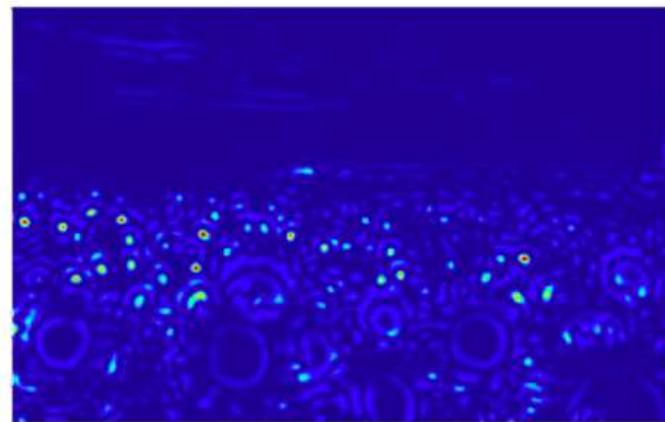
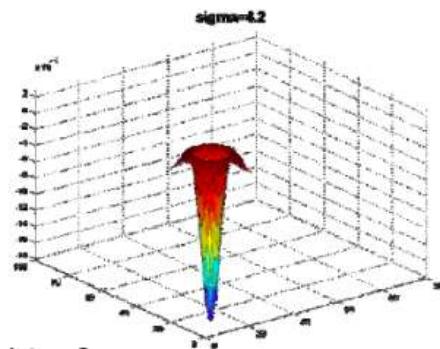
Laplaciano del Gaussiano

- Ejemplo



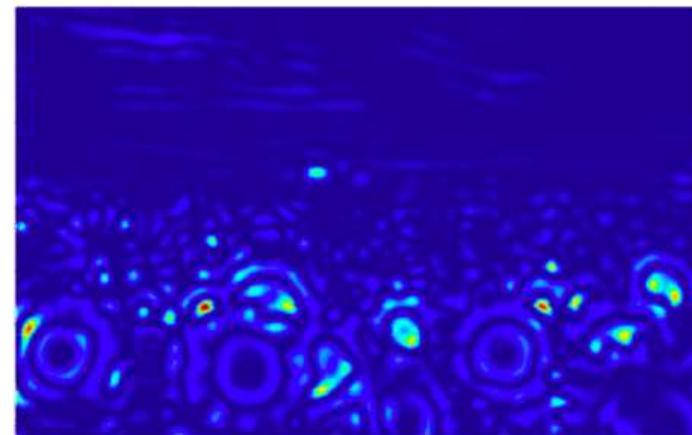
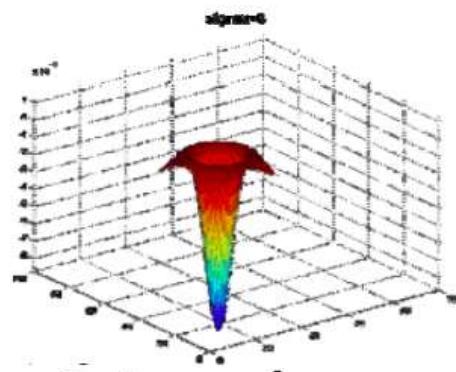
Laplaciano del Gaussiano

- Ejemplo



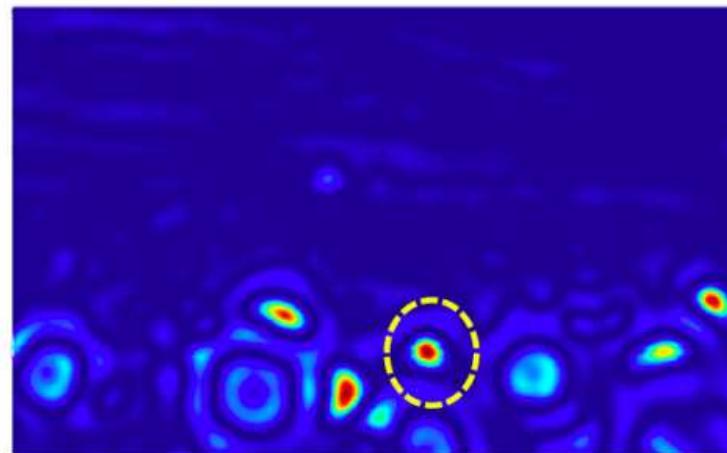
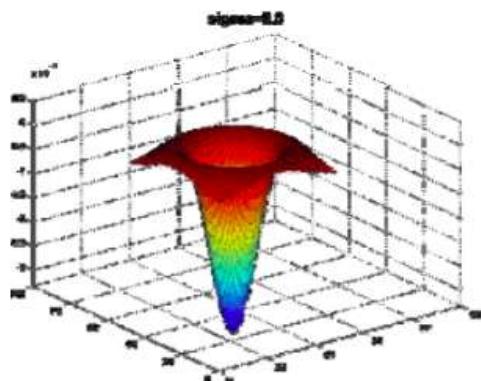
Laplaciano del Gaussiano

- Ejemplo



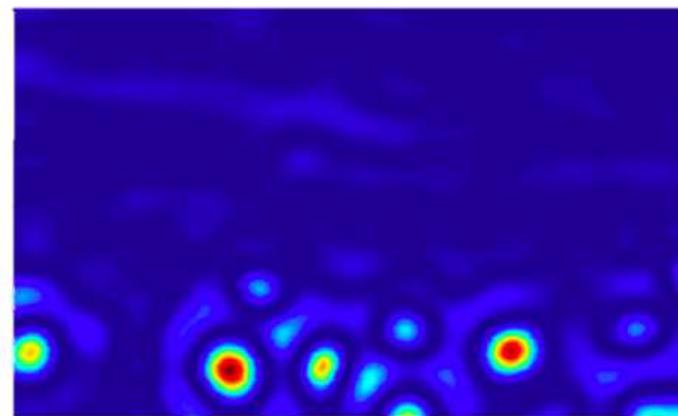
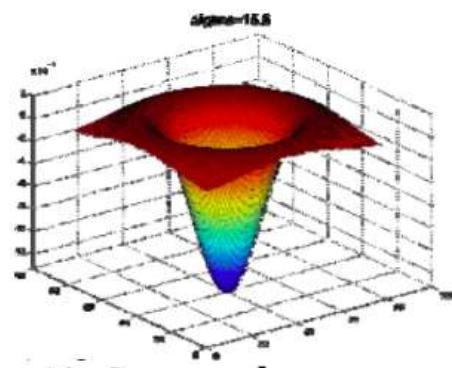
Laplaciano del Gaussiano

- Ejemplo



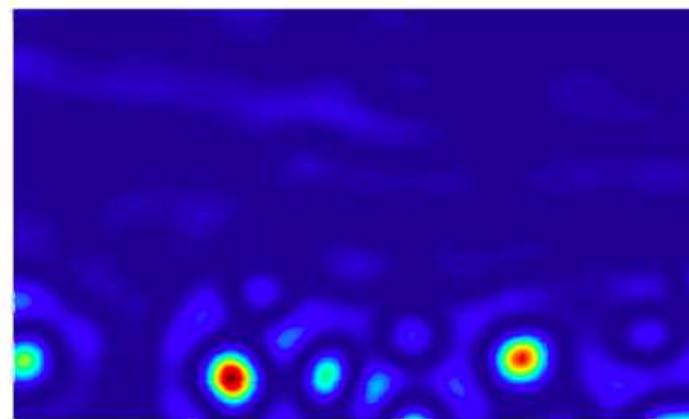
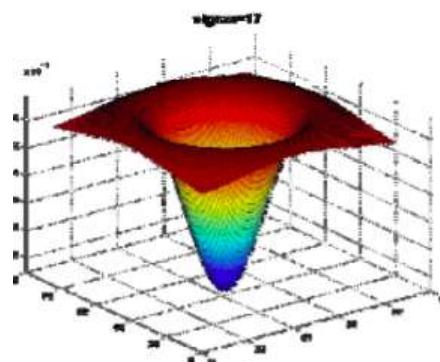
Laplaciano del Gaussiano

- Ejemplo



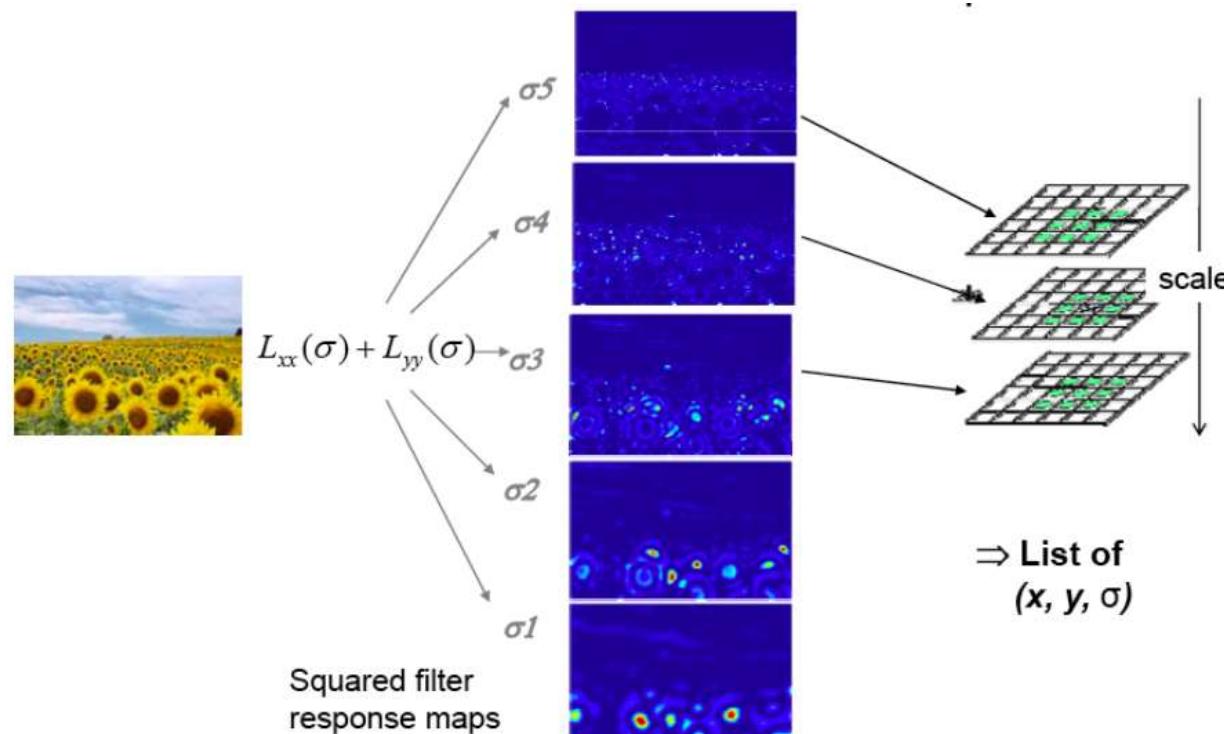
Laplaciano del Gaussiano

- Ejemplo



Laplaciano del Gaussiano

- Puntos de interés son máximos locales en posición y escala



Laplaciano del Gaussiano

- Puntos de interés son máximos locales en posición y escala



Laplaciano del Gaussiano

- Tiempo de procesamiento

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

- Filtros grandes para escalas grandes
- Se puede aproximar al LoG más fácil

Laplaciano del Gaussiano

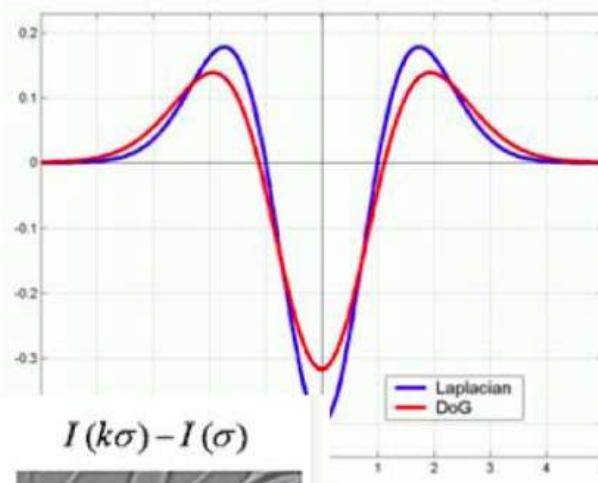
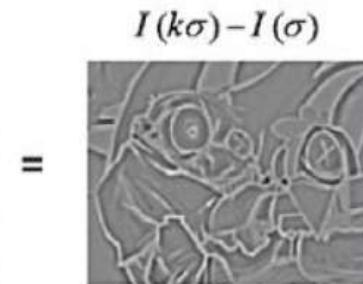
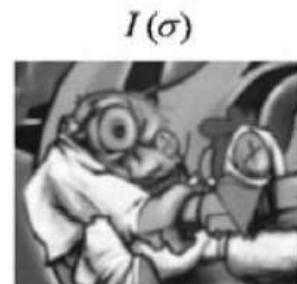
- Aproximación

$$L = \sigma^2 (G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma))$$

(Laplacian)

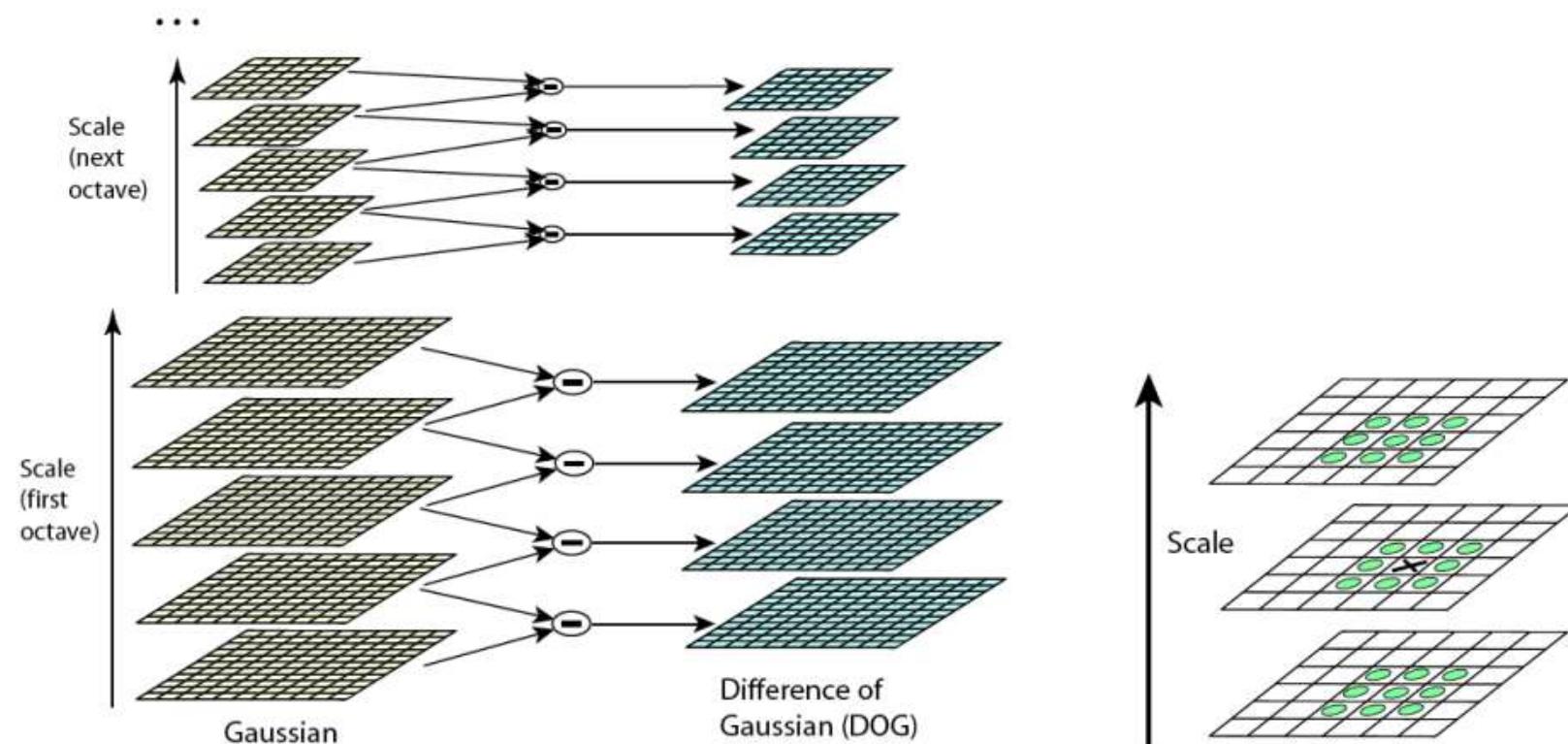
$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)



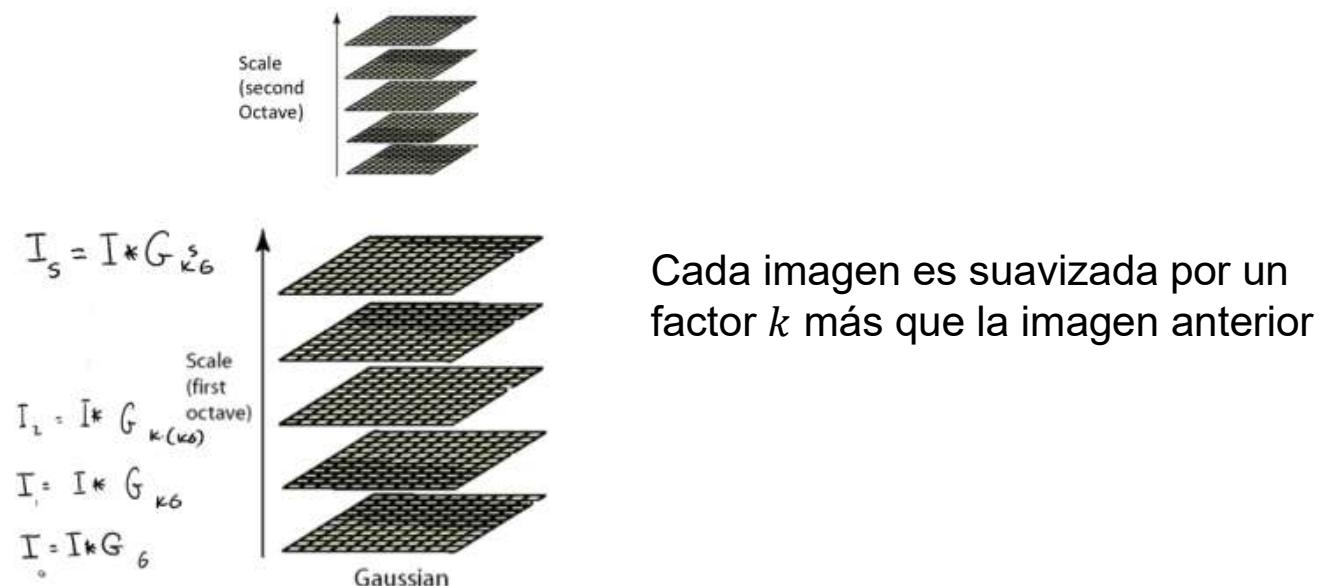
Diferencias de Gaussiano

- Aproximación



Diferencias de Gaussiano

- Se computa una pirámide de Gaussianos

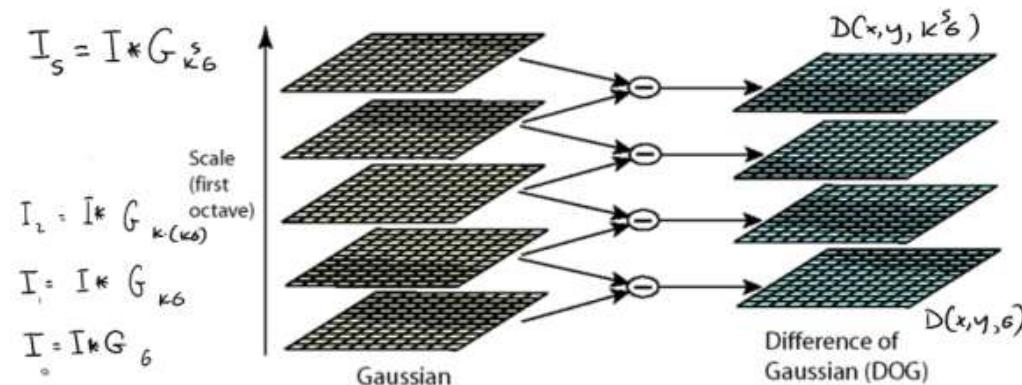


Diferencias de Gaussiano

- Se computa una pirámide de Gaussianos
- Computar diferencia de Gaussianos

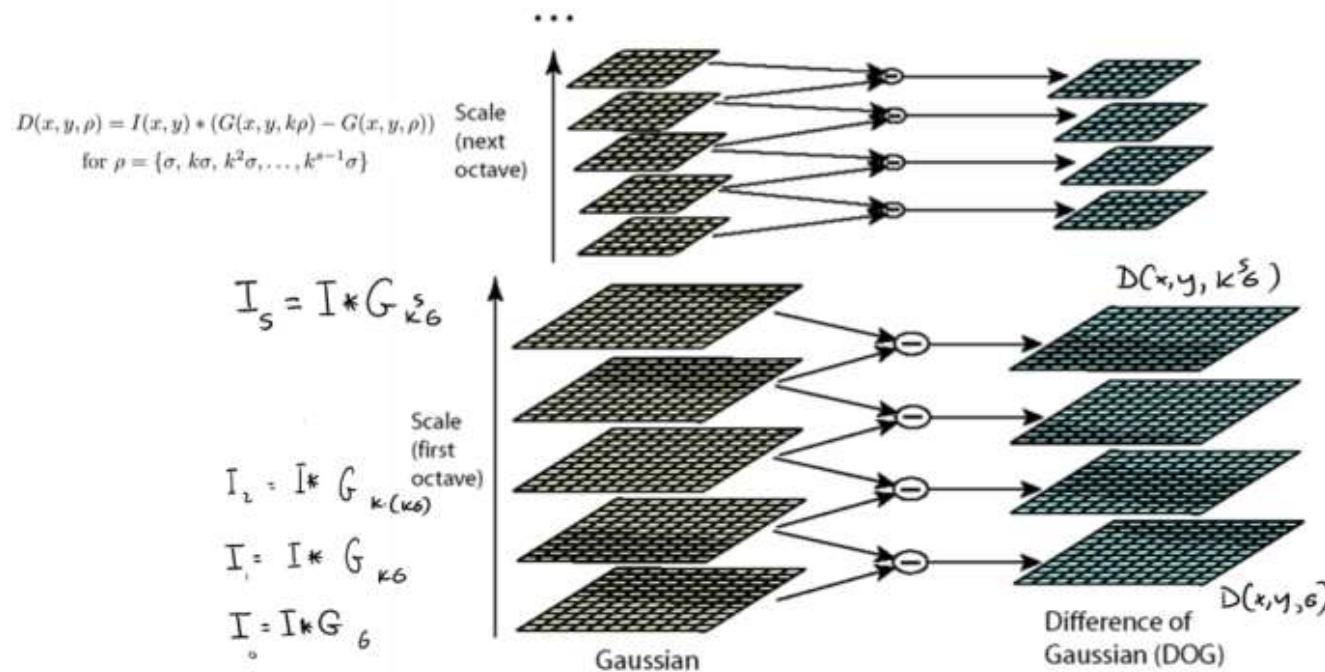
$$D(x, y, \rho) = I(x, y) * (G(x, y, k\rho) - G(x, y, \rho))$$

$$\text{for } \rho = \{\sigma, k\sigma, k^2\sigma, \dots, k^{s-1}\sigma\}, \quad k = 2^{1/s}$$



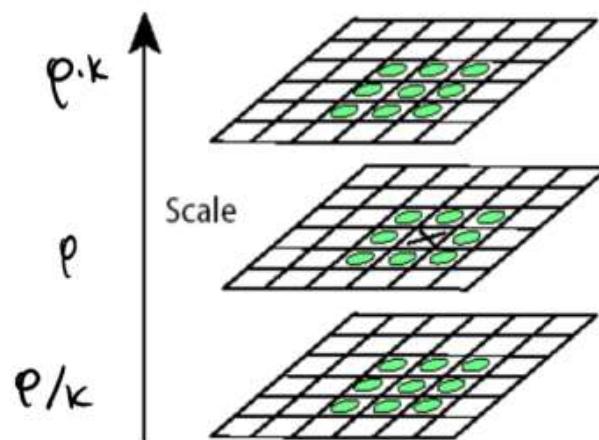
Diferencias de Gaussiano

- Se computa una pirámide de Gaussianos
- Computar diferencia de Gaussianos – en cada escala



Diferencias de Gaussiano

- Se computa una pirámide de Gaussianos
- Computar diferencia de Gaussianos – en cada escala
- Encontrar el máximo local en escala



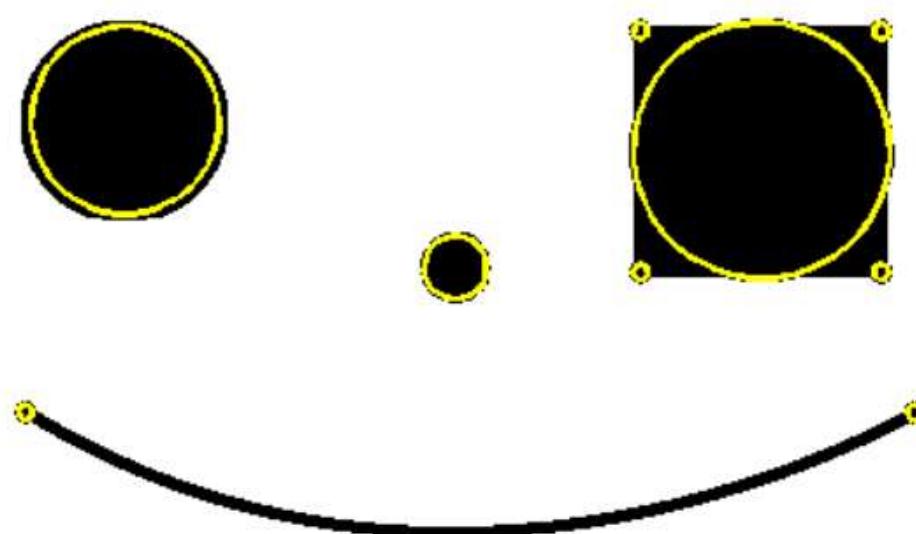
Diferencias de Gaussiano

- Ejemplo



Diferencias de Gaussiano

- Ejemplo – keypoints detectados



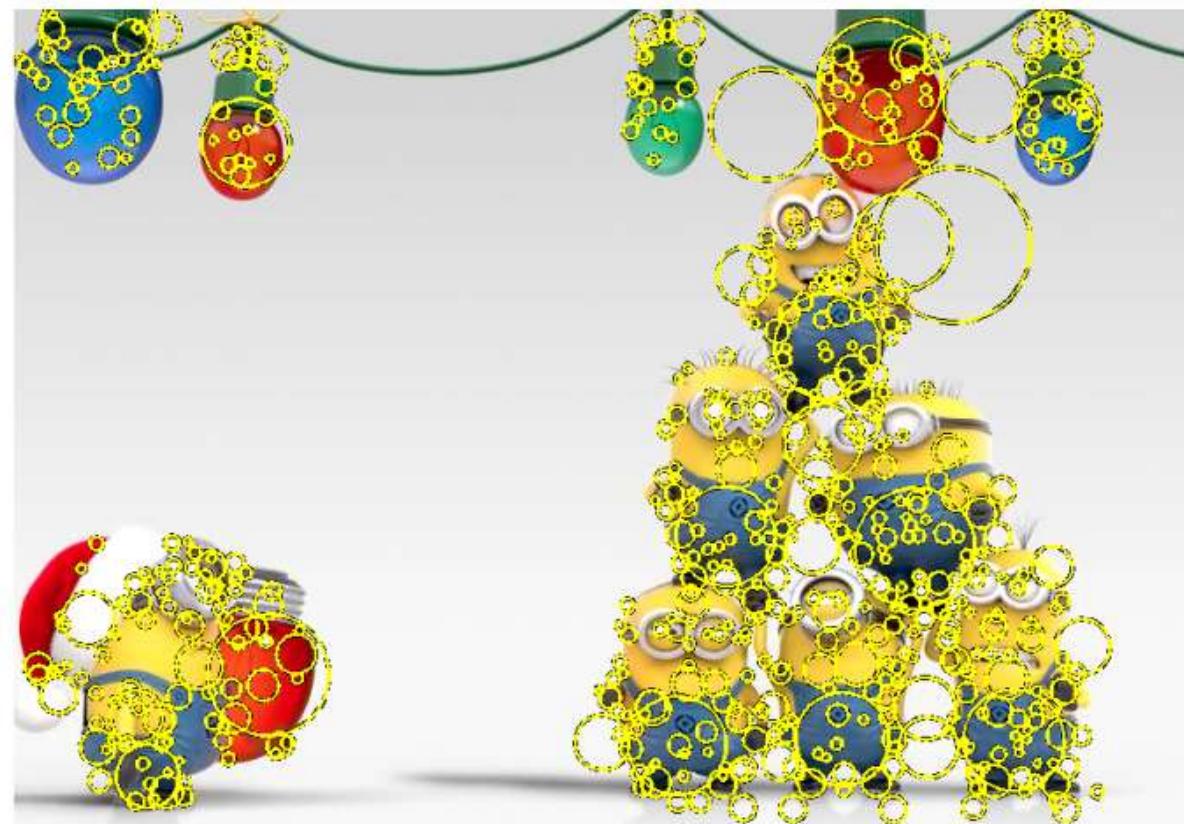
Diferencias de Gaussiano

- Ejemplo



Diferencias de Gaussiano

- Ejemplo – keypoints detectados



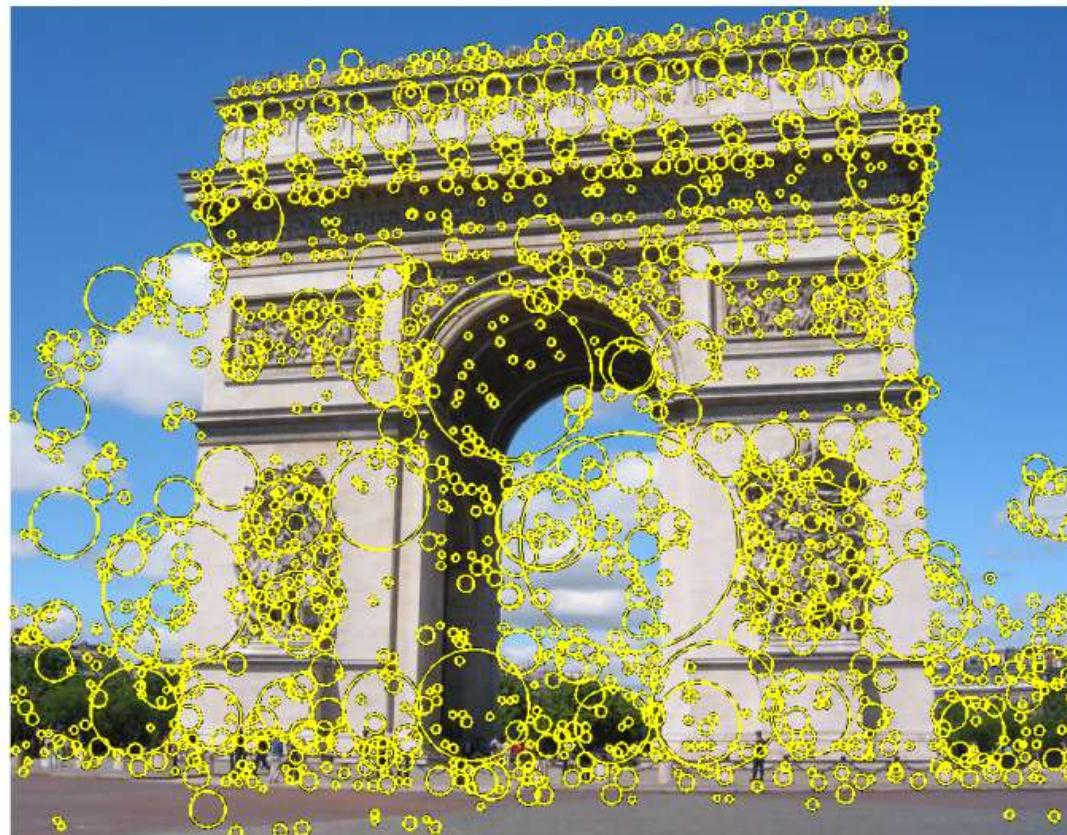
Diferencias de Gaussiano

- Ejemplo



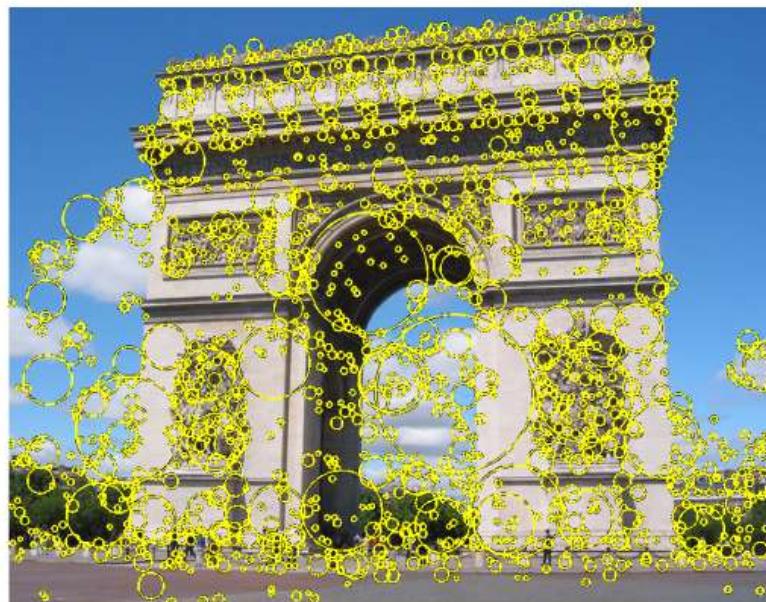
Diferencias de Gaussiano

- Ejemplo



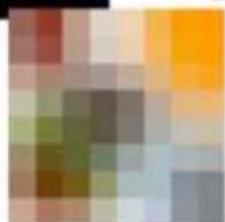
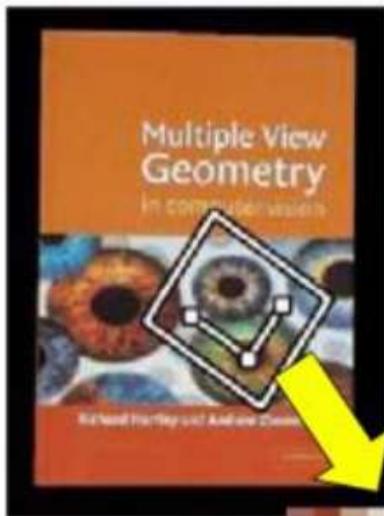
Diferencias de Gaussiano

- Ejemplo



Descriptores Locales

Invarianzas



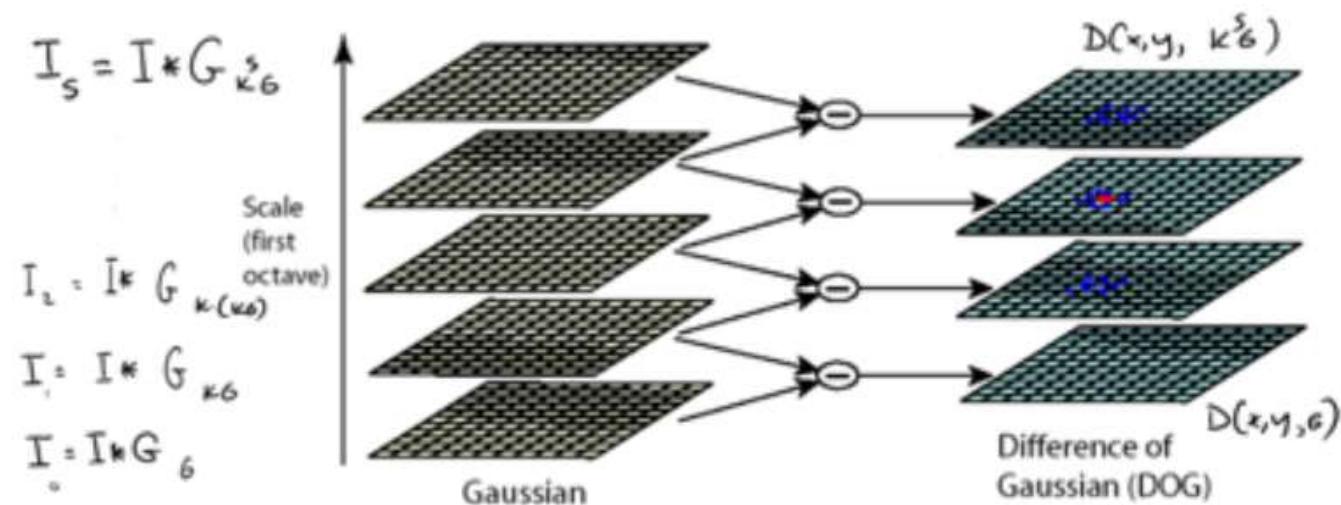
e.g. scale,
translation,
rotation

Invarianzas



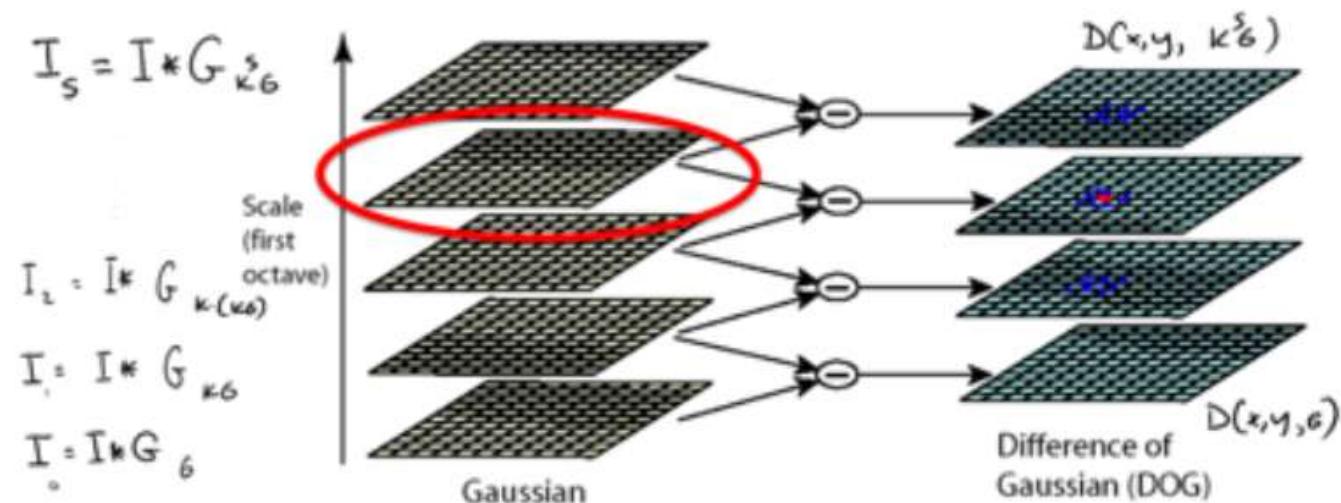
SIFT

- Scale Invariant Feature Transform
- El detector de keypoints entrega información de la escala



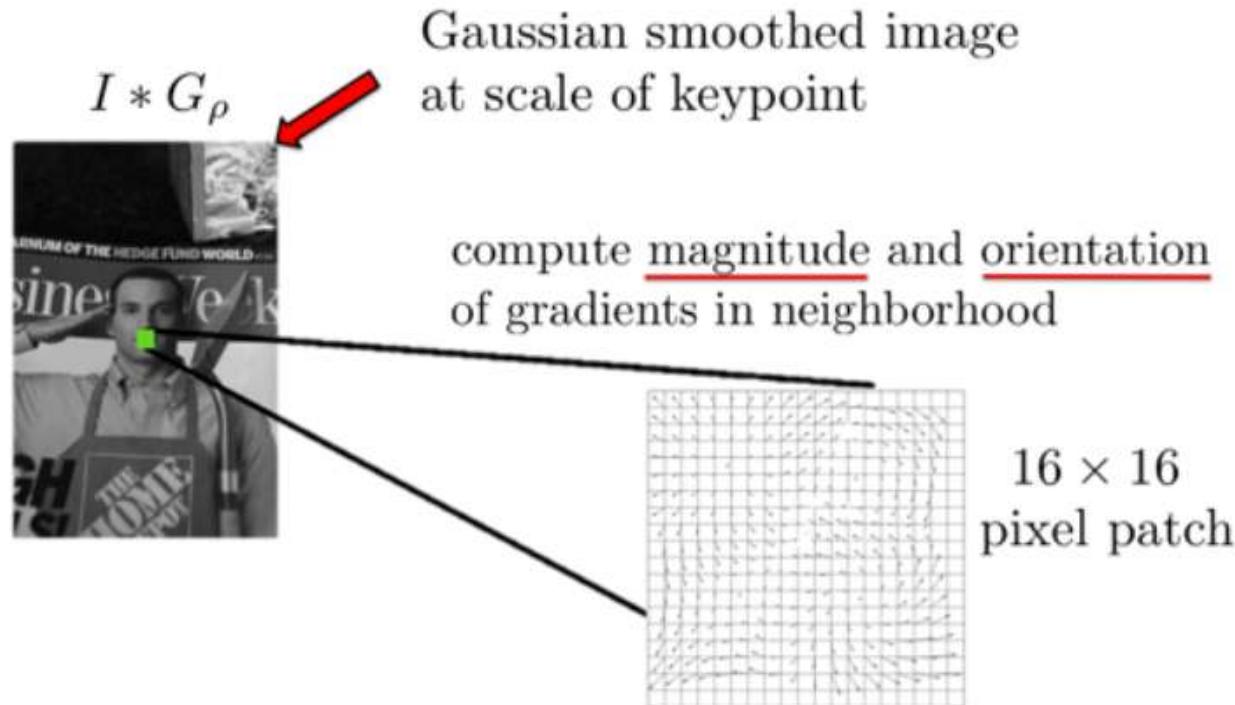
SIFT

- Para cada keypoint, tomamos la imagen convolucionada en la correspondiente escala



SIFT

- Computar la magnitud y orientación del gradiente alrededor de cada keypoint



SIFT

- Computar la magnitud y orientación del gradiente alrededor de cada keypoint

magnitude of gradient:

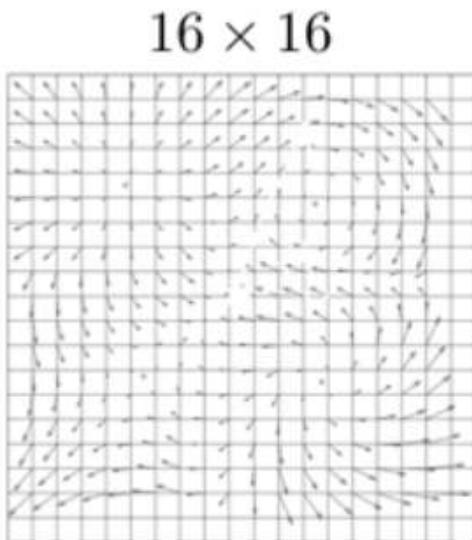
$$|\nabla I(x, y)| = \sqrt{\left(\frac{\partial(I(x, y) * G_\rho)}{\partial x}\right)^2 + \left(\frac{\partial(I(x, y) * G_\rho)}{\partial y}\right)^2}$$

gradient orientation:

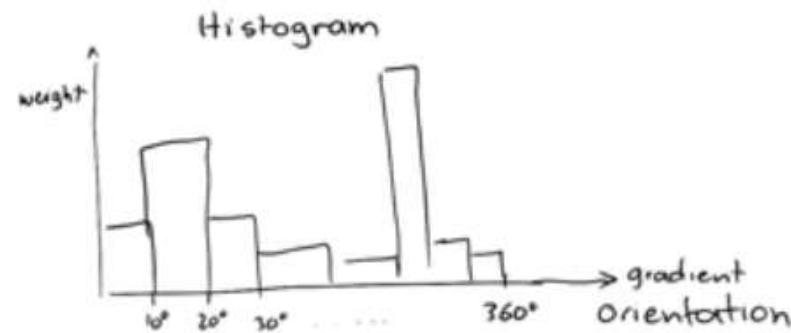
$$\theta(x, y) = \arctan\left(\frac{\partial I * G_\rho}{\partial y} / \frac{\partial I * G_\rho}{\partial x}\right)$$

SIFT

- Computar la orientación dominante de cada keypoint

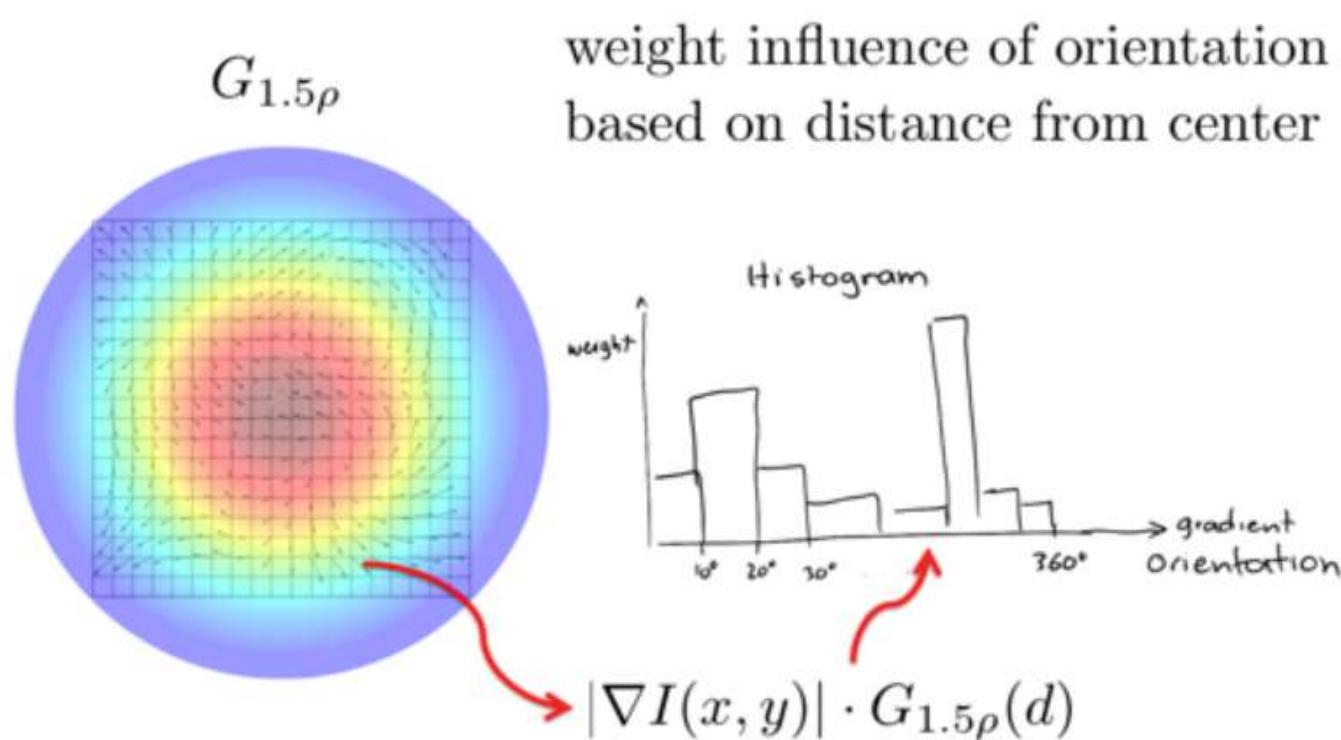


compute histograms of orientations
by orientation increments of 10°



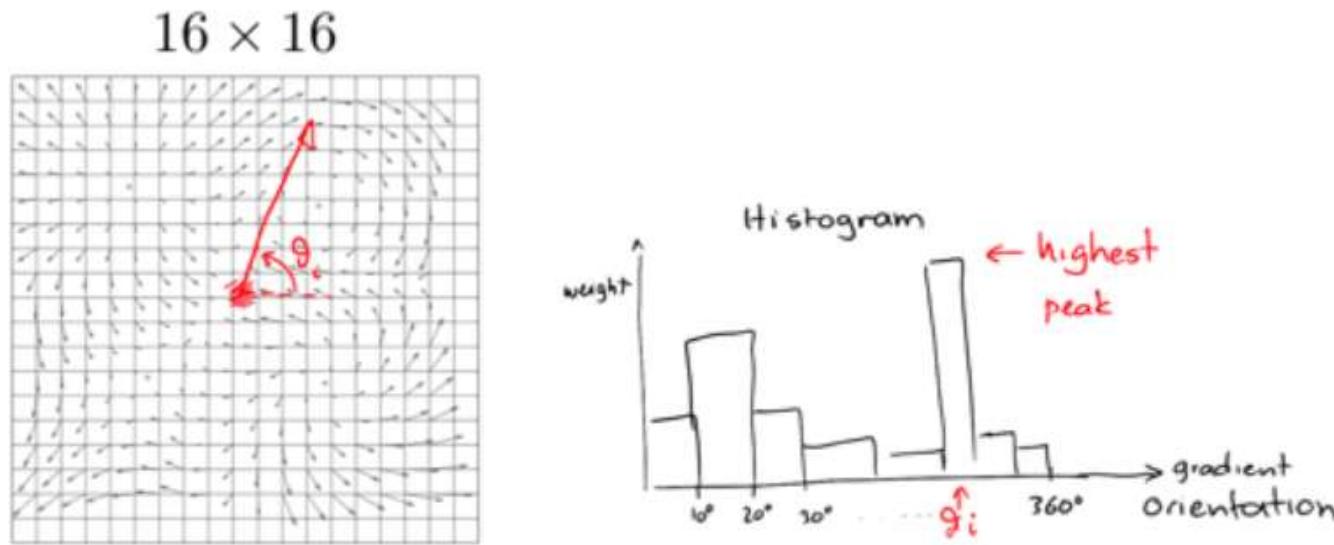
SIFT

- Computar la orientación dominante de cada keypoint
- Orientaciones cercanas al centro deberían contribuir más



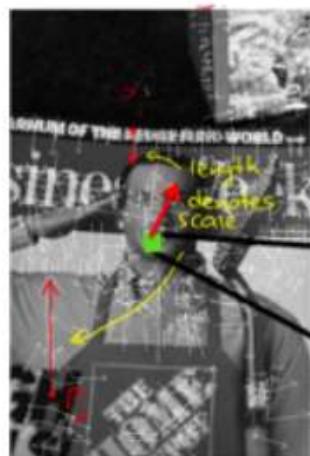
SIFT

- Computar la orientación dominante de cada keypoint
- Orientaciones cercanas al centro deberían contribuir más
- Orientación máxima en el histograma

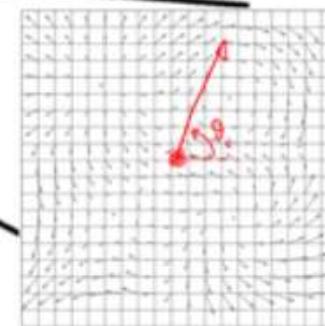


SIFT

- Computar la orientación dominante de cada keypoint



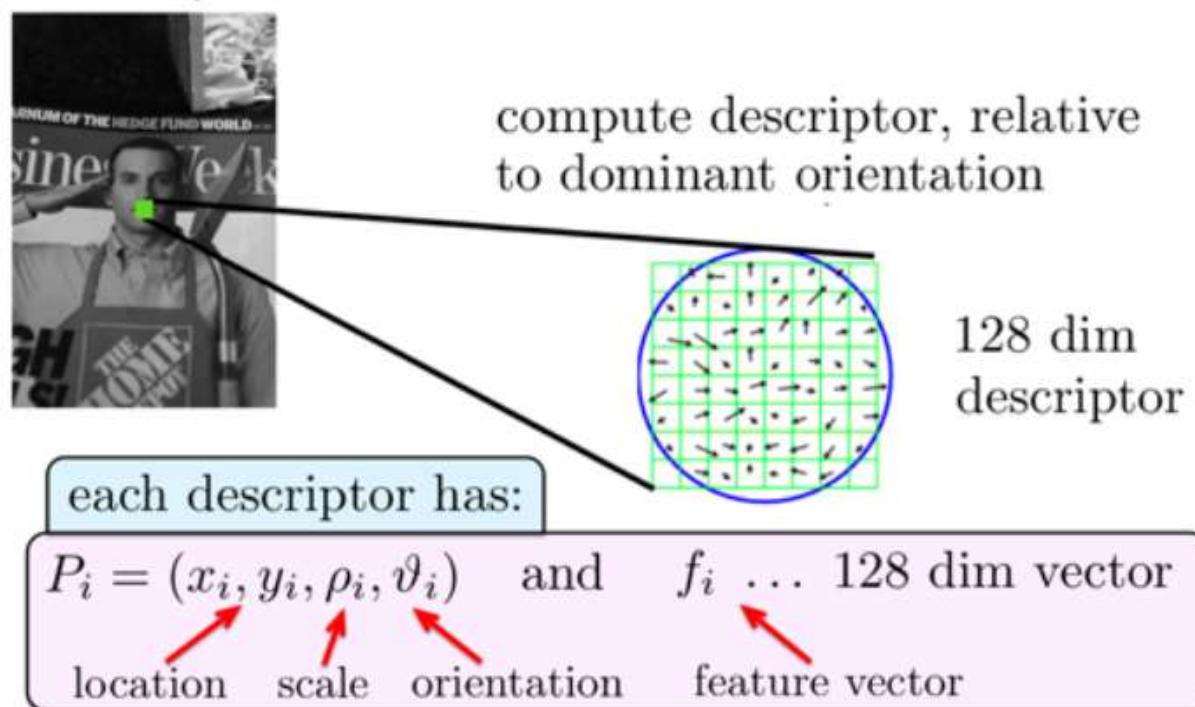
compute magnitude and orientation
of gradients in neighborhood



16×16
pixel patch

SIFT Descriptor

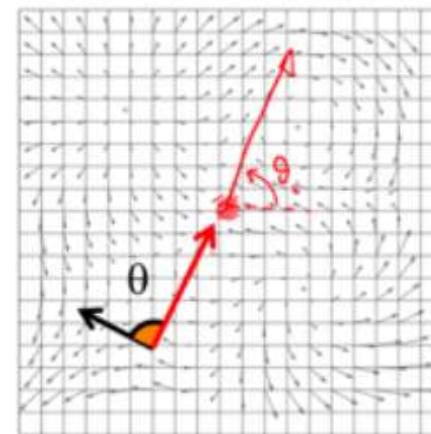
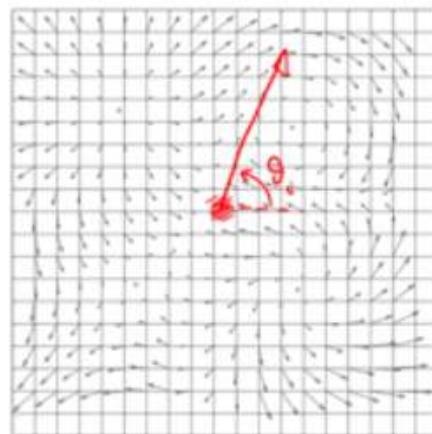
- Computar un descriptor 128-dimensional: grid de 4x4 .
Cada celda es un histograma de 8 bins con orientaciones



SIFT Descriptor

- Computar orientaciones relativas a la orientación dominante

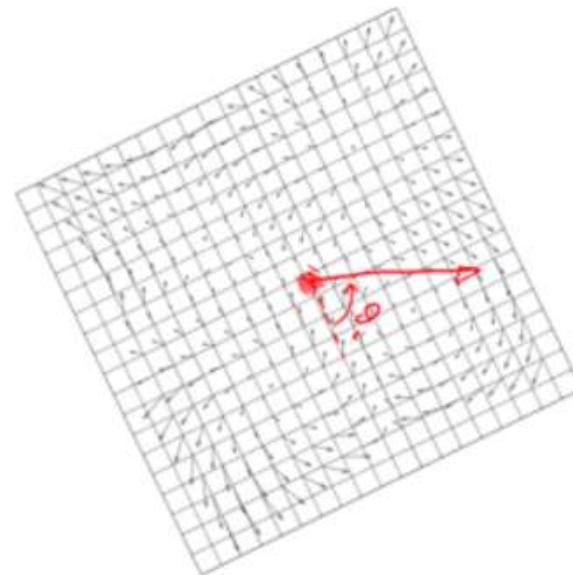
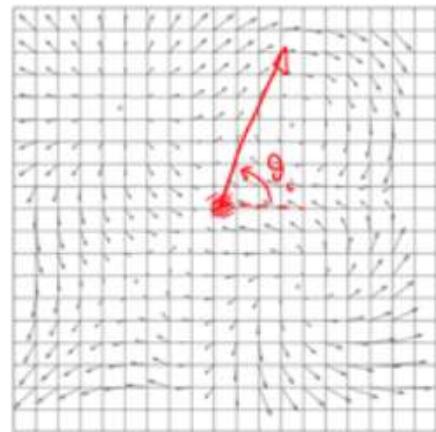
16×16 patch
centered in (x_i, y_i)



SIFT Descriptor

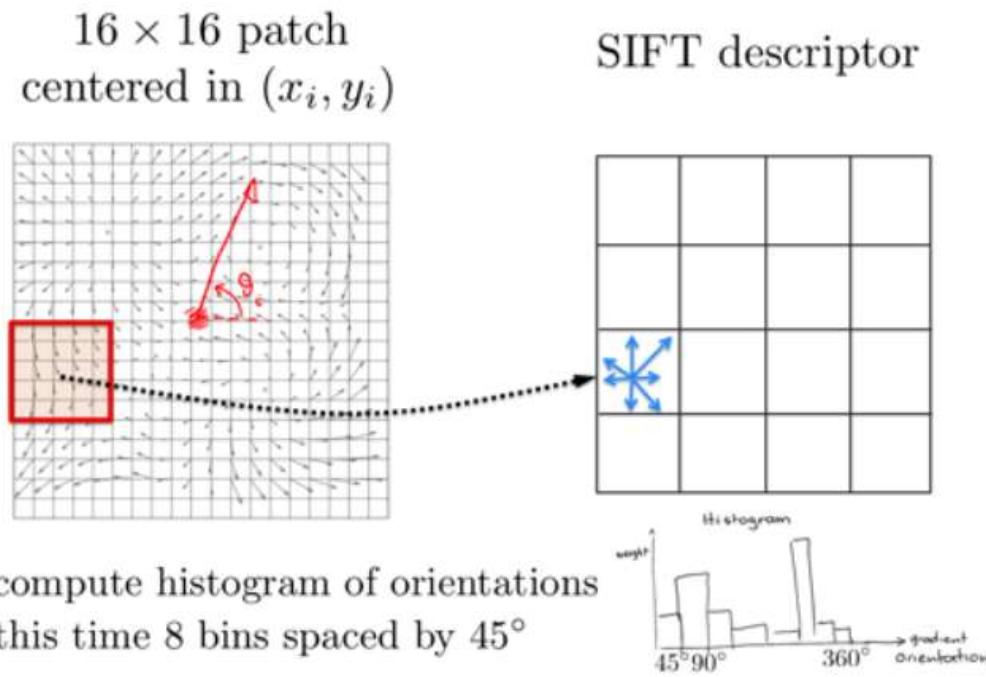
- Computar orientaciones relativas a la orientación dominante

16×16 patch
centered in (x_i, y_i)



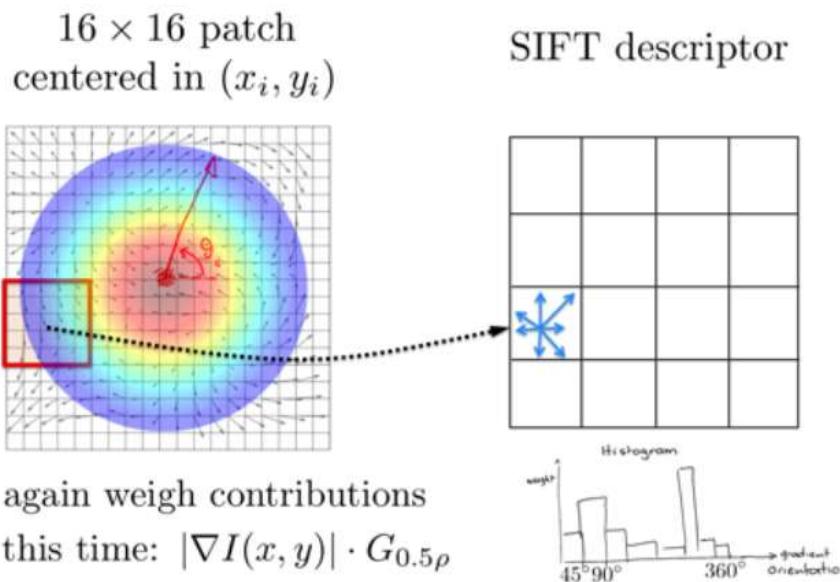
SIFT Descriptor

- Computar orientaciones relativas a la orientación dominante
- Formar una grid de 4×4 . Para cada celda se computa un histograma de orientaciones de 8 bins



SIFT Descriptor

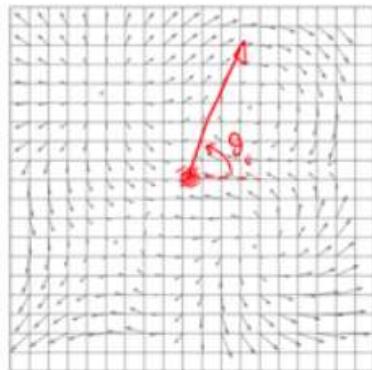
- Computar orientaciones relativas a la orientación dominante
- Formar una grid de 4x4. Para cada celda se computa un histograma de orientaciones de 8 bins



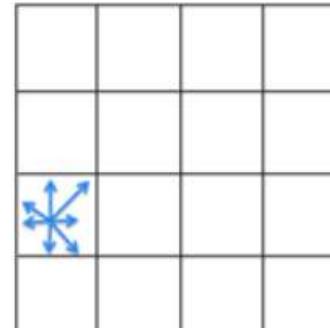
SIFT Descriptor

- Computar orientaciones relativas a la orientación dominante
- Formar una grid de 4×4 . Para cada celda se computa un histograma de orientaciones de 8 bins

16×16 patch
centered in (x_i, y_i)



SIFT descriptor



$$f_i = \begin{array}{c} \downarrow \\ \text{---} | \text{---} | \text{---} | \dots | \text{---} \\ \text{---} | \text{---} | \text{---} | \dots | \text{---} \\ \text{---} | \text{---} | \text{---} | \dots | \text{---} \\ \text{---} | \text{---} | \text{---} | \dots | \text{---} \\ \text{---} | \text{---} | \text{---} | \dots | \text{---} \\ \text{---} | \text{---} | \text{---} | \dots | \text{---} \\ \text{---} | \text{---} | \text{---} | \dots | \text{---} \\ \text{---} | \text{---} | \text{---} | \dots | \text{---} \end{array}$$

Diagram illustrating the SIFT descriptor as a 1D vector f_i composed of 16 bins. A green bracket above the vector indicates its width, labeled n_f . A red bracket below the vector indicates its depth, also labeled n_f .

SIFT Descriptor

- Para reducir efecto de contraste, el descriptor se normaliza
- Posterior proceso:
 - Se recortan valores a 0.2 y se vuelve a normalizar