

Máster Universitario en
Nuevas Tecnologías en Informática

Asignatura “Visión Artificial”

Extracción de características I: Puntos y bordes

Facultad de Informática
Universidad de Murcia
Curso 2018/19

Introducción a las *features*

- Imagen = array matricial de valores de gris/color → poco descriptivo.
- Buscamos una representación más informativa o significativa, adecuada, p.e, para *matching* entre imágenes (continuo, o “wide baseline”). *Útil* en tareas como:
 - Profundidad a partir de dos (estéreo) o más vistas (reconstrucción 3D).
 - *Image retrieval* (obtener imágenes que contienen un modelo).
 - Localización (p.e. de un robot móvil en un determinado entorno).
 - Calibrar cámaras (*matching* con plantillas de calibración).
 - Panoramas.
 - ...

Introducción a las *features*

- Ejemplo de extracción de features: puntos y líneas



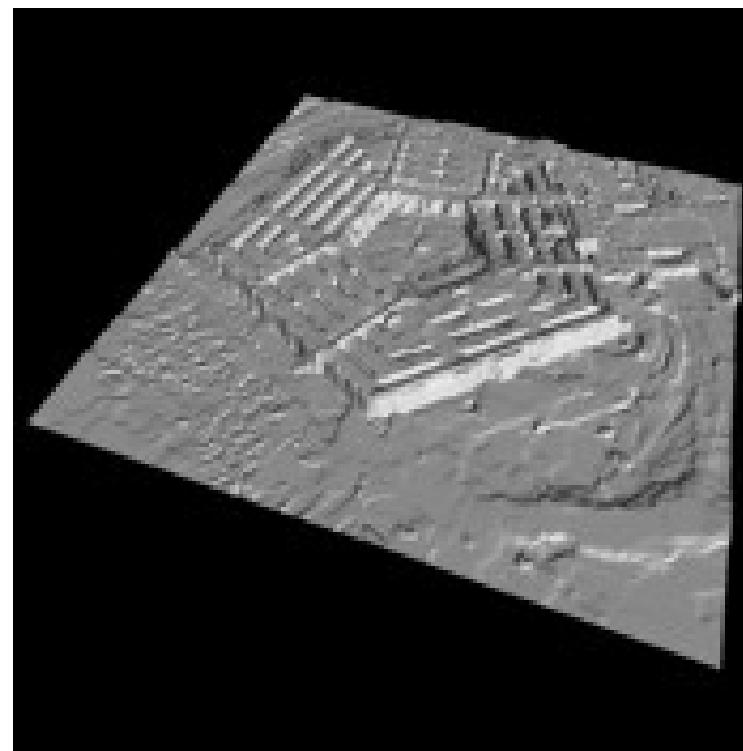
Introducción a las *features*

- Ejemplo de matching entre *features*:



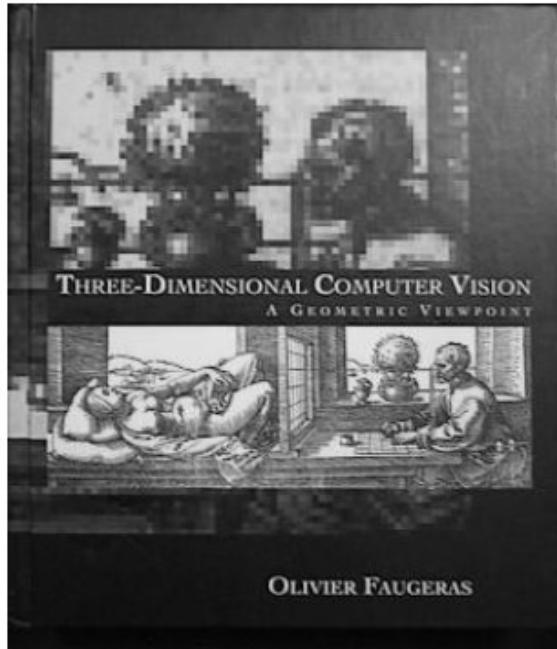
Ejemplo de reconstrucción 3D

- Resultado:
 - Usando técnicas de geometría proyectiva, y las restricciones de posición de cámaras y escena impuestas por el matching anterior, *stereo denso* (mapa de profundidad):



Ejemplo de reconocimiento de objetos

Queremos
Encontrar esto...



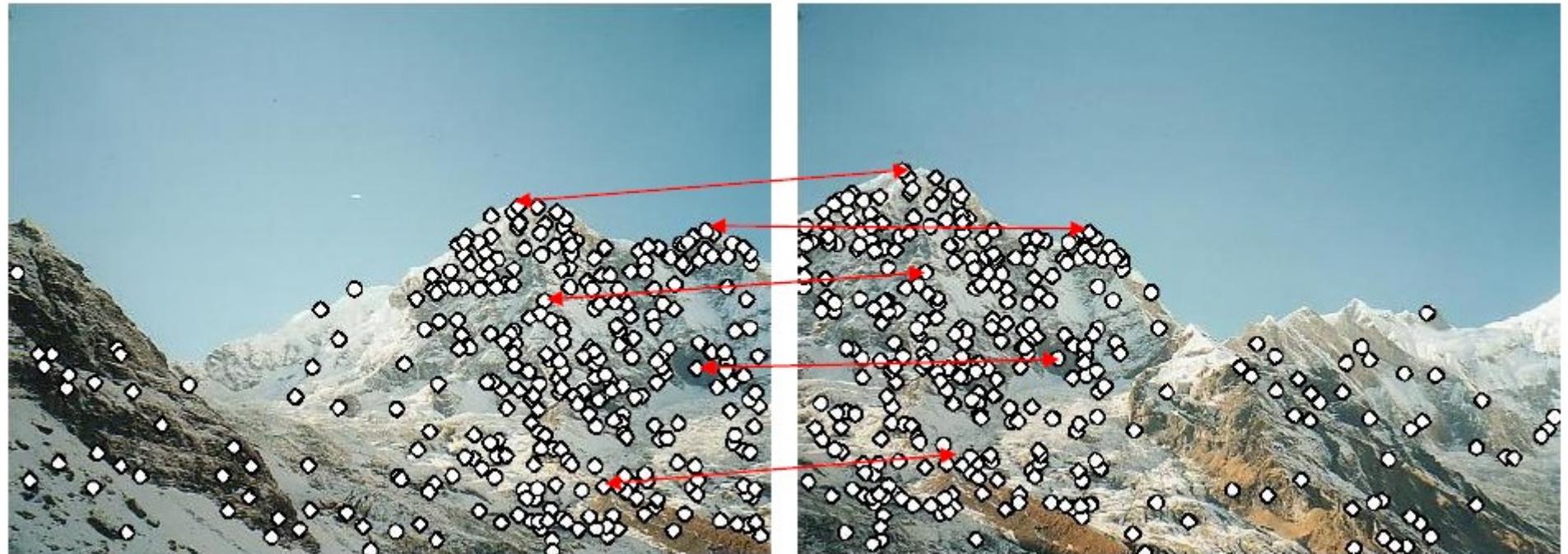
Ejemplo de construcción de panoramas

- Imágenes de partida: dos o más con solapamiento aparente:



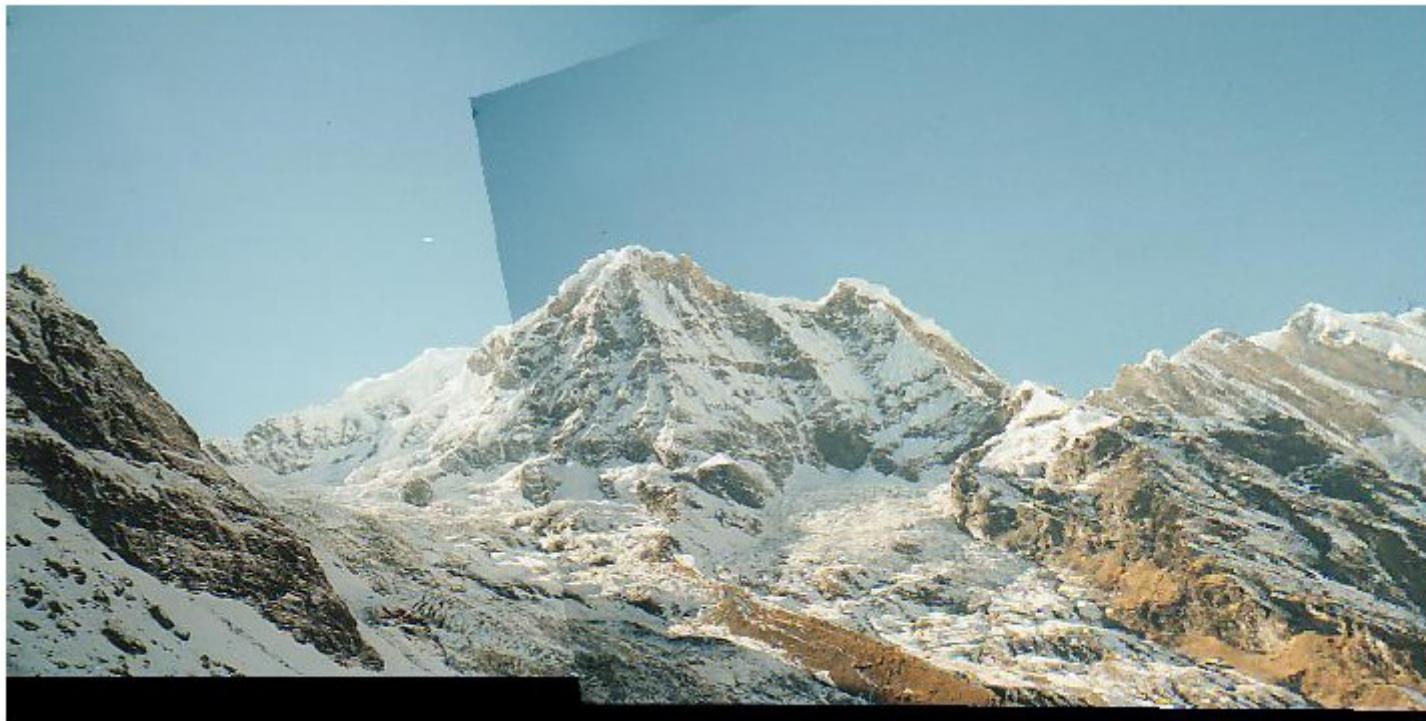
Ejemplo de construcción de panoramas

- Extracción de características, y *matching*:



Ejemplo de construcción de panoramas

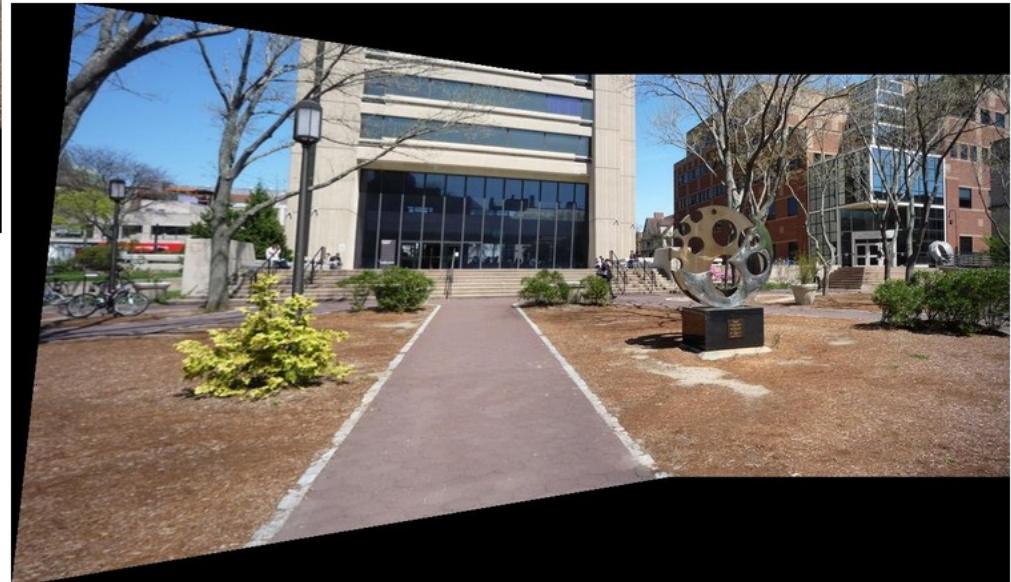
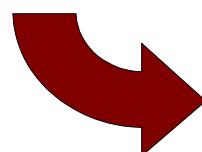
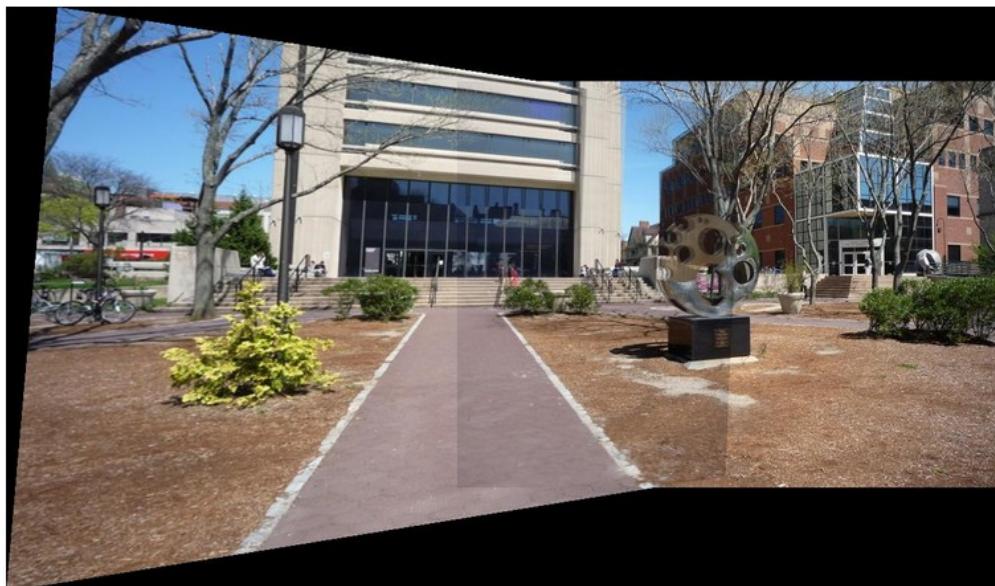
- *Image stitching*: Basado en estimación de la homografía interimagen, usada para el alineamiento.



Posteriormente
habría que
“trabajar”
el *blending*...

Ejemplo de construcción de panoramas (II)

- Ejemplo de *blending* usando *graph cuts*:



Métodos de extracción de características

- Métodos para intentar hacer decisiones más o menos locales en cada punto de la imagen, para decidir si en dicho punto existe una *feature* de un tipo dado o no.
 - Los métodos de extracción se consideran procesamiento de “bajo nivel”.
 - Por supuesto, bastante distintos según el tipo de característica a buscar.
- Tipos de *features*:
 - Puntos aislados, bordes, regiones conectadas, segmentos...
 - Definición ciertamente laxa, dependiente de la aplicación.

Múltiples tipos de *features*

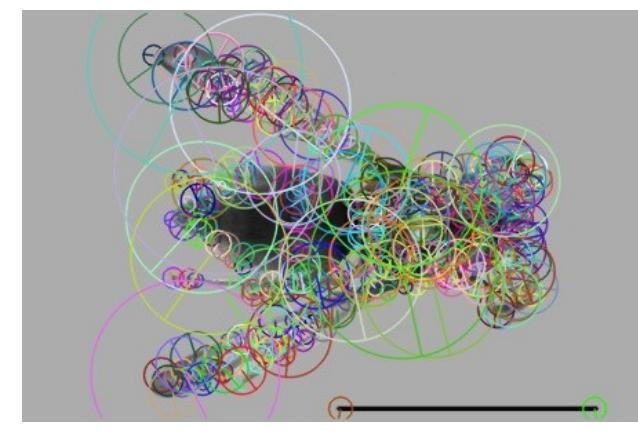
- Puntos con orientación local (a), regiones (b), bordes (c), segmentos (d), *blobs* con escala y orientación locales (SIFT) (e), puntos sin escala (f), ...



(a)



(b)



(e)



(c)



(d)

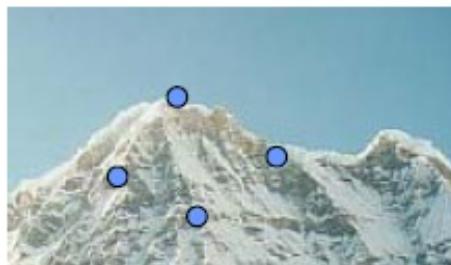


(f)

Características deseables en las features

- **Repetibilidad:**

- Las características deben ser detectadas en ambas imágenes (¡en otro caso, no hay posibilidad de *matching*!):



- **Discernibilidad:**

- Necesidad de un buen *descriptor*, para quiar el *matching*:

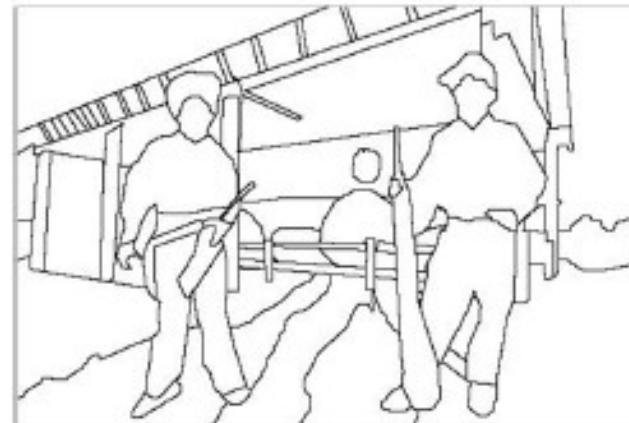
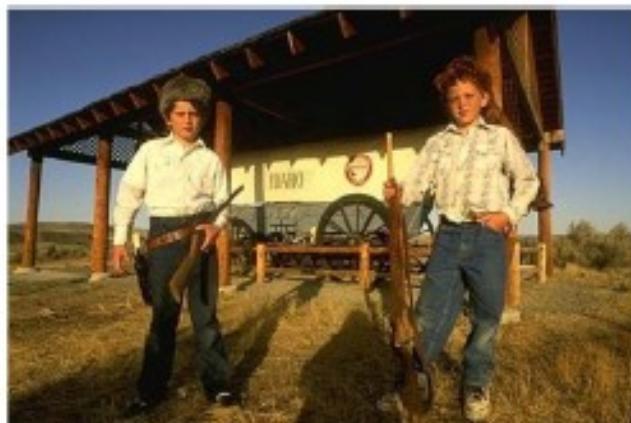
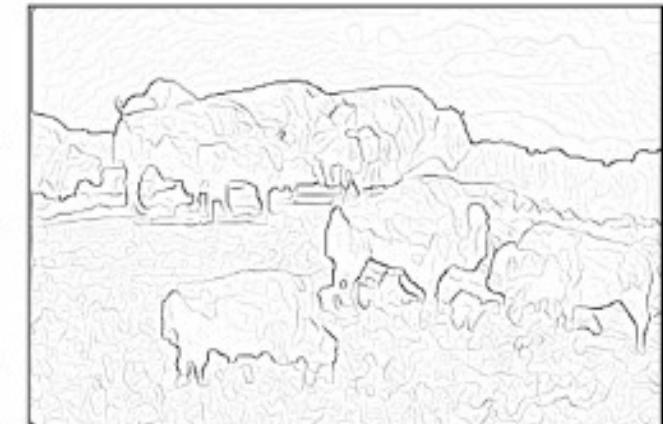
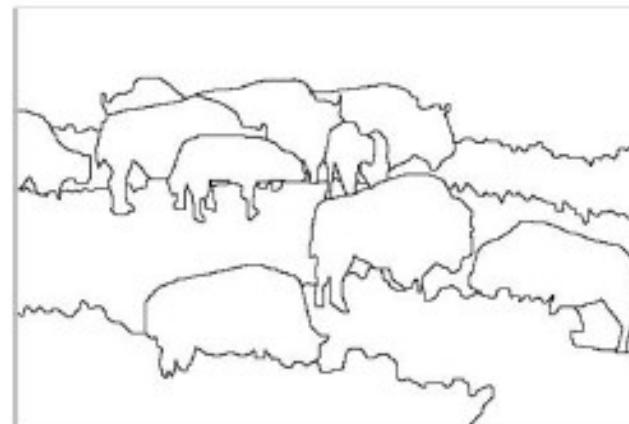


Características deseables en las *features*

- **Estabilidad** en la posición en distintas imágenes (precisión en la localización).
- Buscar **invarianza a transformaciones euclídeas / afines / perspectiva** (en la medida de lo posible)
- Buscar **invarianza a cambios de iluminación.**
- Y por supuesto... **eficiencia!**

Métodos de extracción de bordes

- La percepción natural de los bordes está muy relacionada con los cambios locales de intensidad (gradiente):



Original

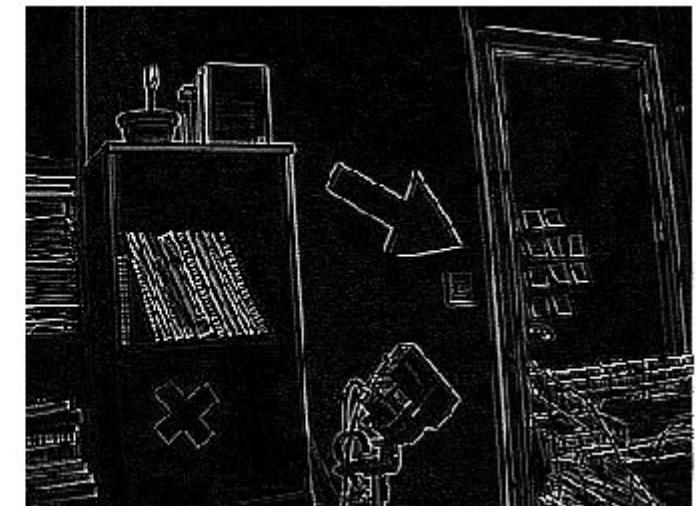
Bordes etiquetados
por humanos

Módulo del
gradiente

Métodos de extracción de bordes

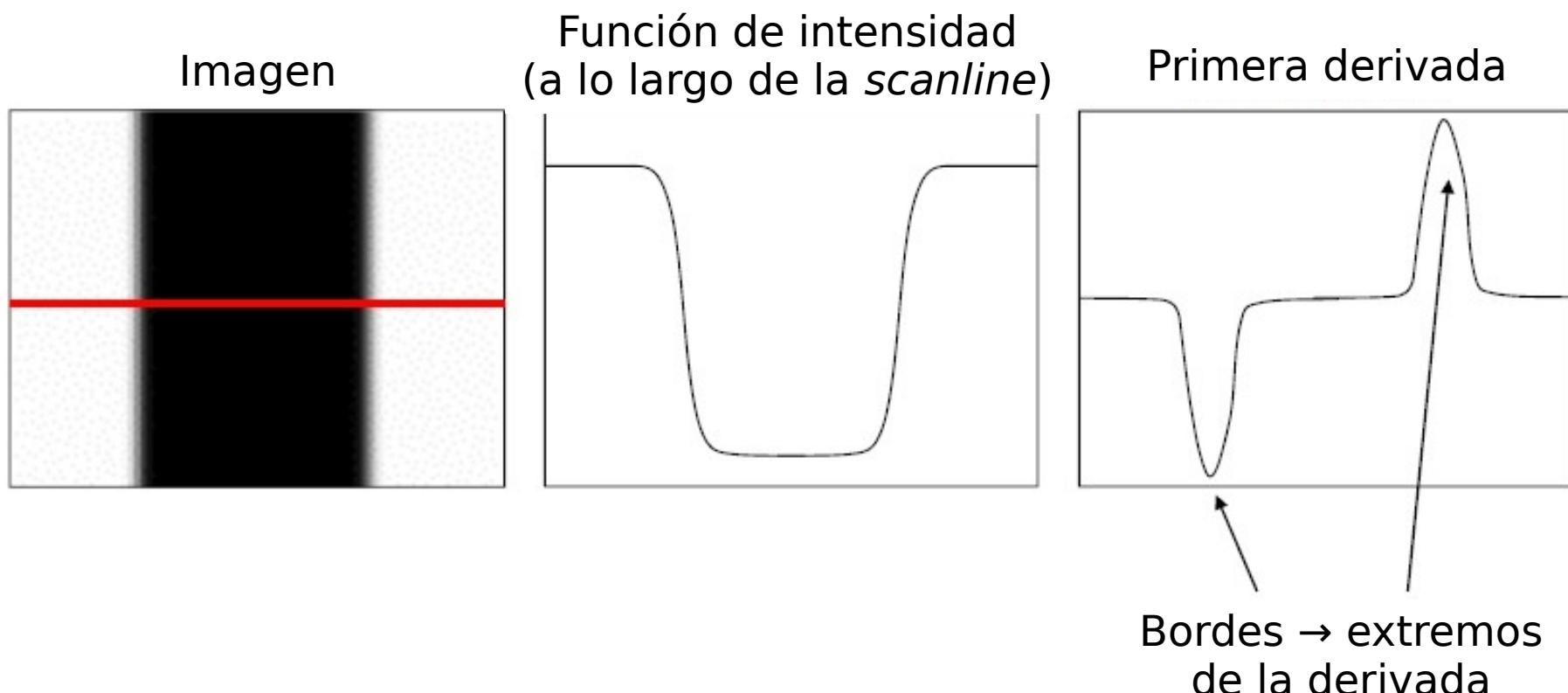
- Aproximación “ingenua”: operador de borde local lineal + umbralizado
- Ejemplo: filtro *hipass* (pasa-alta, deja pasar las “altas frecuencias”):

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



Métodos de extracción de bordes

- Caracterización de bordes (edges):
 - Lugar de cambio rápido en la variación de la función de intensidad:



Estimación local del gradiente

- **Máscaras lineales** más utilizadas para la estimación de las **derivadas direccionales en X e Y**:

- Roberts (simple diferencia finita en X e Y, o bien operador cruzado):

$$\begin{bmatrix} -1 & +1 \end{bmatrix} \quad \begin{bmatrix} -1 \\ +1 \end{bmatrix} \quad \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$

- Prewitt (toma en cuenta entorno algo mayor):

$$\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$$

- Sobel (mejor, ya que en realidad realiza también el equivalente a un presuavizado):

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

Estimación local del gradiente

- Derivadas direccionales en X e Y (se computan con sencillos operadores lineales, como Roberts, Sobel, Prewitt).

Imagen vectorial → $\mathbf{G}(\mathbf{x}) = \nabla I(\mathbf{x}) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)(\mathbf{x}).$

- A partir de ellas, estimación de magnitud y orientación:

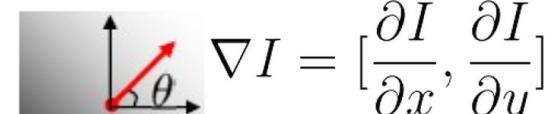
$$\begin{aligned} G_x &= \partial I / \partial x & \mathbf{G} &= \sqrt{G_x^2 + G_y^2} & \Theta &= \text{atan2}(G_y, G_x) \\ G_y &= \partial I / \partial y \end{aligned}$$

- El gradiente es un vector que apunta en la dirección de máxima variación de la imagen, y es perpendicular a los contornos:


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



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



Estimación local del gradiente

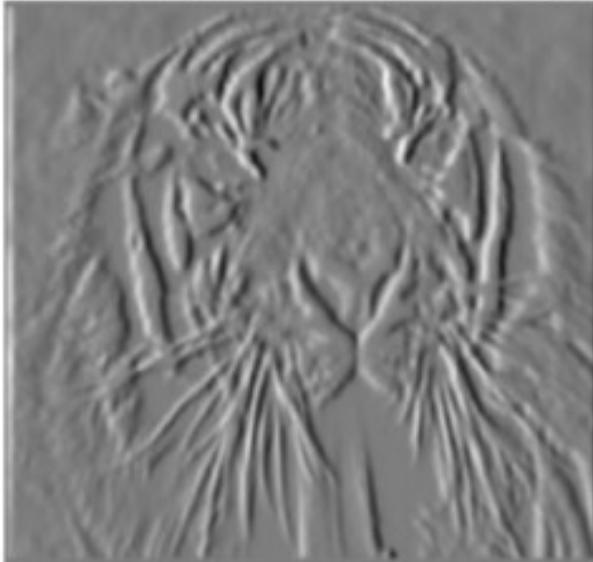
Original



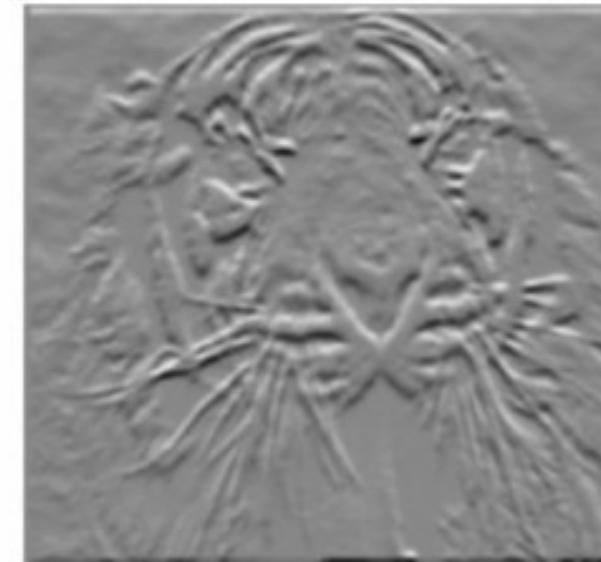
Módulo
del gradiente



Gradiente
en X



Gradiente
en Y



Estimación local del gradiente

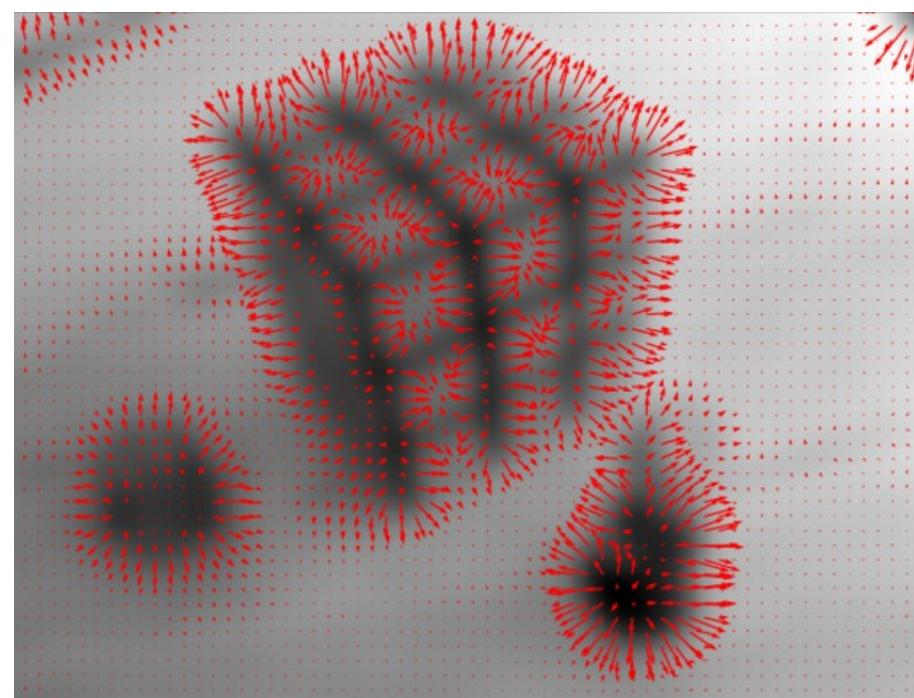
Original



Suavizada
(Filtro gaussiano)

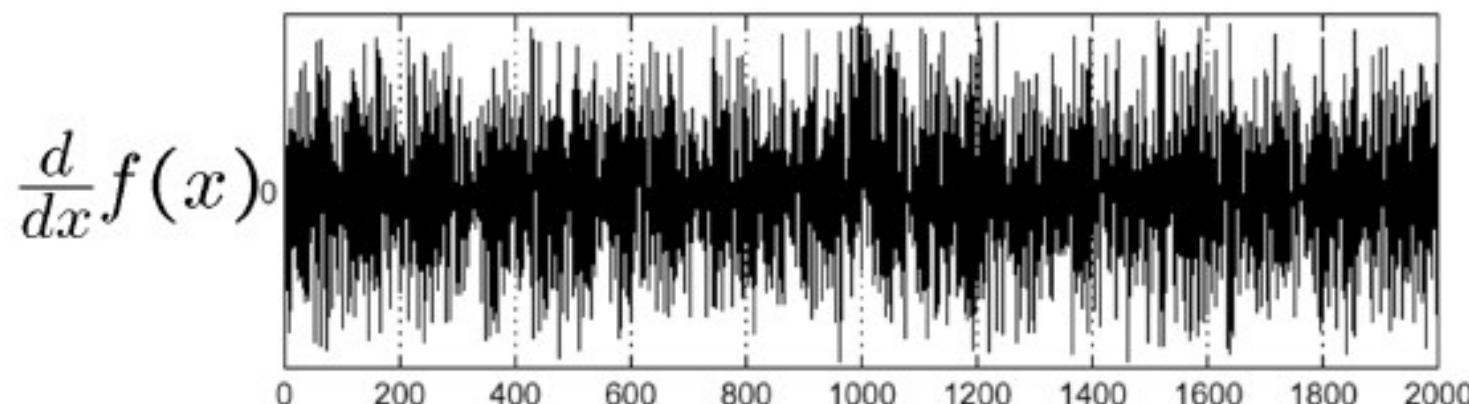
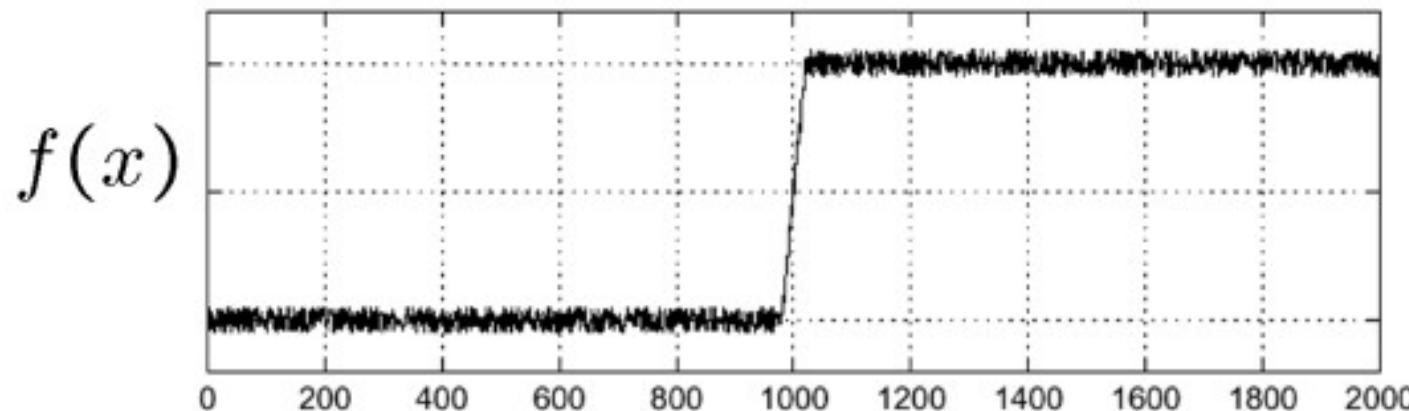


Gradientes
(vectores 2D)



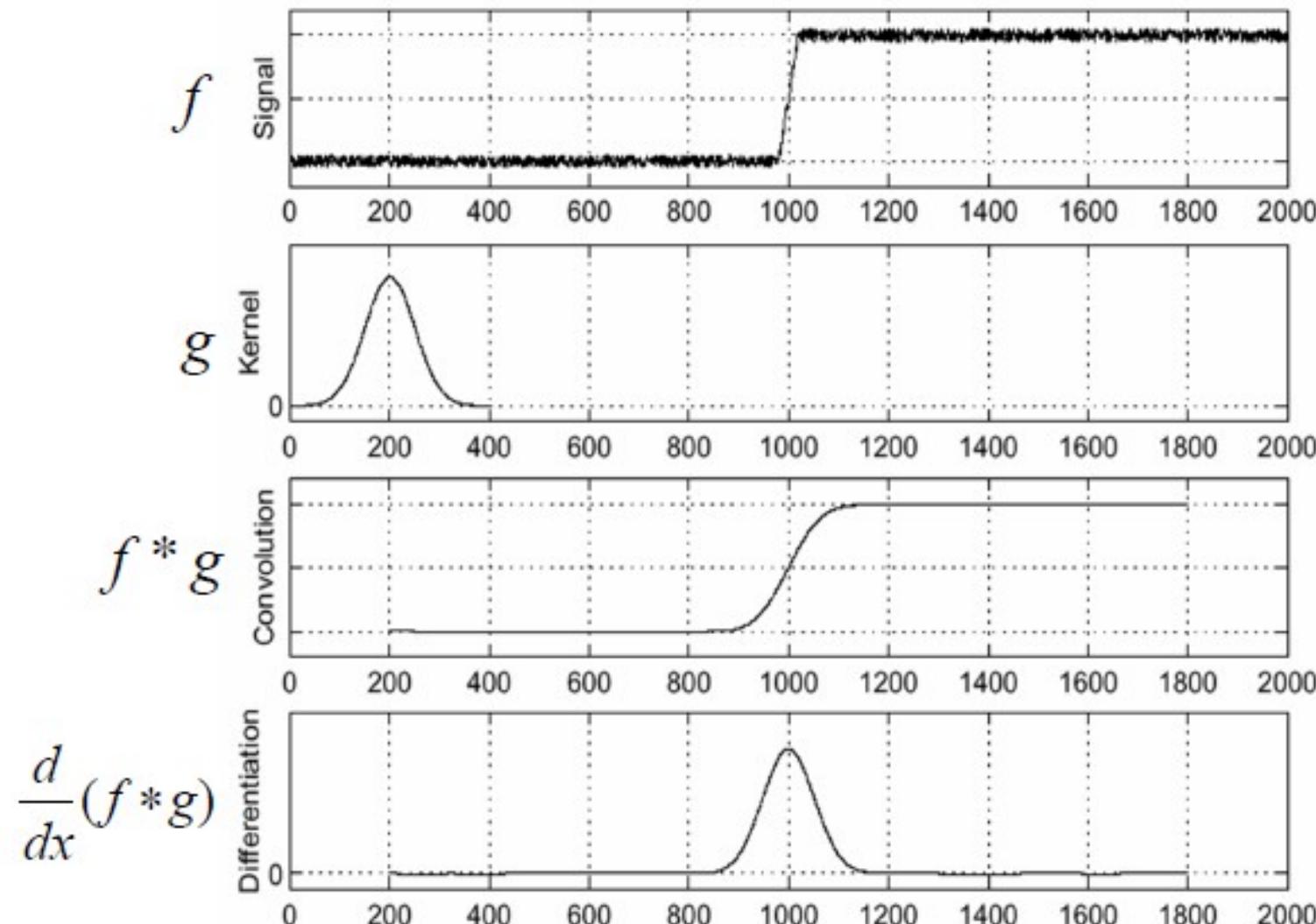
Efectos del ruido en la estimación

- ¡Ojo! La simple diferencia finita ($G_x = I_x - I_{x-1}$) puede responder demasiado al ruido:



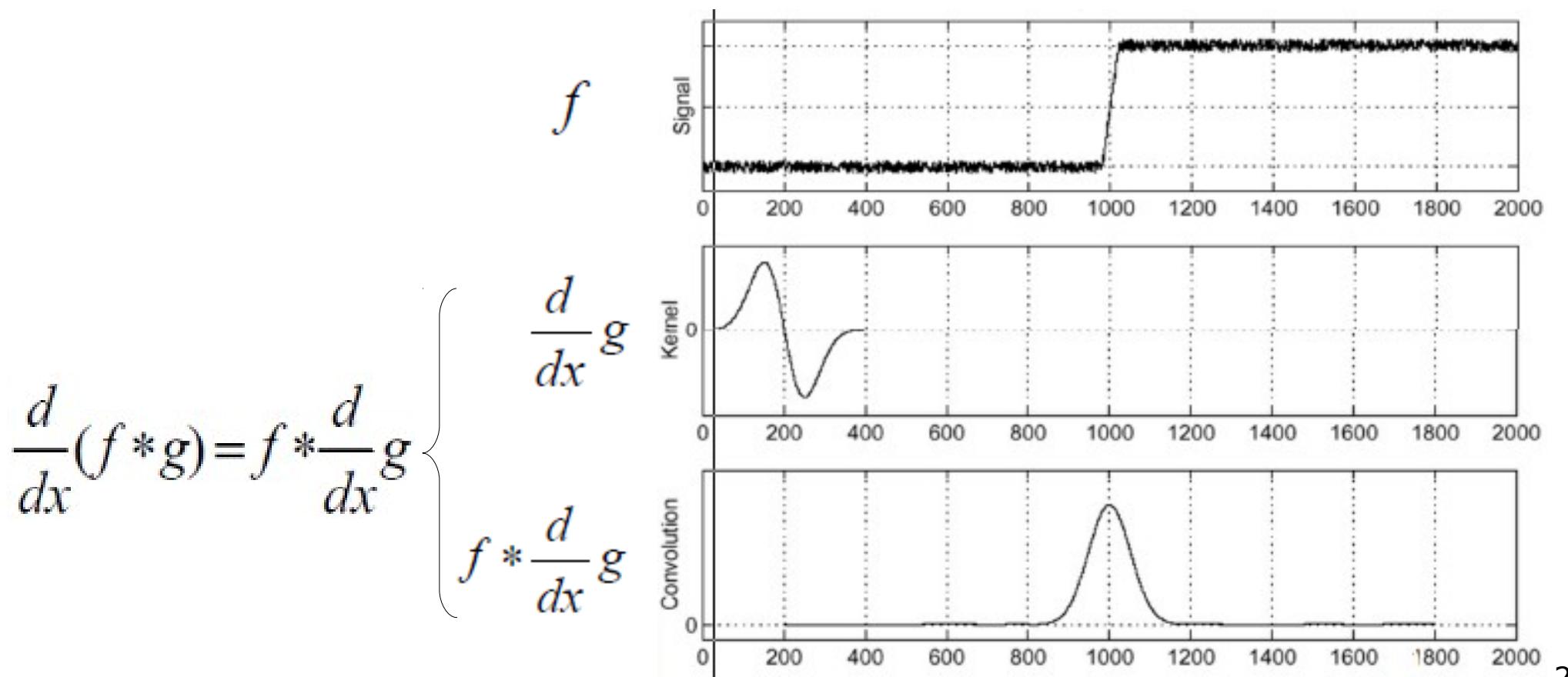
Efectos del ruido en la estimación

- Solución: suavizar primero (p.e. con filtro Gaussiano)

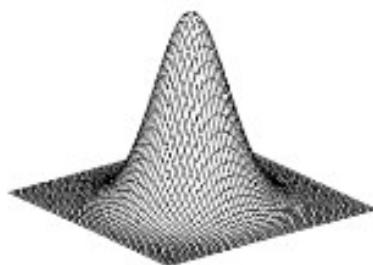


Gradiente suavizado

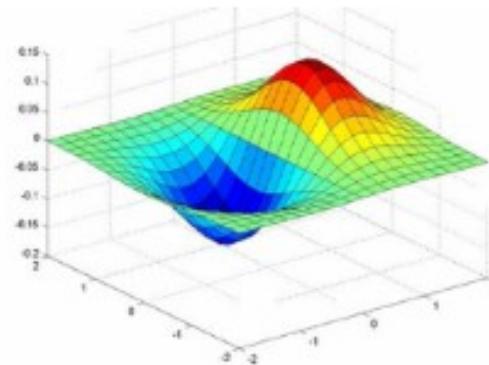
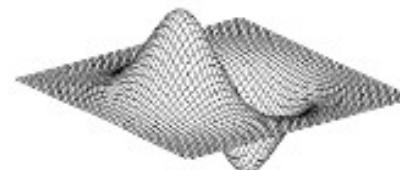
- Puesto que tanto la diferenciación como el suavizado son convoluciones, son asociativas, puede ser más económico combinarlas en una sola operación:



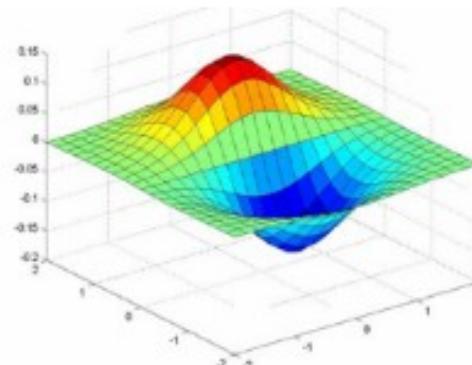
Gradiente suavizado



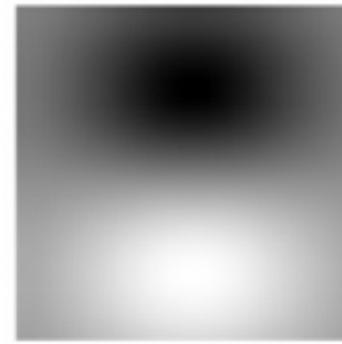
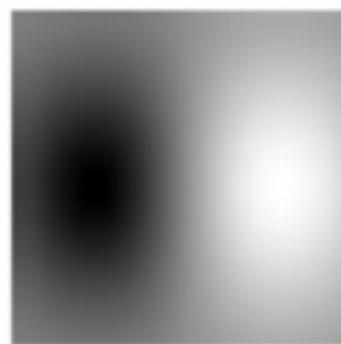
$$\ast [1 \ -1] =$$



x-direction



y-direction



Gradiente suavizado multiescala

- El operador suavizado + gradiente puede hacerse a varias escalas (extensión del operador de Sobel):

$$\mathbf{J}_\sigma(\mathbf{x}) = \nabla[G_\sigma(\mathbf{x}) * I(\mathbf{x})] = [\nabla G_\sigma](\mathbf{x}) * I(\mathbf{x})$$

- Sólo hay que aplicar esta fórmula para generar las máscaras, decidiendo para cada escala σ el tamaño apropiado:

$$\nabla G_\sigma(\mathbf{x}) = \left(\frac{\partial G_\sigma}{\partial x}, \frac{\partial G_\sigma}{\partial y} \right)(\mathbf{x}) = [-x \ -y] \frac{1}{\sigma^3} \exp \left(-\frac{x^2 + y^2}{2\sigma^2} \right)$$

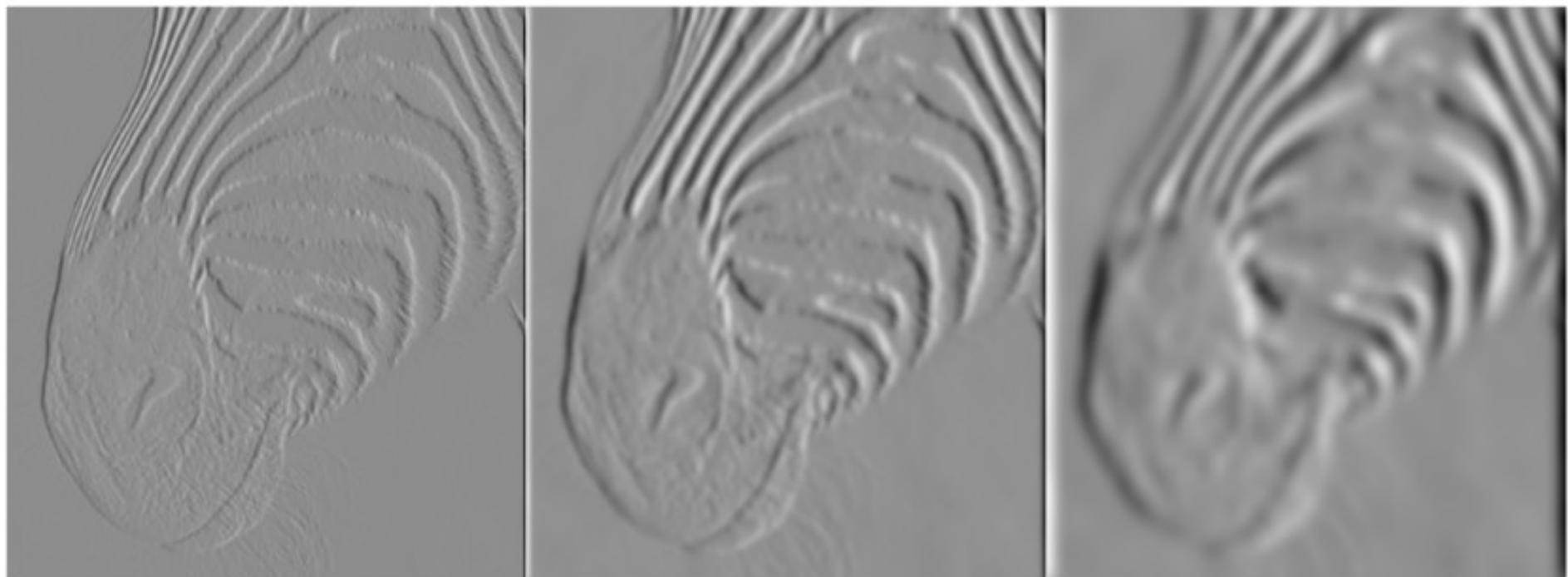


Gradiente suavizado

- Compromiso (*trade-off*) entre localización y suavizado:

Mejor localización

Menos ruido



Radio suavizado
1 píxel

Radio suavizado
3 píxeles

Radio suavizado
7 píxeles

Resultados diferentes estimadores

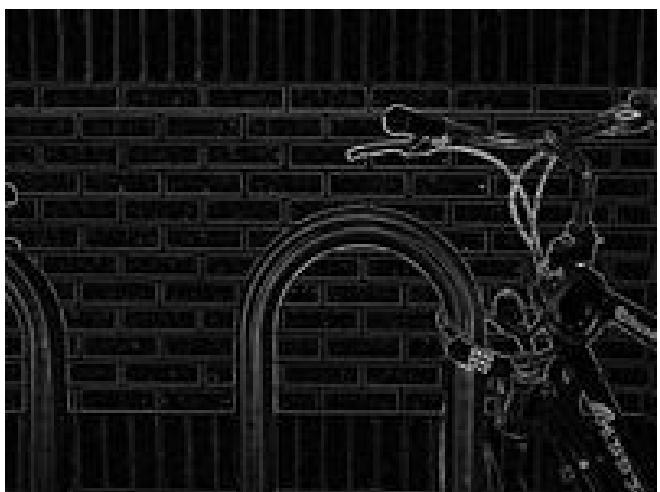
- Comparación de resultados (módulo del gradiente):



Original



Roberts
(cross)



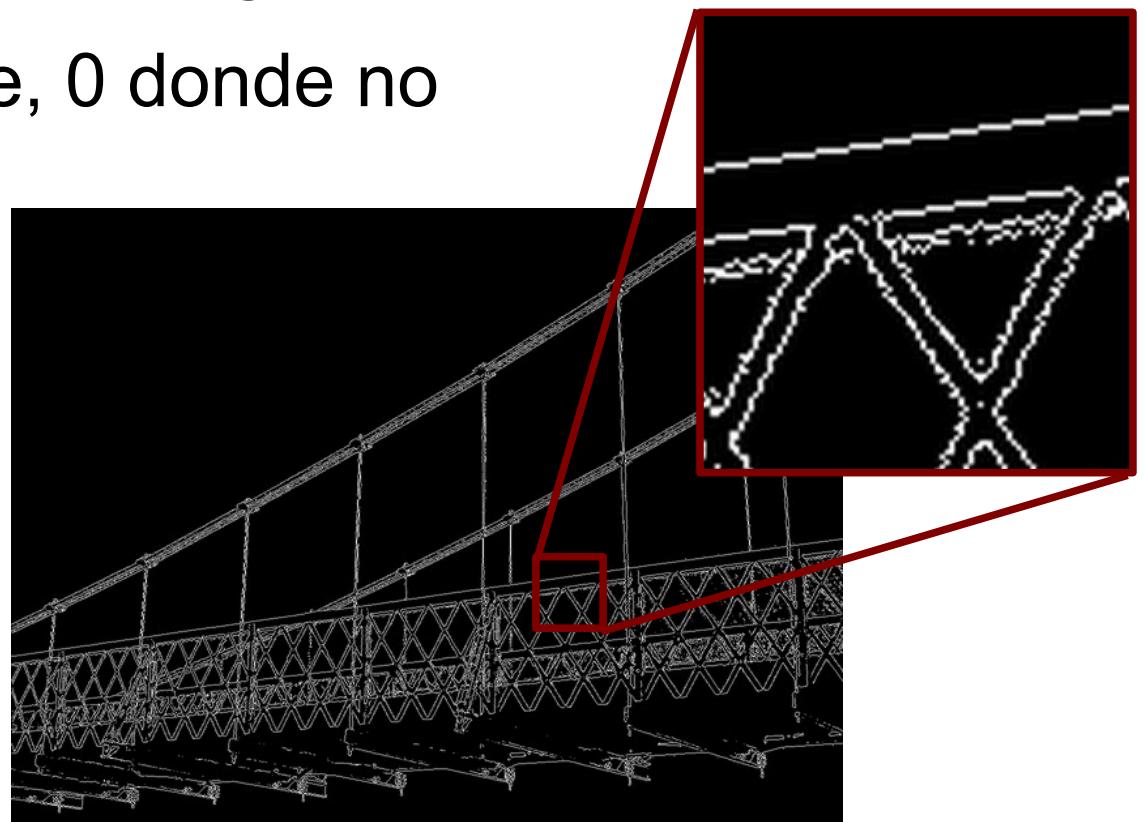
Prewitt



Sobel

Ejemplo clásico de detección de bordes: operador de Canny

- Buscamos binarizar la imagen:
 - 255 donde hay borde, 0 donde no



Operador de Canny

- Problemas del simple umbralizado del módulo del gradiente:
 - Discontinuidad, acumulaciones inhomogéneas de píxeles activos (duplicidad), etc.
- Operador de Canny (1986):
 - Intenta resolver los problemas de duplicidad y discontinuidad.
- Propiedades:
 - Buena detección.
 - Buena localización.
 - Mínima respuesta.

Operador de Canny

- Problemas del simple umbralizado del módulo del gradiente:
 - Después del suavizado, queda demasiado “grueso”:
 - ¿Dónde está el borde realmente?
 - ¿Cómo enlazar para pasar de “puntos de borde” a “contornos”?

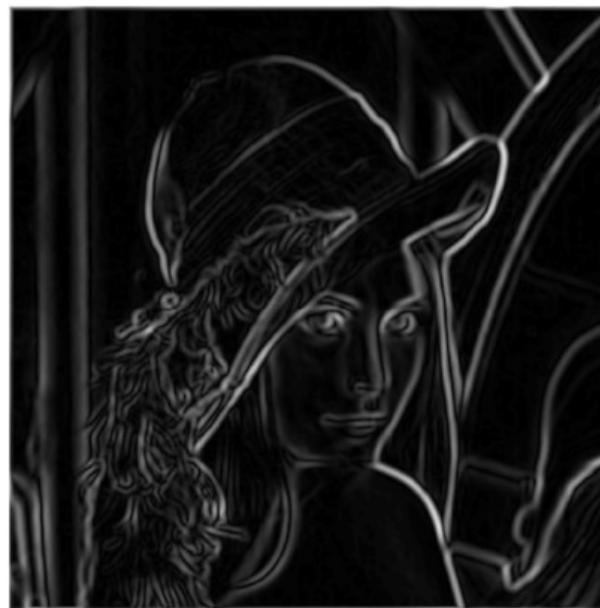


Operador de Canny

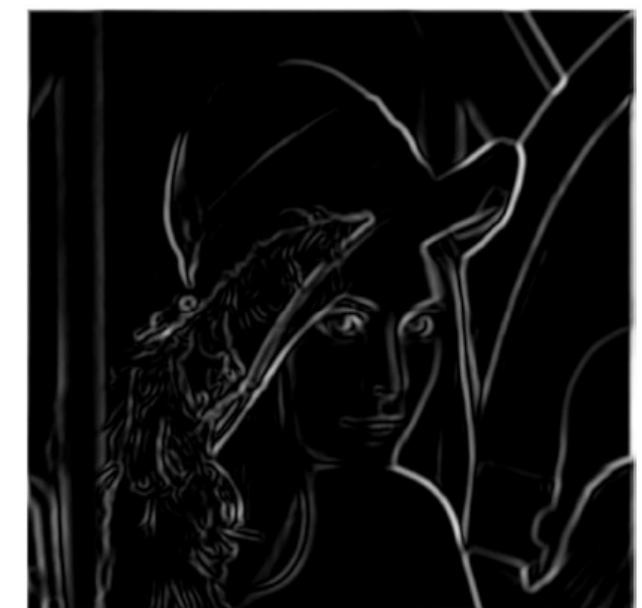
- **Paso 1:** Cálculo del gradiente



Original



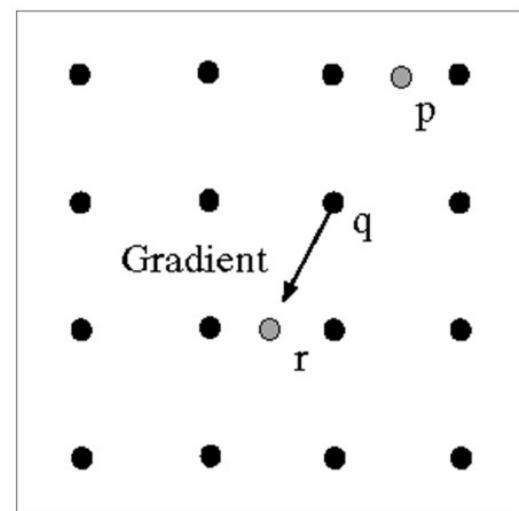
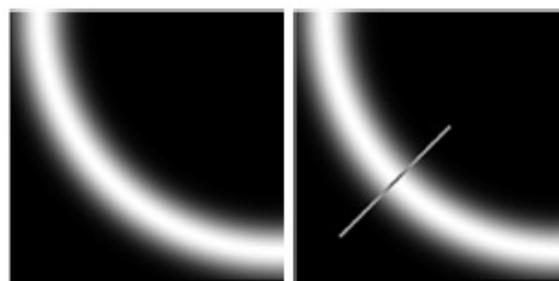
Módulo del
gradiente



Umbralizado
del módulo
del gradiente

Operador de Canny

- **Paso 2:** Supresión de no máximos:
 - Utilizamos la dirección del gradiente



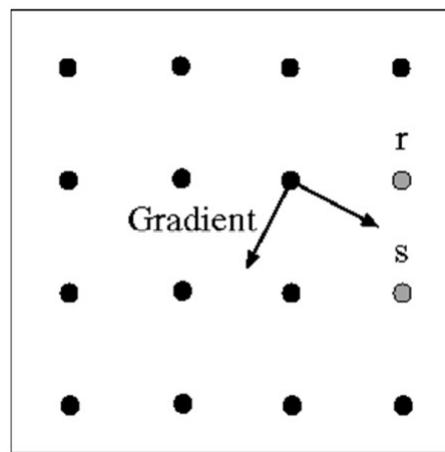
Hay un máximo en q si el valor es mayor que p y r (valores interpolados).

Operador de Canny

- **Paso 3: Histéresis**



Original



Los bordes son perpendiculares al gradiente: si el píxel marcado es de borde, el siguiente será r o s (*edge linking*)



Umbral alto
(bordes “fuertes”)



Umbral bajo
(bordes “débiles”)



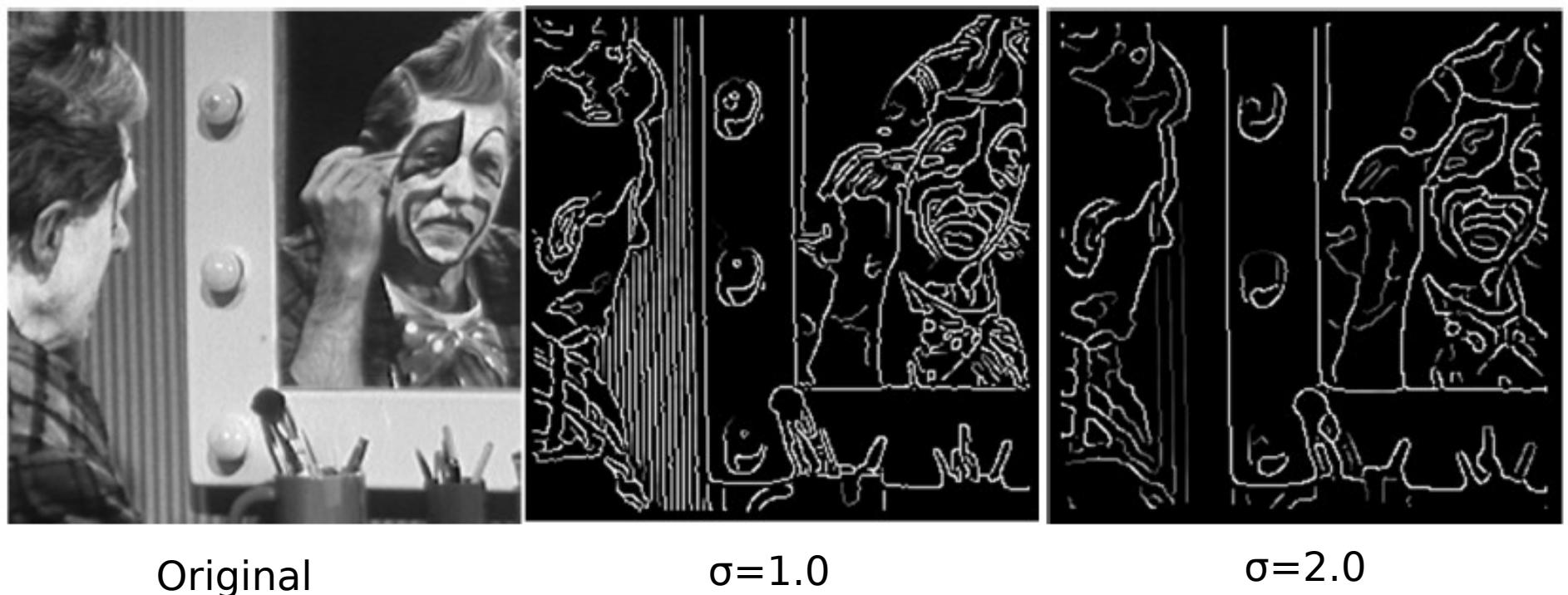
Histéresis

Operador de Canny

- Pasos del operador:
 - 1) **Cálculo de gradiente** sobre imagen suavizada (p.e., Sobel): G_x , G_y , magnitud (raíz de suma de cuadrados) y orientación (arcotangente).
 - 2) **Supresión de no-máximos**: Se buscan y marcan sólo los máximos locales del valor de la magnitud del gradiente en la orientación marcada por el mismo. Así se obtienen “bordes delgados”.
 - 3) **Histéresis**: en el proceso anterior, para que un máximo local se marque como respuesta, tiene que cumplirse:
 - Que la magnitud sea mayor que un umbral **High**.
 - O bien que esté entre un umbral **Low** y el umbral **High**, siempre que el píxel esté conectado a un píxel ya activo.
 - Pero si la magnitud es menor que **Low**, el píxel se descarta directamente.

Operador de Canny

- La **elección de la escala** depende de los tipos de bordes que nos interesen:



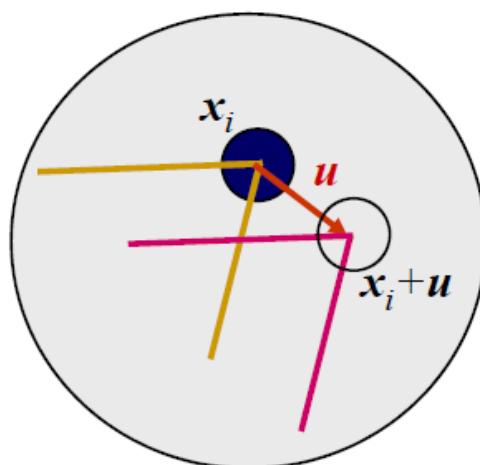
Original

$\sigma=1.0$

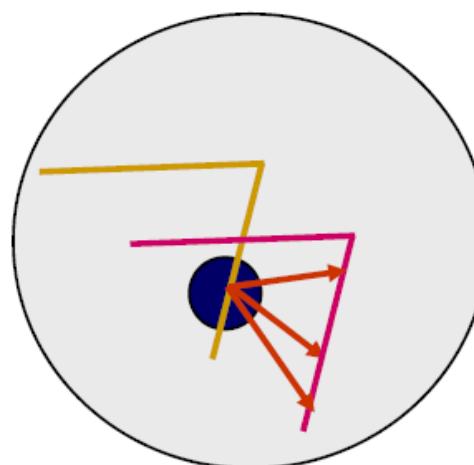
$\sigma=2.0$

Métodos de extracción de puntos

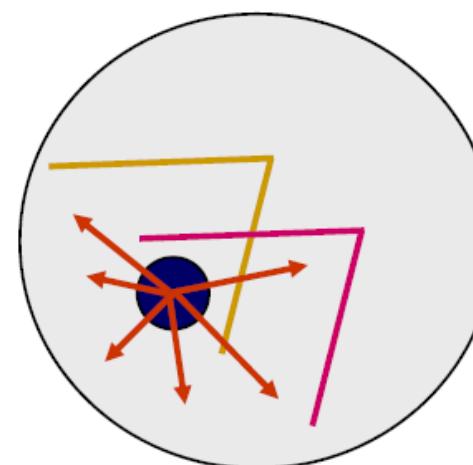
- ¿Cómo encontrar localizaciones de la imagen para las cuales podremos encontrar **correspondencias con fiabilidad** en otra imagen?



(a) Buena localización
2D (*corner-like*)



(b) Mala localización
1D (problema de apertura)



(c) Mala localización
2D (sin textura)

Imagen 1
Imagen 2

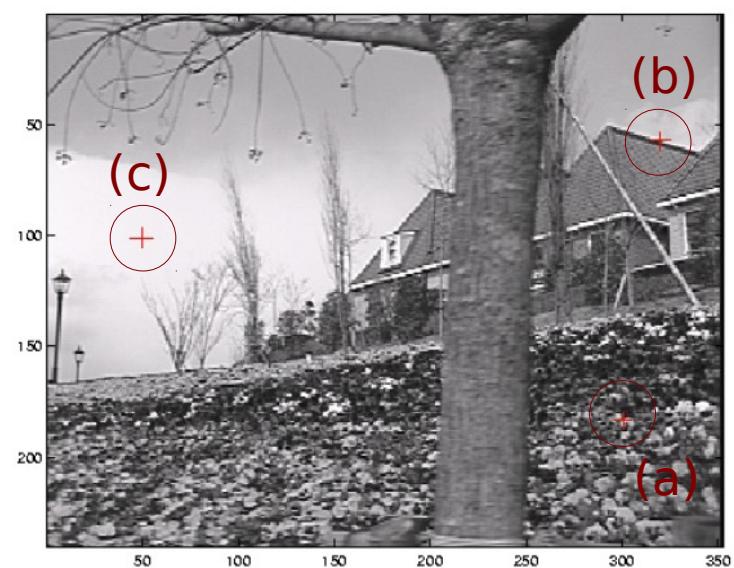
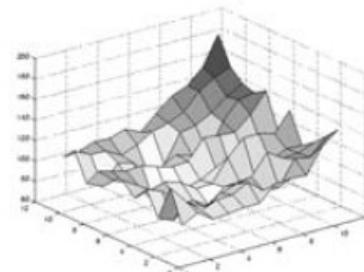
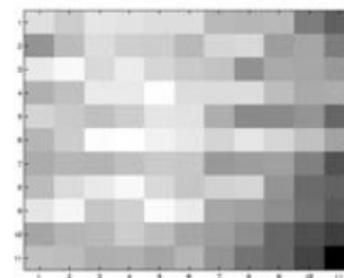
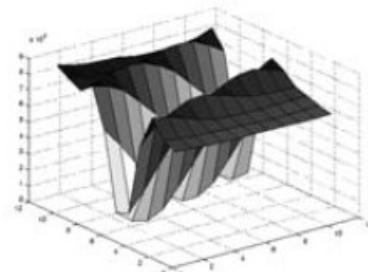
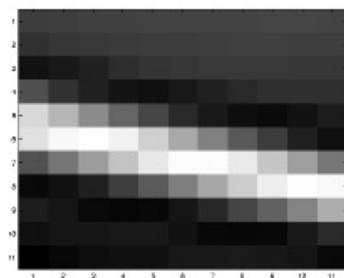
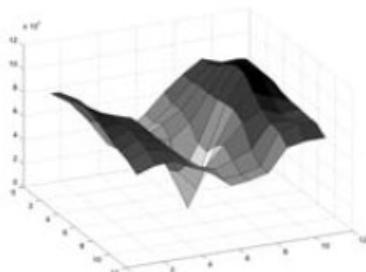
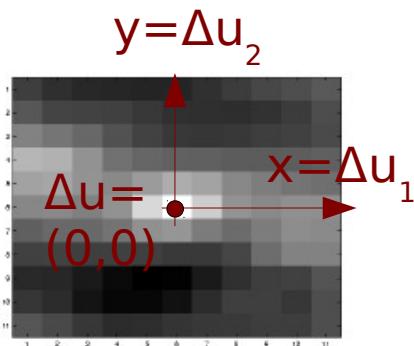
Puntos de imagen “buenos” para relocalizar

- Criterio más simple para comparar dos parches de imagen:
 - Suma ponderada de diferencias al cuadrado (*weighted sum of square differences*):
$$E_{WSSD}(\mathbf{u}) = \sum_i w(\mathbf{x}_i) [I_1(\mathbf{x}_i + \mathbf{u}) - I_0(\mathbf{x}_i)]^2$$

$\mathbf{u} \rightarrow$ desplazamiento entre posiciones
 $w(\mathbf{x}) \rightarrow$ Ponderación (usualmente ventana gaussiana)
 - A priori no sabemos con qué parches de otra imagen nos vamos a comparar → Aún posible métrica de *matching* consigo mismo, con respecto a pequeñas variaciones $\Delta\mathbf{u}$ (autocorrelación):
$$E_{AC}(\Delta\mathbf{u}) = \sum_i w(\mathbf{x}_i) [I_0(\mathbf{x}_i + \Delta\mathbf{u}) - I_0(\mathbf{x}_i)]^2$$

Superficie de correlación

- Tres ejemplos de superficies de correlación $E_{AC}(\Delta u)$:



(a) Buen mínimo local único
(buena localización)

(b) Mínimo a lo largo de una recta (problema de apertura)

(c) Mínimo poco estable (pobre localización)

Superficie de correlación

- Expansión de Taylor para aproximar $E_{AC}(\Delta u)$:

$$\begin{aligned} E_{AC}(\Delta \mathbf{u}) &= \sum_i w(\mathbf{x}_i) [I_0(\mathbf{x}_i + \Delta \mathbf{u}) - I_0(\mathbf{x}_i)]^2 \\ &\approx \sum_i w(\mathbf{x}_i) [I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i) \cdot \Delta \mathbf{u} - I_0(\mathbf{x}_i)]^2 \\ &= \sum_i w(\mathbf{x}_i) [\nabla I_0(\mathbf{x}_i) \cdot \Delta \mathbf{u}]^2 \\ &= \Delta \mathbf{u}^T \mathbf{A} \Delta \mathbf{u}, \quad \xleftarrow{\text{Función cuadrática de } \Delta \mathbf{u}: (\mathbf{A} \text{ es su matriz Hessiana)}} \end{aligned}$$

Donde:

$$\nabla I_0(\mathbf{x}_i) = \left(\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y} \right)(\mathbf{x}_i)$$

Tensor de estructura

- A es la matriz de correlación que proporciona información sobre la forma (cuadrática) local de $E_{AC}(\Delta u)$:

$$A = \sum_{u_1} \sum_{u_2} w(u_1, u_2) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix},$$

- Veremos que a partir de sus dos valores propios λ_1 y λ_2 → determinaremos si la *feature* se presta a *una buena localización*.

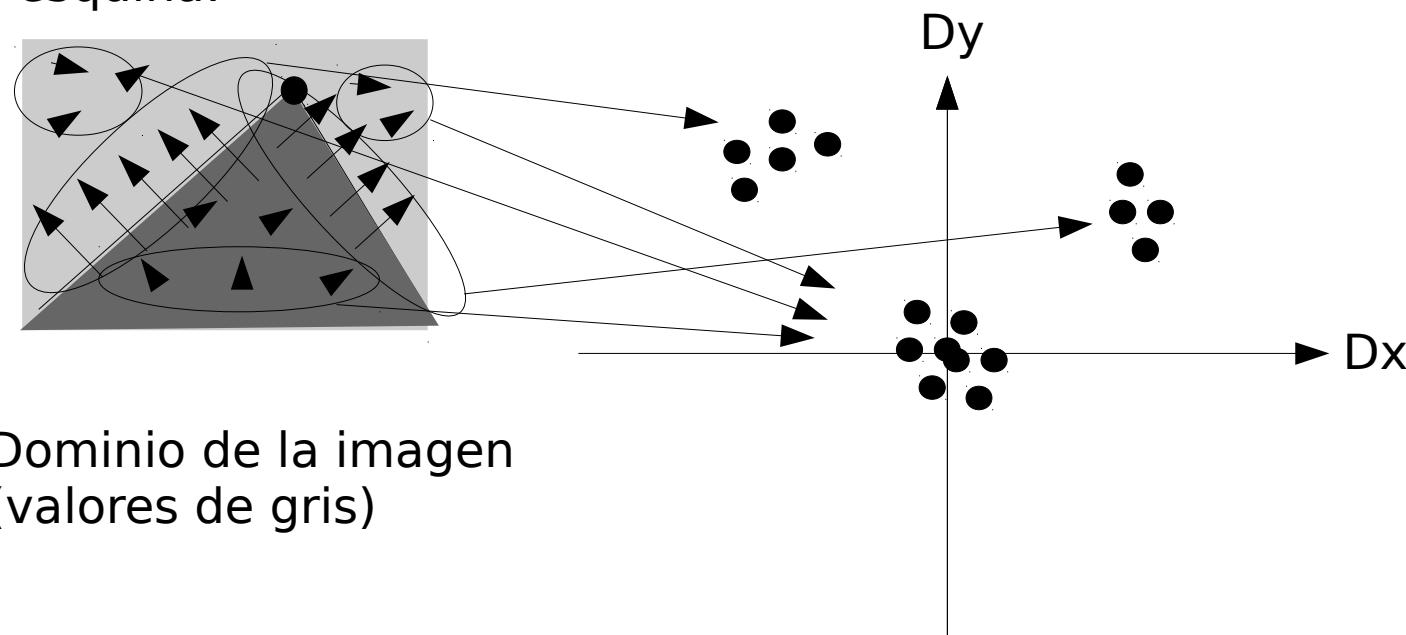
$$(\lambda_1, \lambda_2) = \text{eigvals}\left(\begin{bmatrix} a & b \\ b & c \end{bmatrix}\right) = \frac{1}{2} \left(a + c \pm \sqrt{4b^2 + (a - c)^2} \right)$$

- A veces, se denomina a la matriz A el **tensor de estructura**.

Interpretación geométrica

- Estudio de valores del gradiente en un entorno local :

Entorno local
de una esquina:

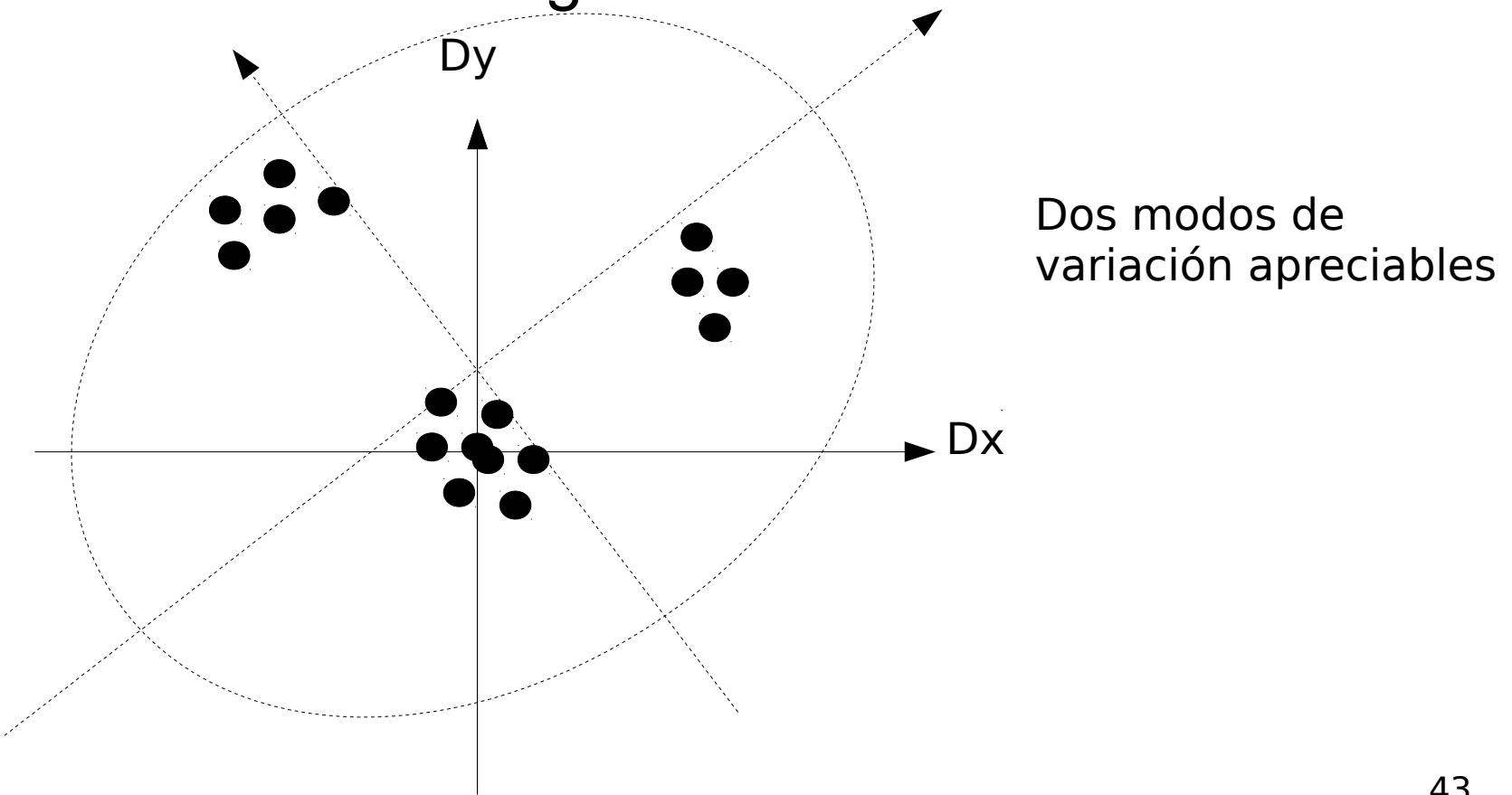


Dominio de la imagen
(valores de gris)

Dominio del gradiente
(nube de puntos)

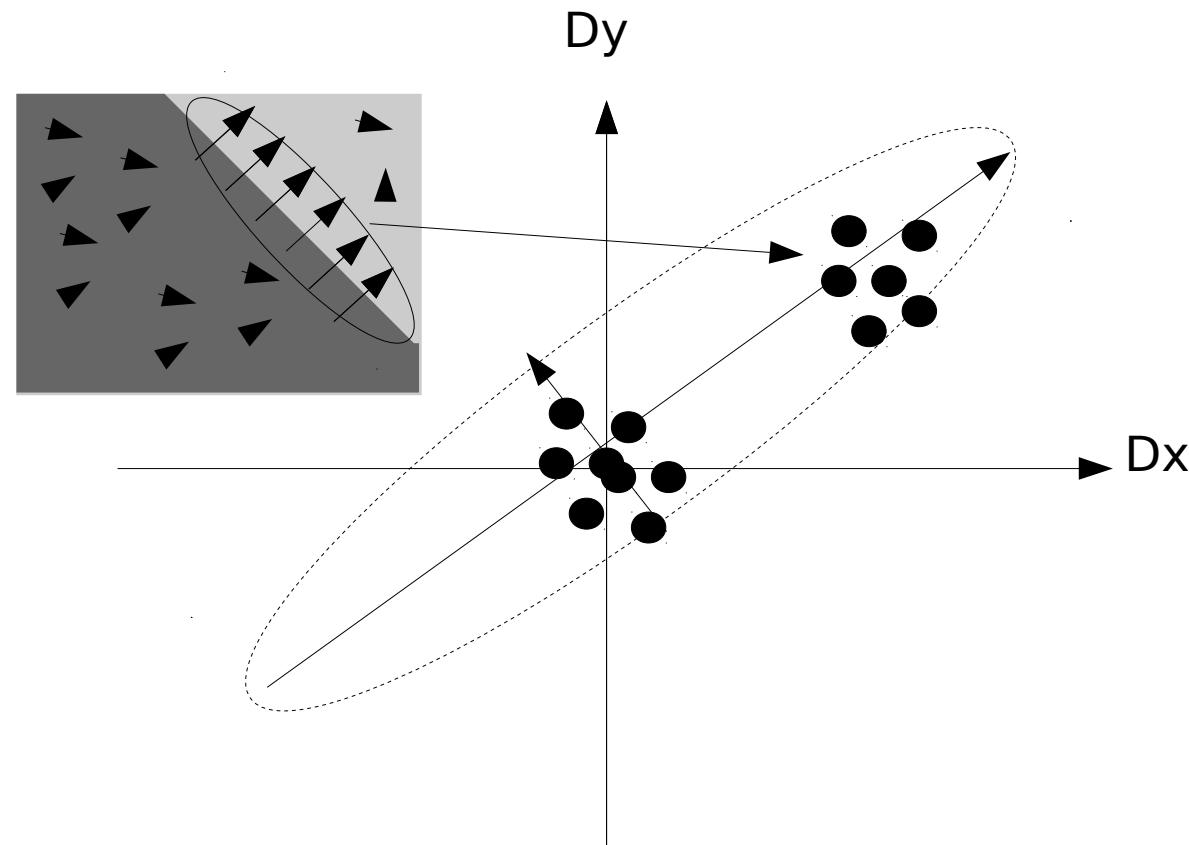
Interpretación geométrica

- Caracterización de las direcciones principales de la distribución de los vectores gradiente:



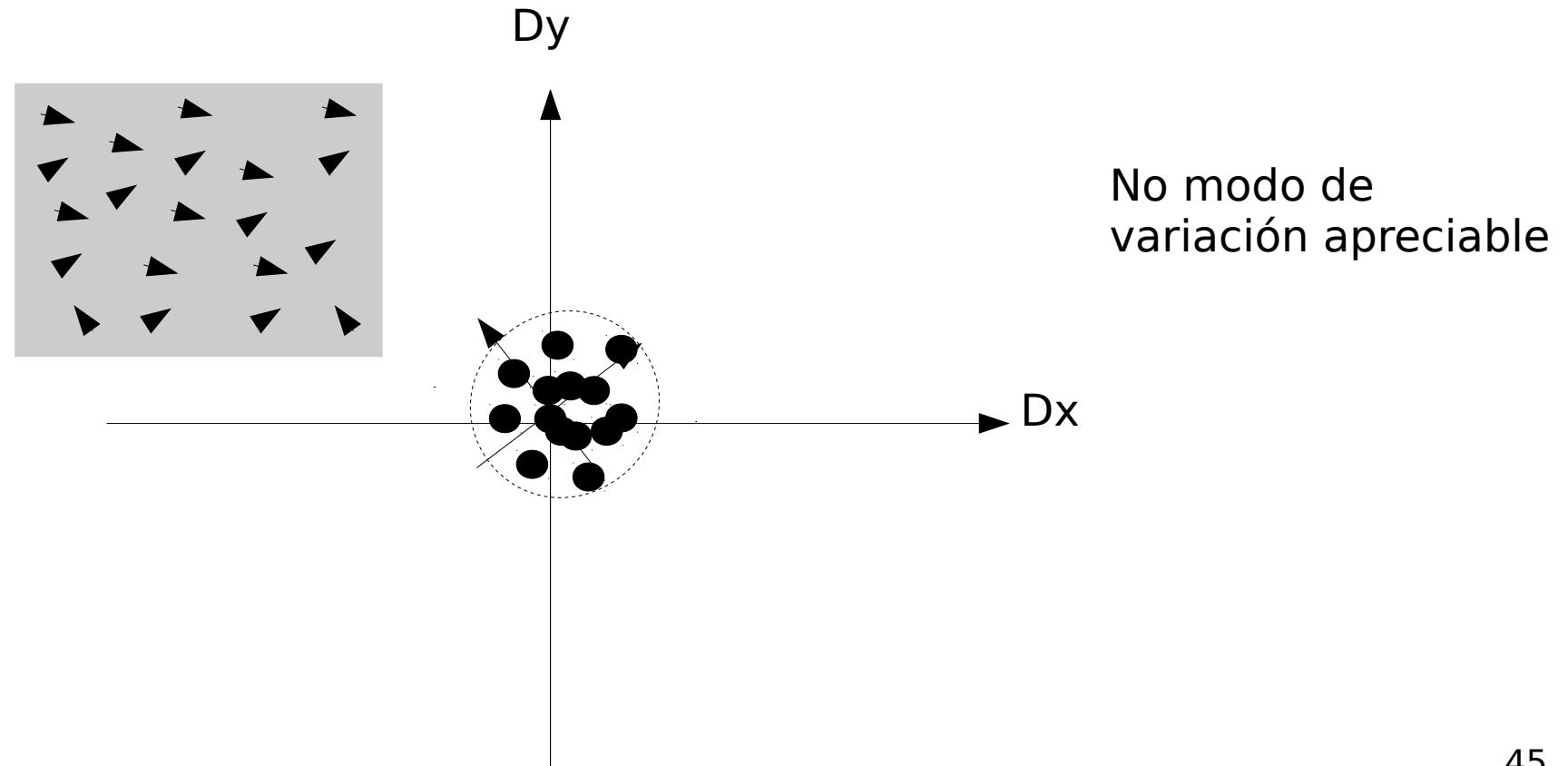
Interpretación geométrica

- Comparar con un borde recto...:



Interpretación geométrica

- ... y con una zona más o menos “lisa”:



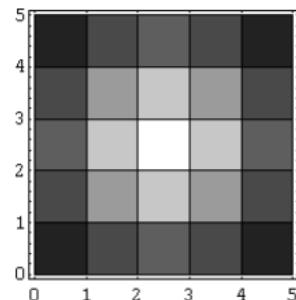
Método de extracción de “esquinas”

- Resumen de la idea:

- En el entorno local de cada pixel (ventana), calcular la siguiente matriz, que caracteriza la distribución espacial del gradiente:

$$A = \sum_u \sum_v w(u, v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix},$$

(Opcional)
Ventana de
“weighting”

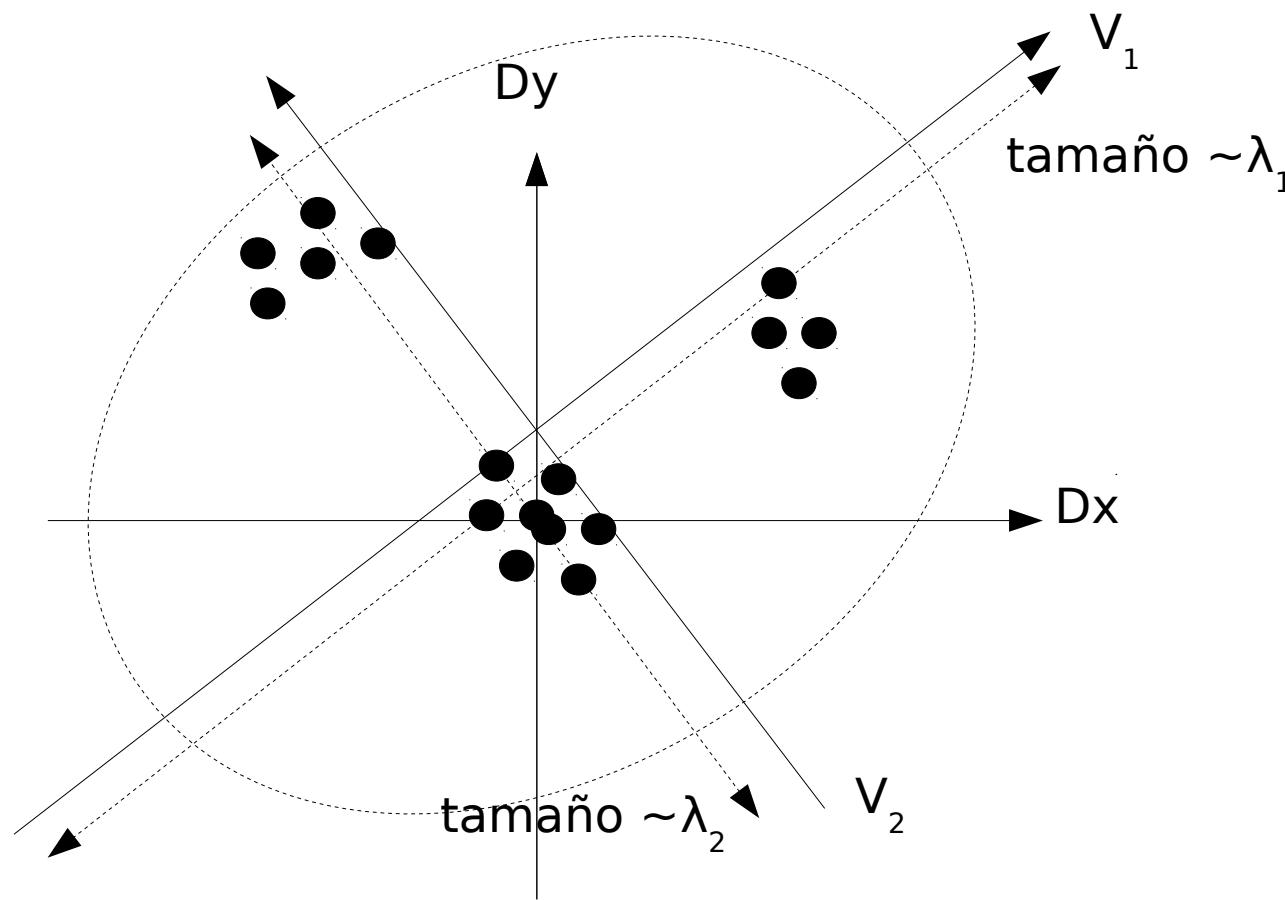


I_x = derivada parcial en X

I_y = derivada parcial en Y

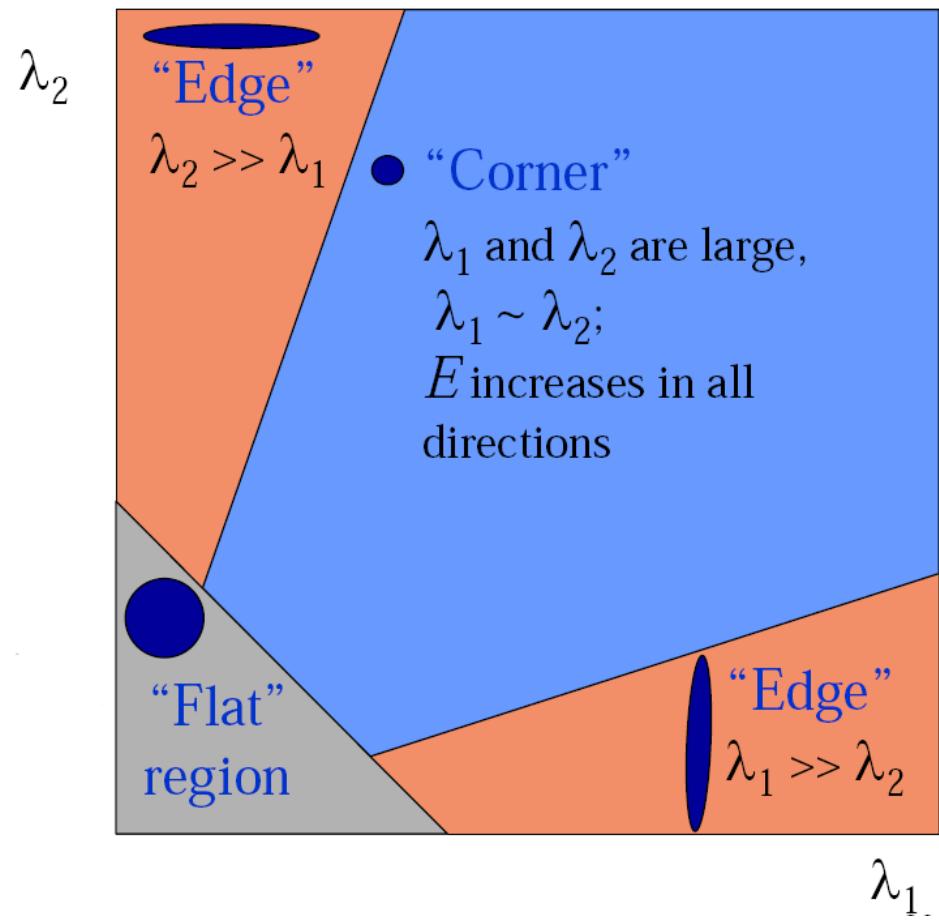
Método de extracción de “esquinas”

- Análisis de valores λ_1 y λ_2 y vectores propios v_1 y v_2 de la matriz A:



Método de extracción de “esquinas”

- Para cada punto de interés (esquina, o *corner*), A debería tener dos valores propios “grandes”. Así:
 - Si $\lambda_1 \sim 0$ y $\lambda_2 \sim 0$ → No hay característica de interés en este entorno (zona “lisa”).
 - Si $\lambda_1 >> 0$ y $\lambda_2 \sim 0$ → Simplemente hay un borde apreciable en este entorno (pero no una esquina).
 - Si $\lambda_1 >> 0$ y $\lambda_2 >> 0$ → Se ha detectado una esquina en este punto.



Operador de Harris / alternativas

- Cálculo exacto valores propios es innecesario, sólo interesa saber si hay más de una dirección dominante → **test de Harris**:

$$M_c = \lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 = \det(A) - \kappa \text{trace}^2(A)$$

$\downarrow \quad \quad \quad \downarrow$

$$a_{11} * a_{22} - a_{21} * a_{12} \quad (a_{11} + a_{22})^2$$

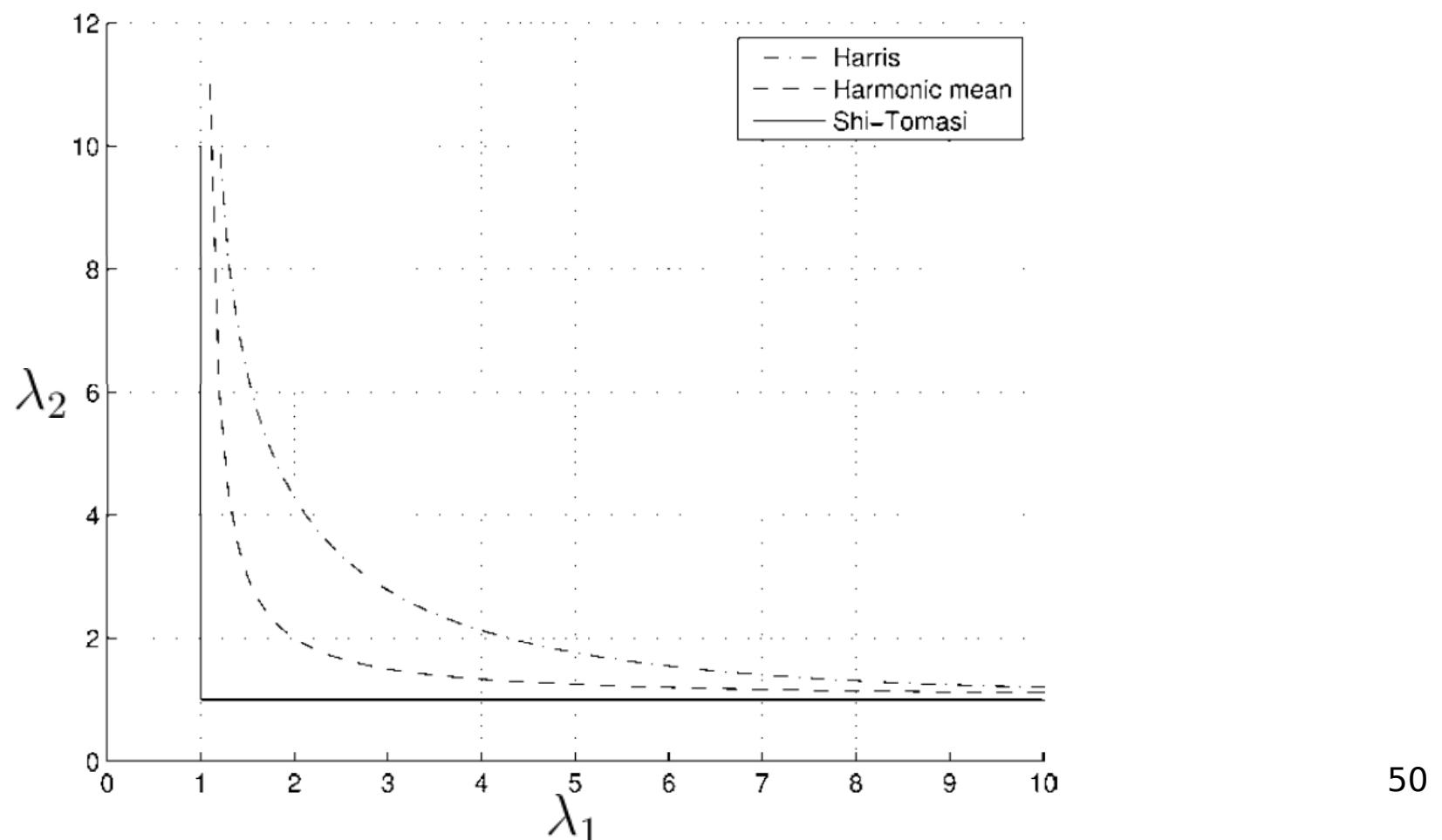
donde k es un parámetro de sensitividad que se determina por prueba y error. Valores entre 0.05 y 0.15 suelen ser válidos.

Mc ~ 0 → región plana
Mc < 0 → borde recto
Mc >> 0 → esquina

- Shi-Tomasi, de todos modos, demuestran que en realidad tomar el **$\min(\lambda_1, \lambda_2)$** es mejor.
- Otra posibilidad es la **media armónica**:
$$\frac{\det(A)}{\text{tr}(A)} = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

Operador de Harris / alternativas

- *Isocontornos* de los tres tipos de test:



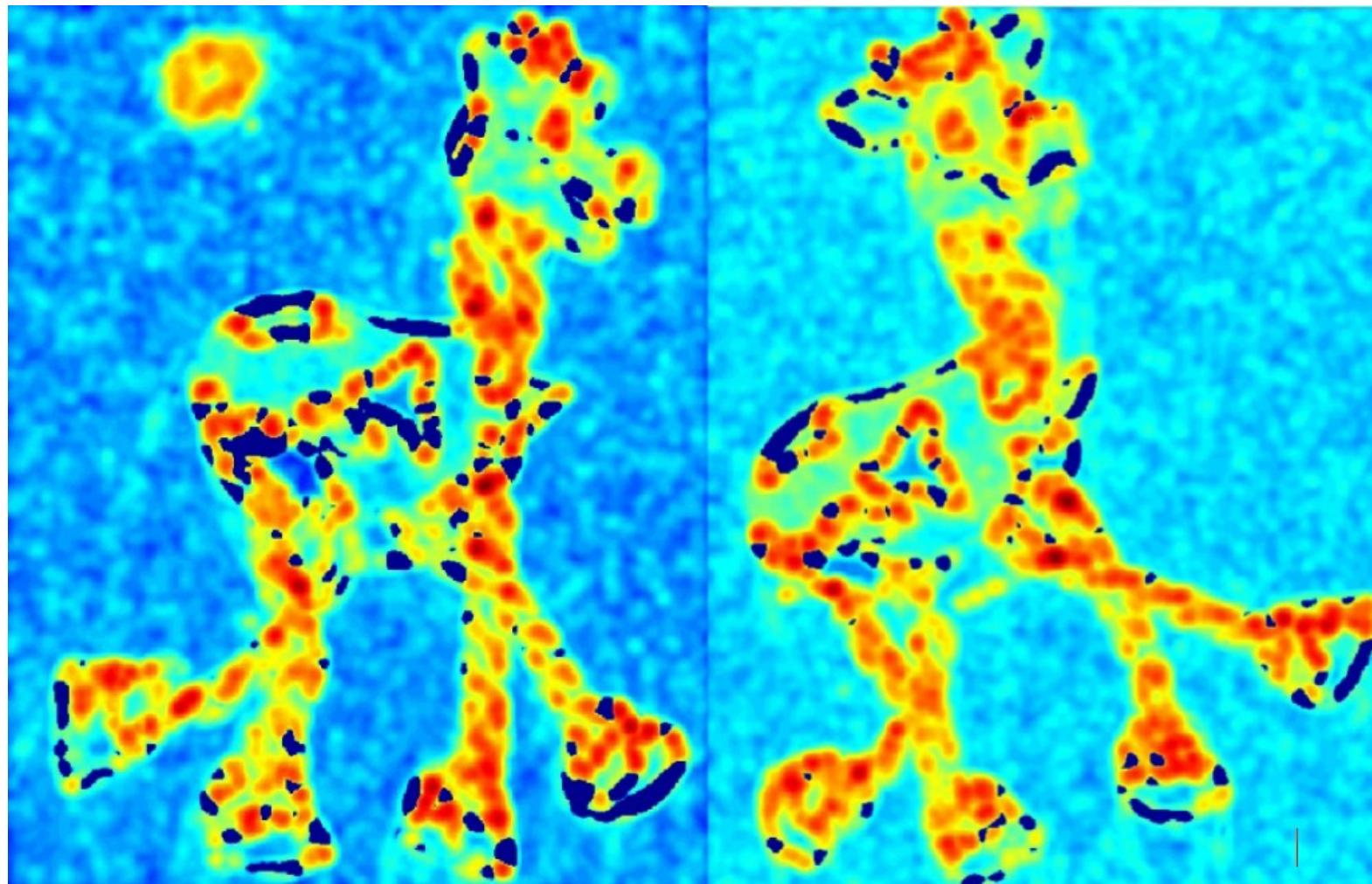
Ejemplo de uso del operador de Harris

- Imágenes de entrada originales:



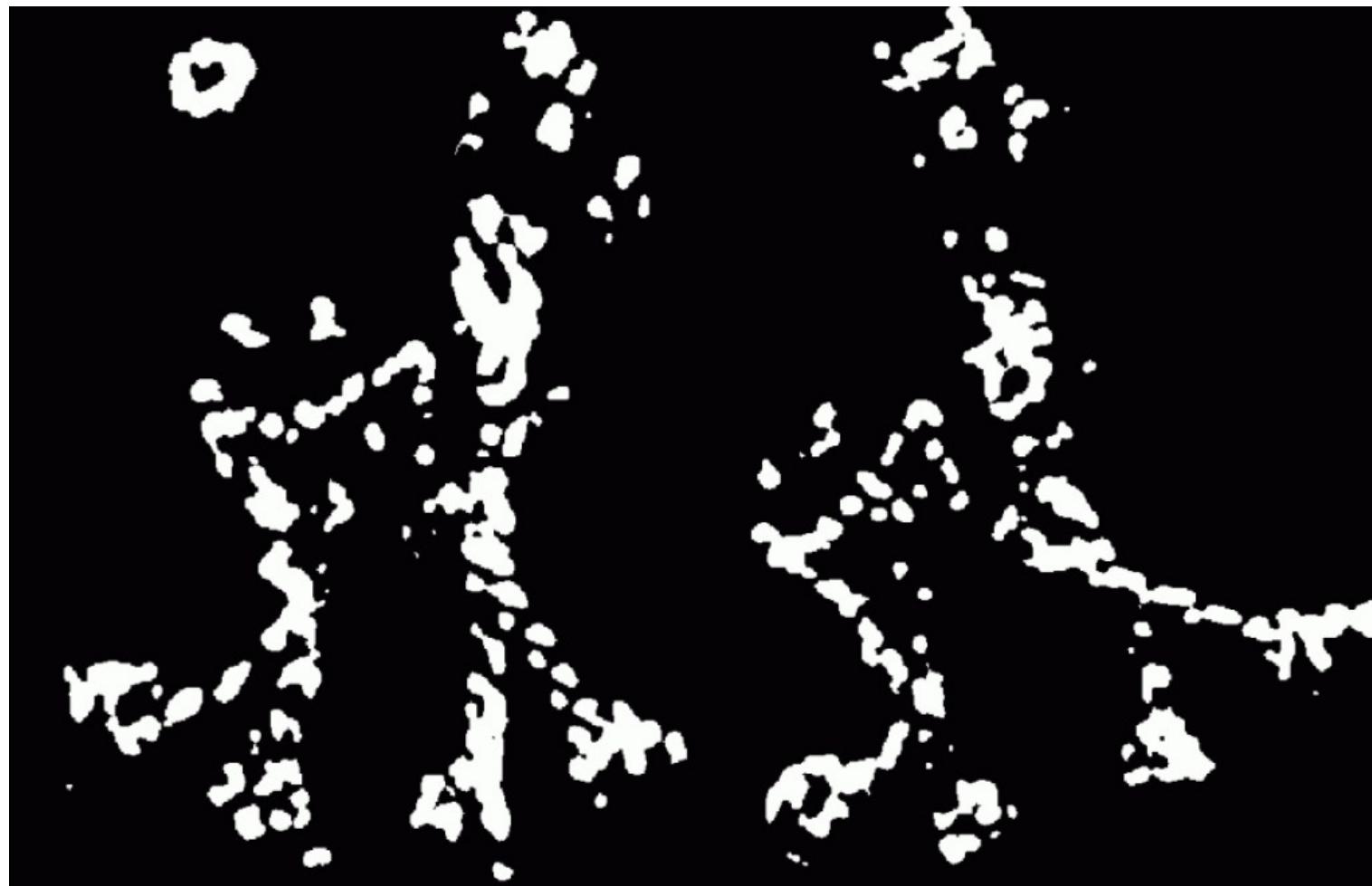
Ejemplo de uso del operador de Harris

- Salida del operador de respuesta M_c (naranja=más alto):



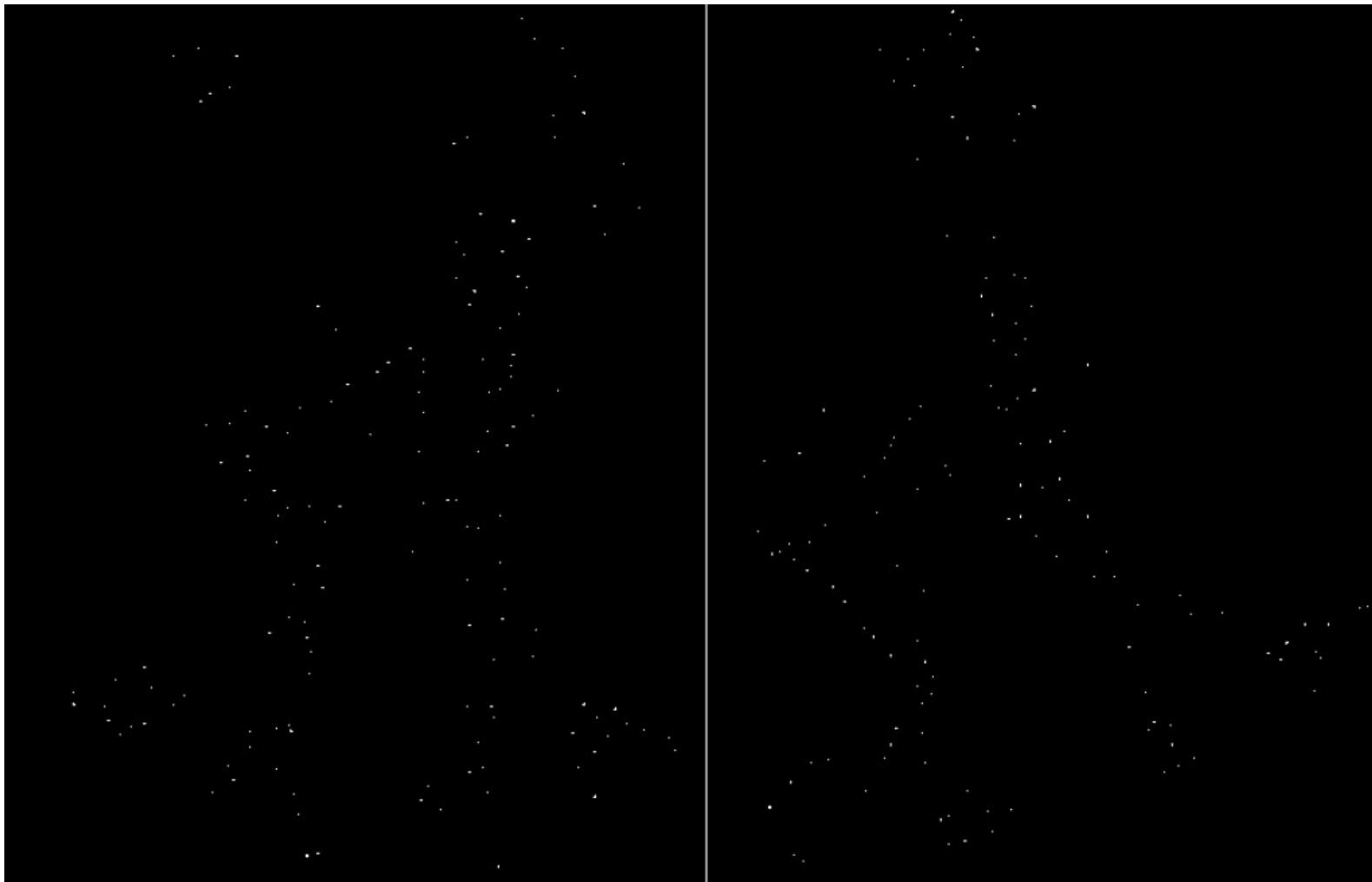
Ejemplo de uso del operador de Harris

- Umbralizado del valor de respuesta:



Ejemplo de uso del operador de Harris

- Detección de máximos locales:



Ejemplo de uso del operador de Harris

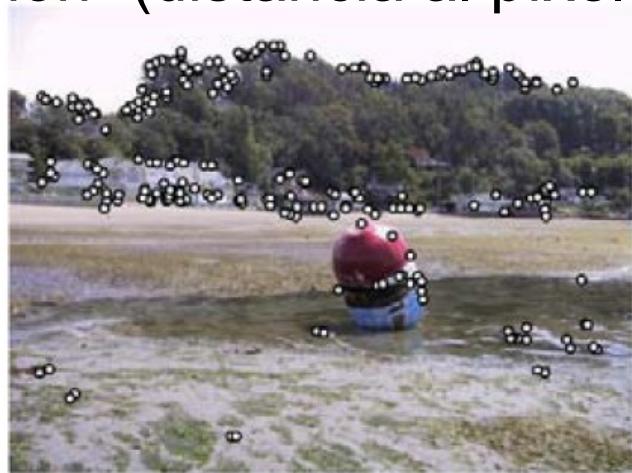
- Esquinas detectadas sobre imágenes originales:



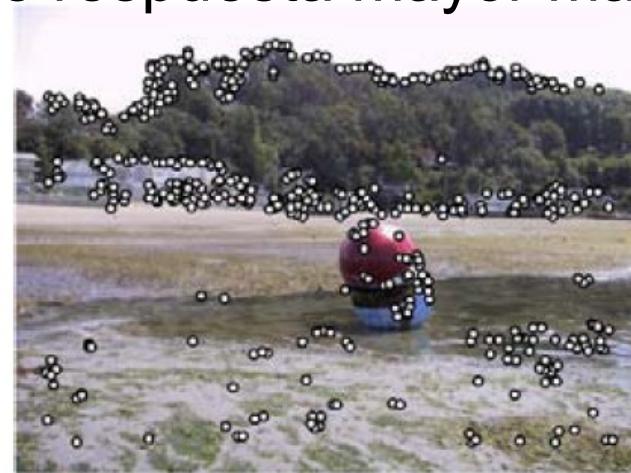
Supresión de no máximos adaptativa

- **ANMS (Adaptive Non-maxima Suppression)**
 - Se ordenan puntos (que superan un umbral) por “radio de supresión” (distancia al píxel de respuesta mayor más cercano):

250 mayores respuestas



500 mayores respuestas



250 mayores respuestas,
Radio supresión
24 píxeles



500 mayores respuestas,
Radio supresión
16 píxeles



Hessiano local

- Usando una idea similar a la del detector de Harris anterior, pero sustituyendo la matriz A con la matriz H (ambas de tamaño 2×2) de segundas derivadas de la imagen, en lugar de la calculada con gradientes:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

- Este es exactamente el Hessiano de la aproximación local cuadrática al parche de la imagen (aquí consideramos $(x,y)=(0,0)$ el centro del parche):

$$im(x, y) = f(x, y) \approx f(0, 0) + \nabla f(0, 0) \cdot (x, y) + \frac{1}{2}(x, y) \cdot H \cdot (x, y)^\top$$

- Después, el análisis de valores propios es análogo: si hay una fuerte asimetría en la curvatura local, se rechaza el punto (por no tener estructura suficiente):

Si $\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} > 10 \rightarrow \text{rechazar}$

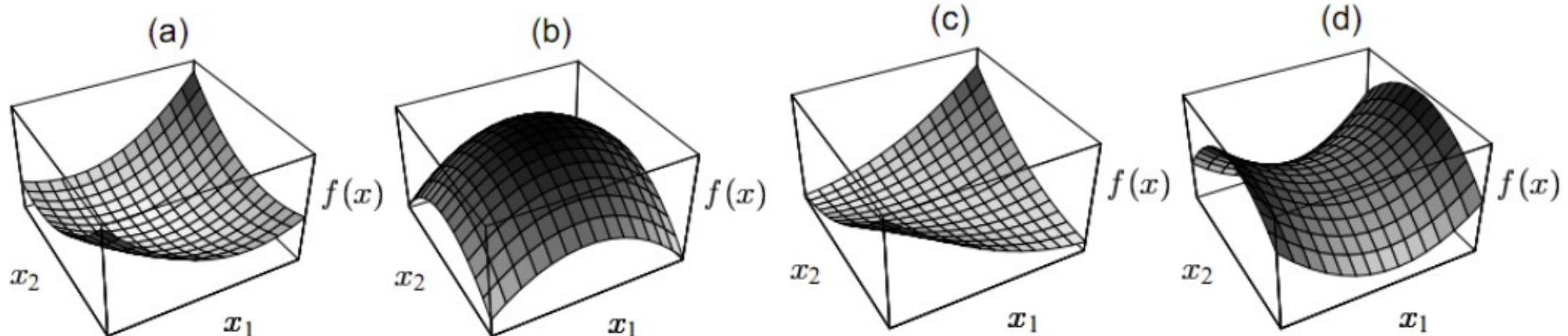
Hessiano local (cont.)

- El detector de *blobs* basado en el Hessiano sigue el mismo test de “falta de estructura suficiente” en SIFT (que veremos en el tema 5):

$$\begin{aligned} \text{Traza de } H &\longrightarrow \text{Tr}(H) = D_{xx} + D_{yy} = \alpha + \beta, & \text{Valores propios de } H \\ \text{Determinante de } H &\longrightarrow \text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta. & \\ \frac{\text{Tr}(H)^2}{\text{Det}(H)} &= \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} & \text{Ratio entre valores propios } (r = \alpha/\beta) \\ \frac{\text{Tr}(H)^2}{\text{Det}(H)} &< \frac{(r+1)^2}{r} & \text{Test sencillo, para un } r \text{ deseado (no necesita calcular valores propios)} \end{aligned}$$

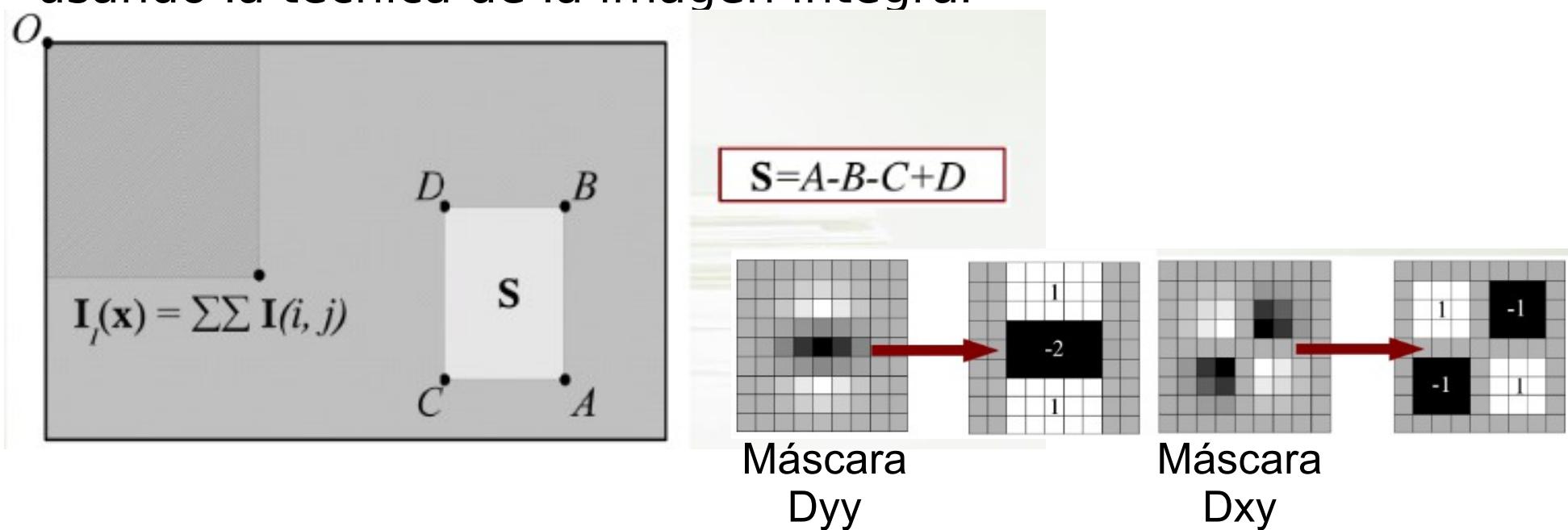
Hessiano local

- (a) y (b): ambos valores propios λ_1 y λ_2 positivos / negativos, y lejanos de cero → *blob oscuro / claro, interesa.*
- (c): un valor propio positivo, otro cercano a cero → problema de apertura, no interesa.
- (d): ambos valores propios bastante mayores que cero en valor absoluto, pero uno positivo y otro negativo → punto de silla (*saddle point*), interesa.
- Para otro tipo de zonas (planas, con o sin pendiente), ambos valores propios cercanos a cero → no interesa.



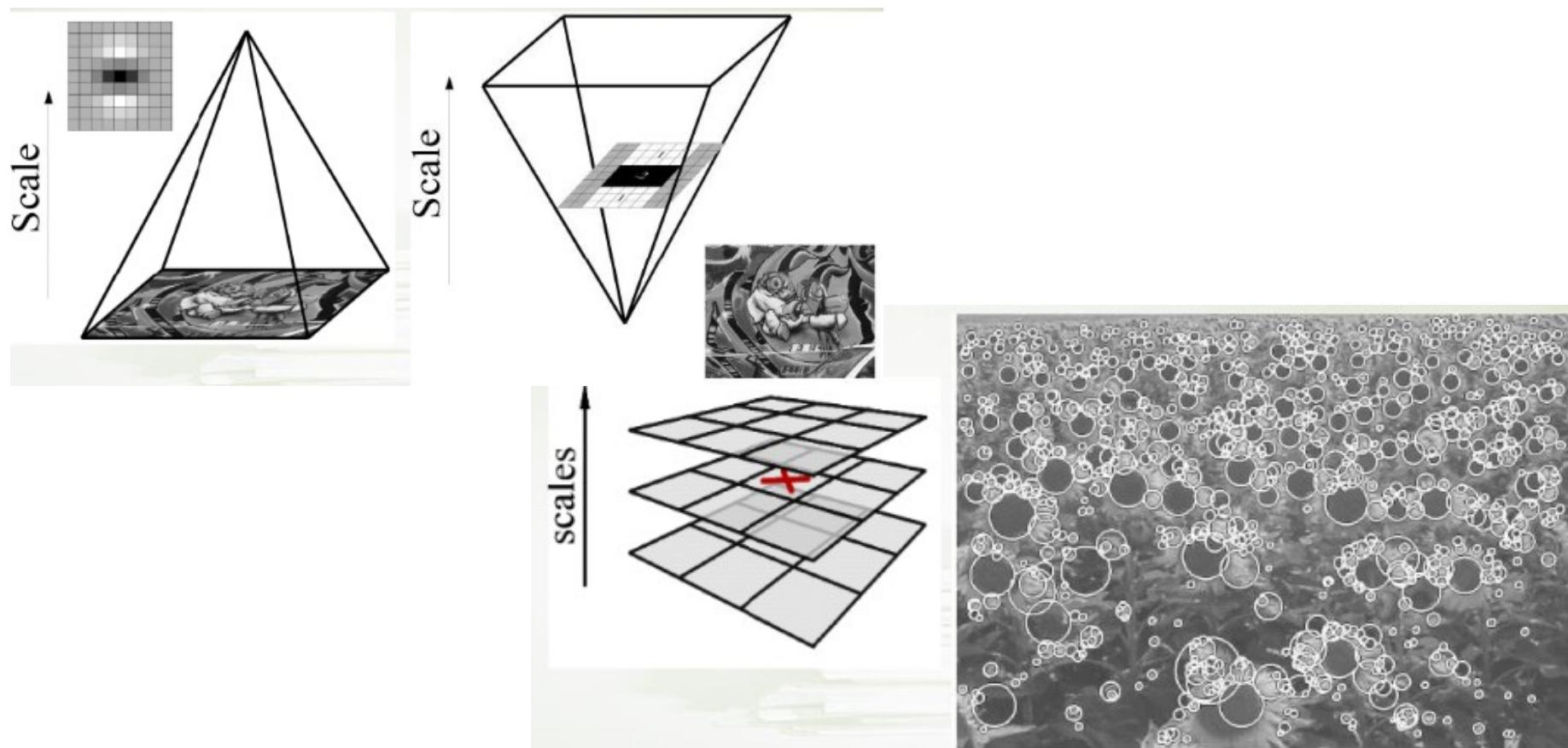
Hessiano local

- Suele usarse más para entornos en escala variable.
- P.e. el detector **SURF**, (*Speeded Up Robust Features*) que trabaja con máscaras de sumas y restas en áreas rectangulares/cuadradas para estimar las segundas derivadas (en lugar de con filtros gaussianos), por ser fácilmente aplicables de modo multiescala usando la técnica de la imagen integral



Hessiano local

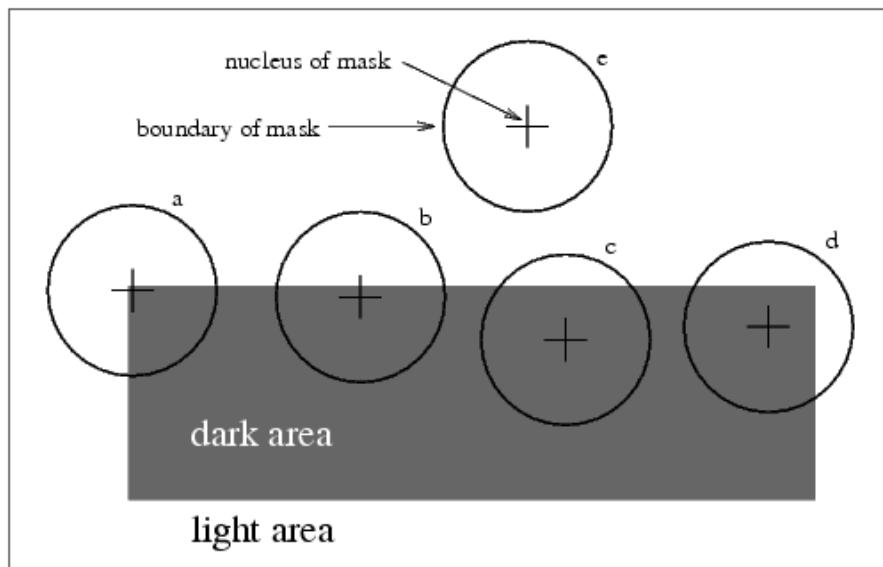
- Ejemplo detección multiescala SURF:



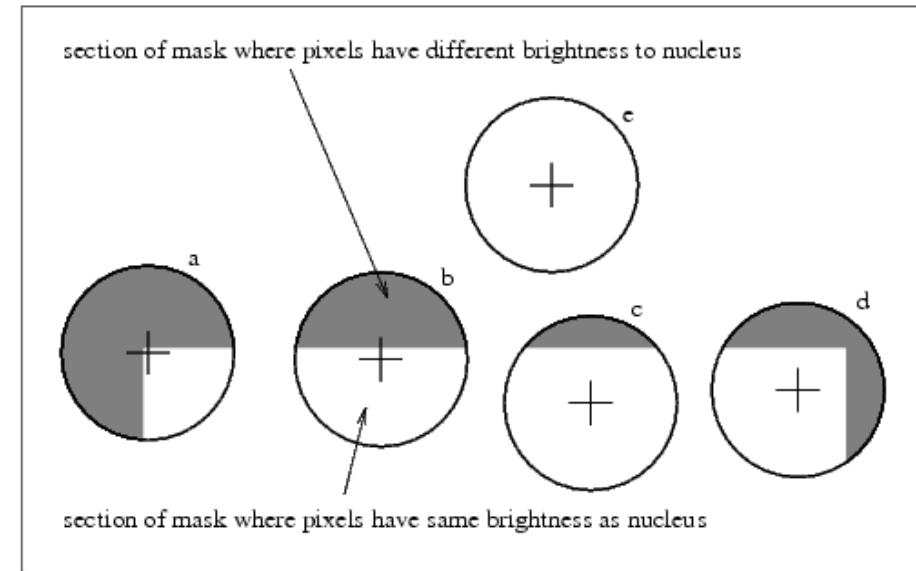
- En realidad, bastante análogo a SIFT, que veremos en más detalle más adelante en el curso.

Más detectores de esquinas: el operador SUSAN

- Operador SUSAN (*Smallest Univalue Segment Assimilating Nucleus*):



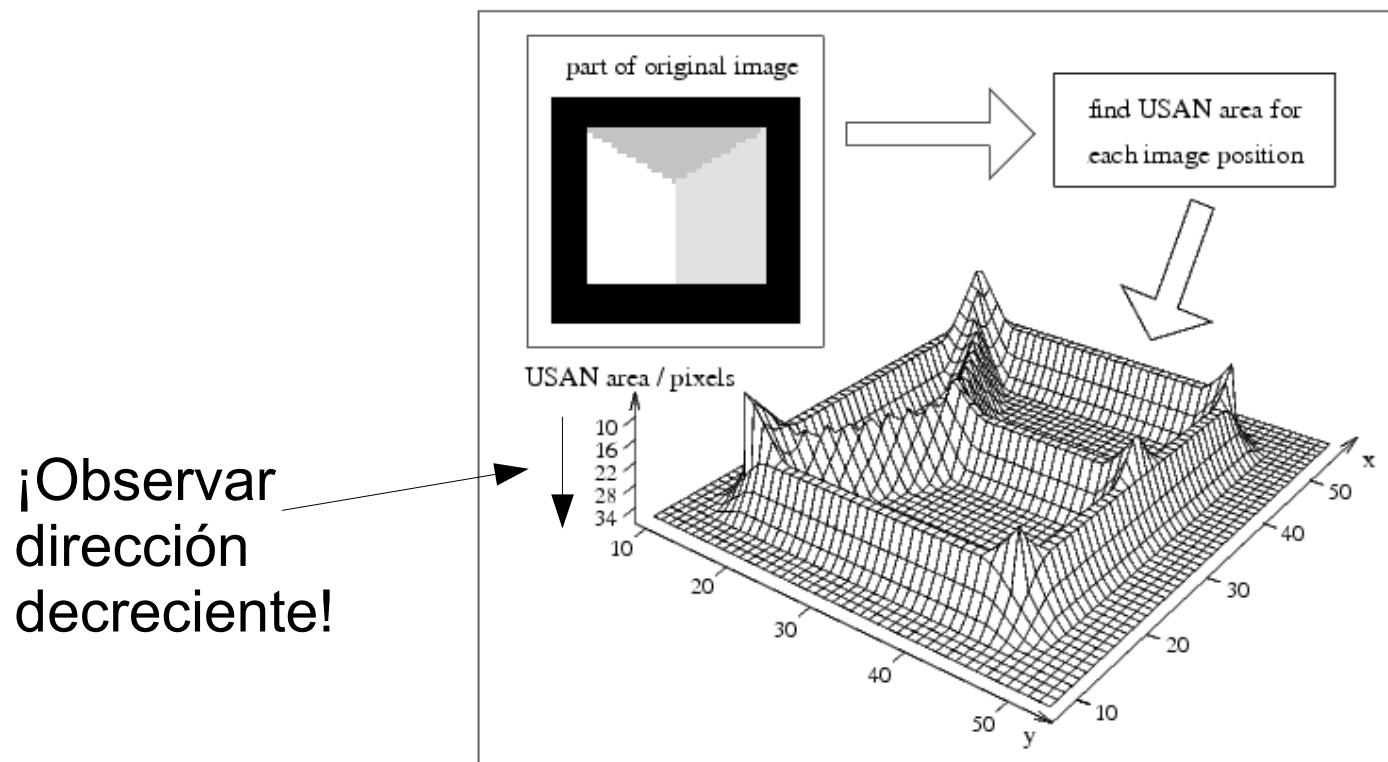
Aplicar máscaras circulares por toda la imagen.



Comprobar en cada punto la sección de la máscara que tiene brillo parecido al del núcleo (USAN)

Operador SUSAN

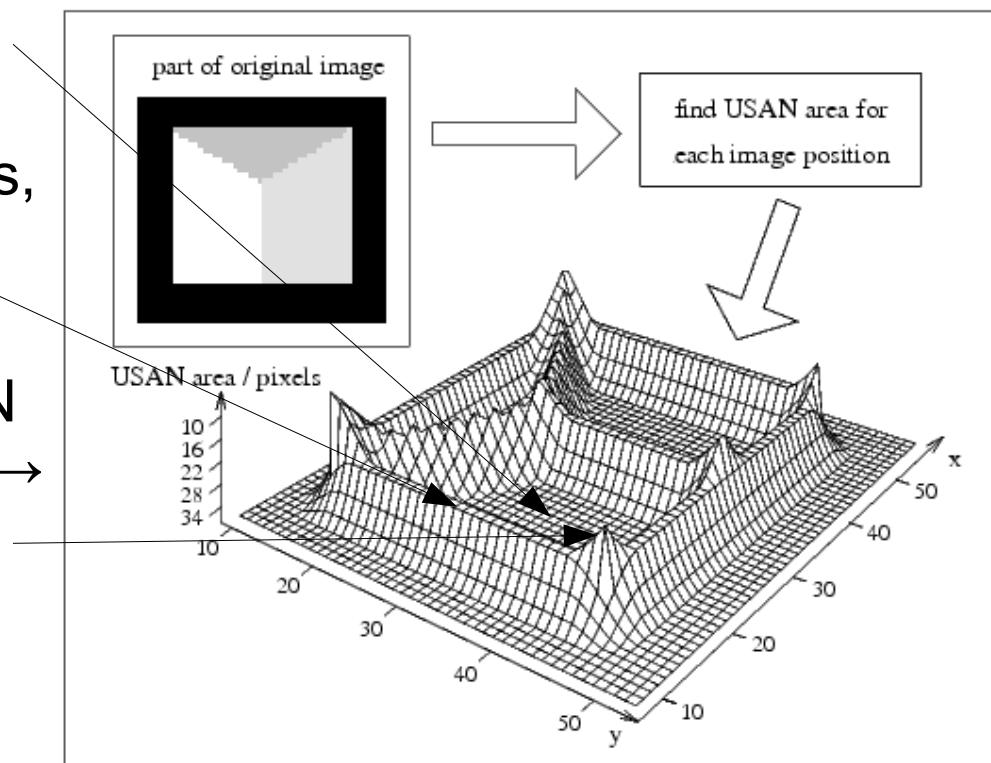
- Área del USAN (Univalue Segment Assimilating Nucleus) en una imagen de test:



Operador SUSAN

- Tres casos:

- (1) Para zonas lisas, el área del USAN es máxima → no interés.
- (2) Para bordes más o menos rectos, el área del USAN es apreciablemente menor → borde.
- (3) Para esquinas, el área del USAN es menor que en el resto de casos → esquina detectada.



Operador SUSAN

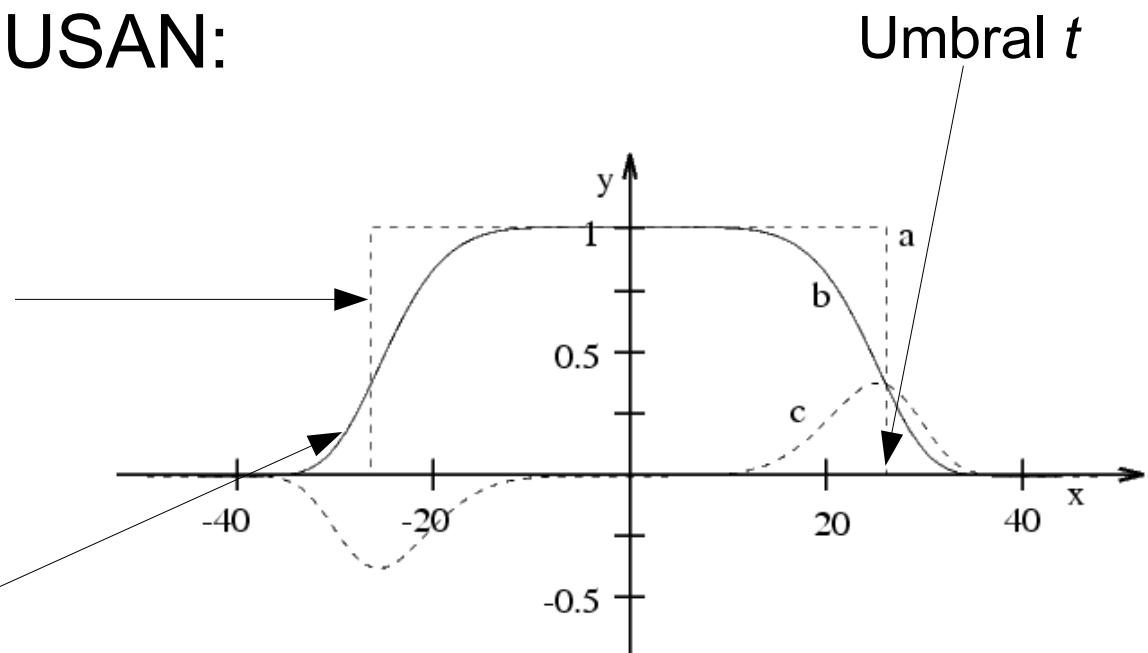
- Función de pertenencia al USAN:

- Discreta (a):

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

- Continua (b; más estable):

$$c(\vec{r}, \vec{r}_0) = e^{-\left(\frac{|I(\vec{r}) - I(\vec{r}_0)|}{t}\right)^6}.$$



- Estimación del área total del USAN:

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

Este valor se compara con un “umbral geométrico” g (usualmente ajustado a $\frac{3}{4}$ de n_{\max} , e.d., del tamaño en píxeles de la máscara). En caso de ser menor, hay un borde. Y si es menor que $\frac{1}{2}$ de n_{\max} entonces es una esquina.

Operador SUSAN

- Valor de respuesta a borde/esquina (cambiando g):

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

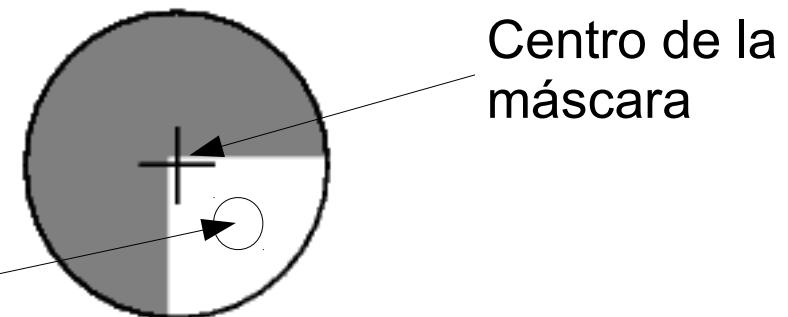


Ya es siempre mayor o igual que cero, y tanto mayor cuanta más “importancia” tenga la esquina (= más “aguda” sea).

- Testeo de seguridad: comprobar que el centro de masas del USAN esté claramente lejos del núcleo:

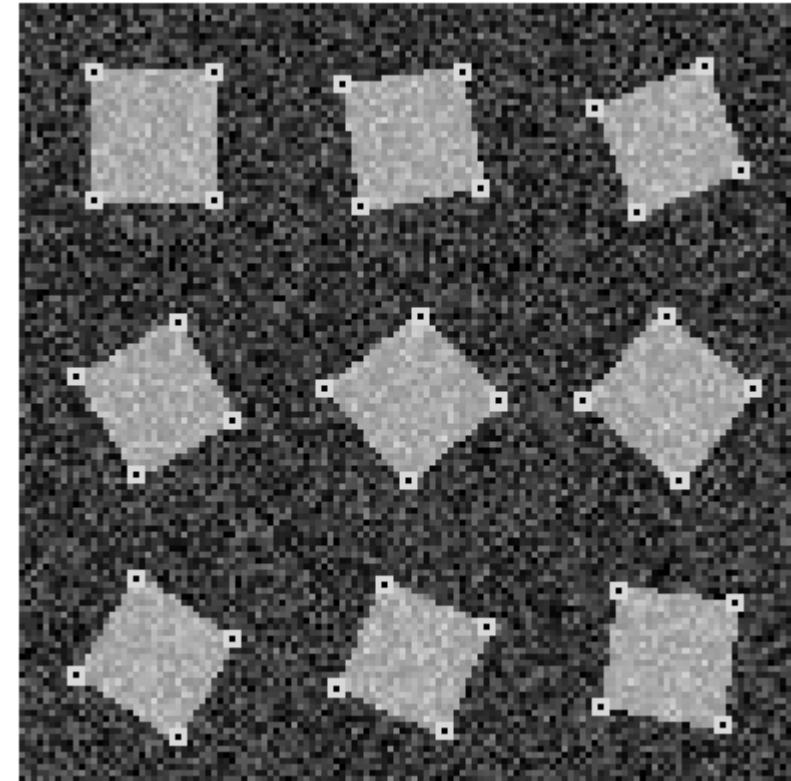
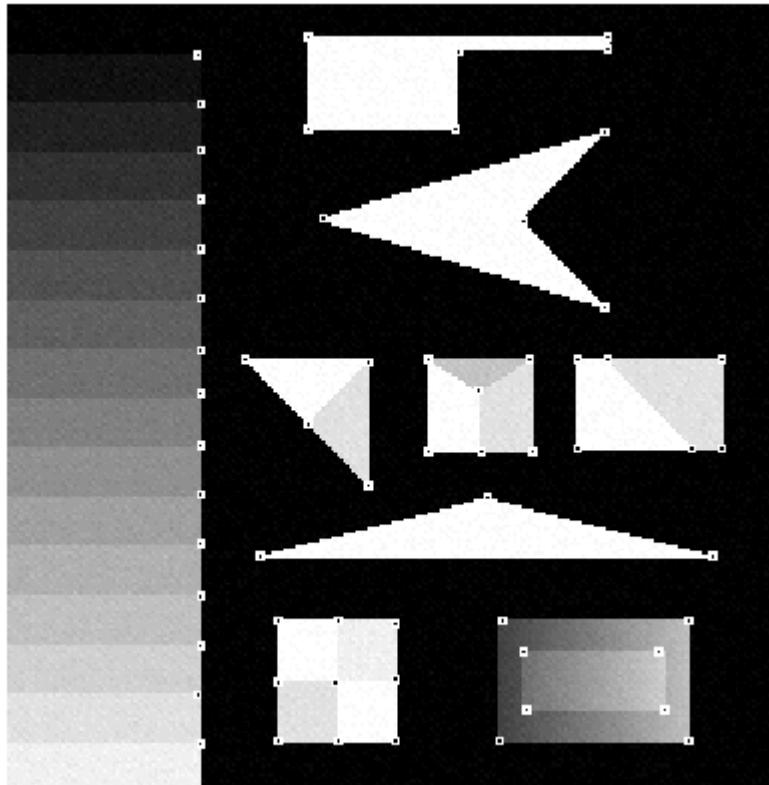
$$\bar{\vec{r}}(\vec{r}_0) = \frac{\sum_{\vec{r}} \vec{r} c(\vec{r}, \vec{r}_0)}{\sum_{\vec{r}} c(\vec{r}, \vec{r}_0)}.$$

Centro de masas
del USAN



Operador SUSAN

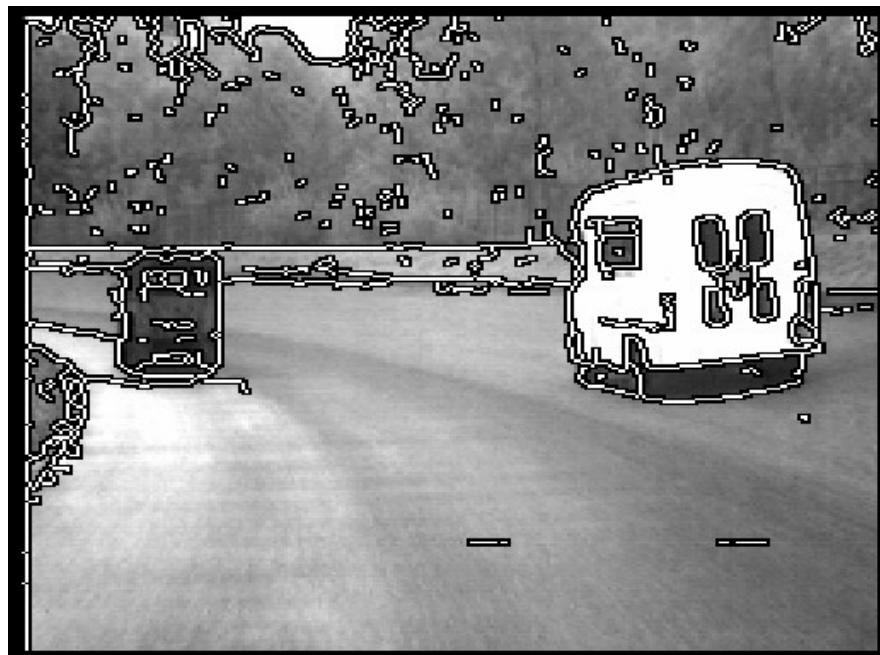
- Ejemplos de detección (imágenes sintéticas con ruido):



Operador SUSAN

- Más información y detalles:

<http://users.fmrif.ox.ac.uk/~steve/susan/>



Bordes



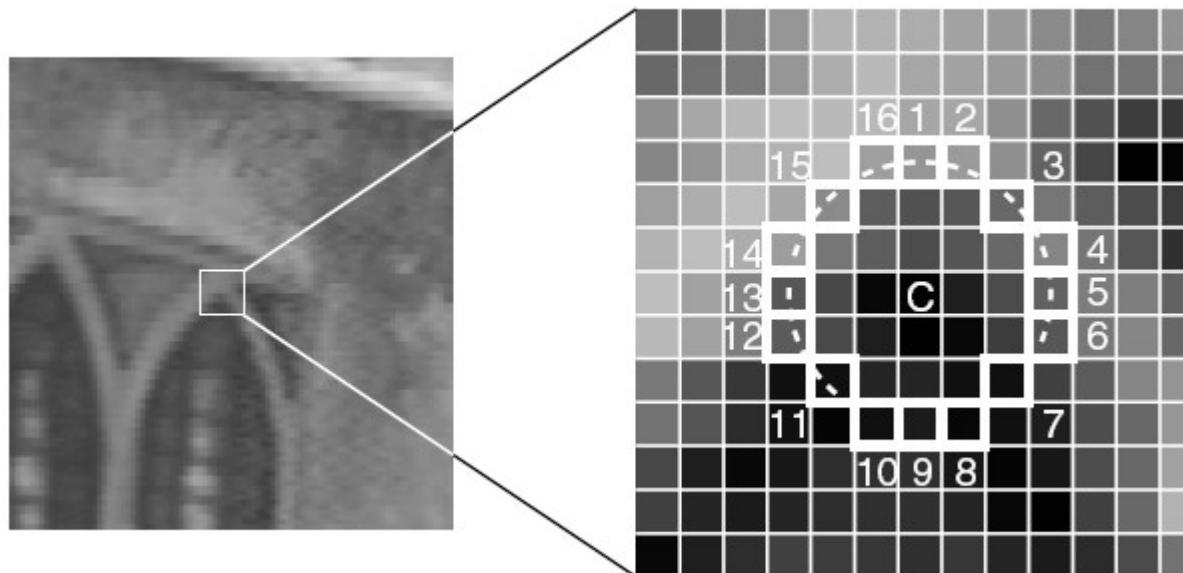
Puntos

Más detectores de esquinas: el operador FAST

- FAST: *Features from Accelerated Segment Test.*
- Se examina el círculo de Bressenham centrado en torno a un píxel (radio 3, 16 píxeles).
- Si la intensidad de al menos 12 píxeles contiguos de esos 16 está siempre un determinado umbral por encima (o por debajo) del valor de la intensidad en el centro de la máscara, se considera que hay una esquina.
- El orden de escaneo no es circular, sino, por eficiencia: N, S, E y W primero, y si al menos tres de estos píxeles no cumplen la condición, el píxel se descarta sin más procesamiento.

Operador FAST

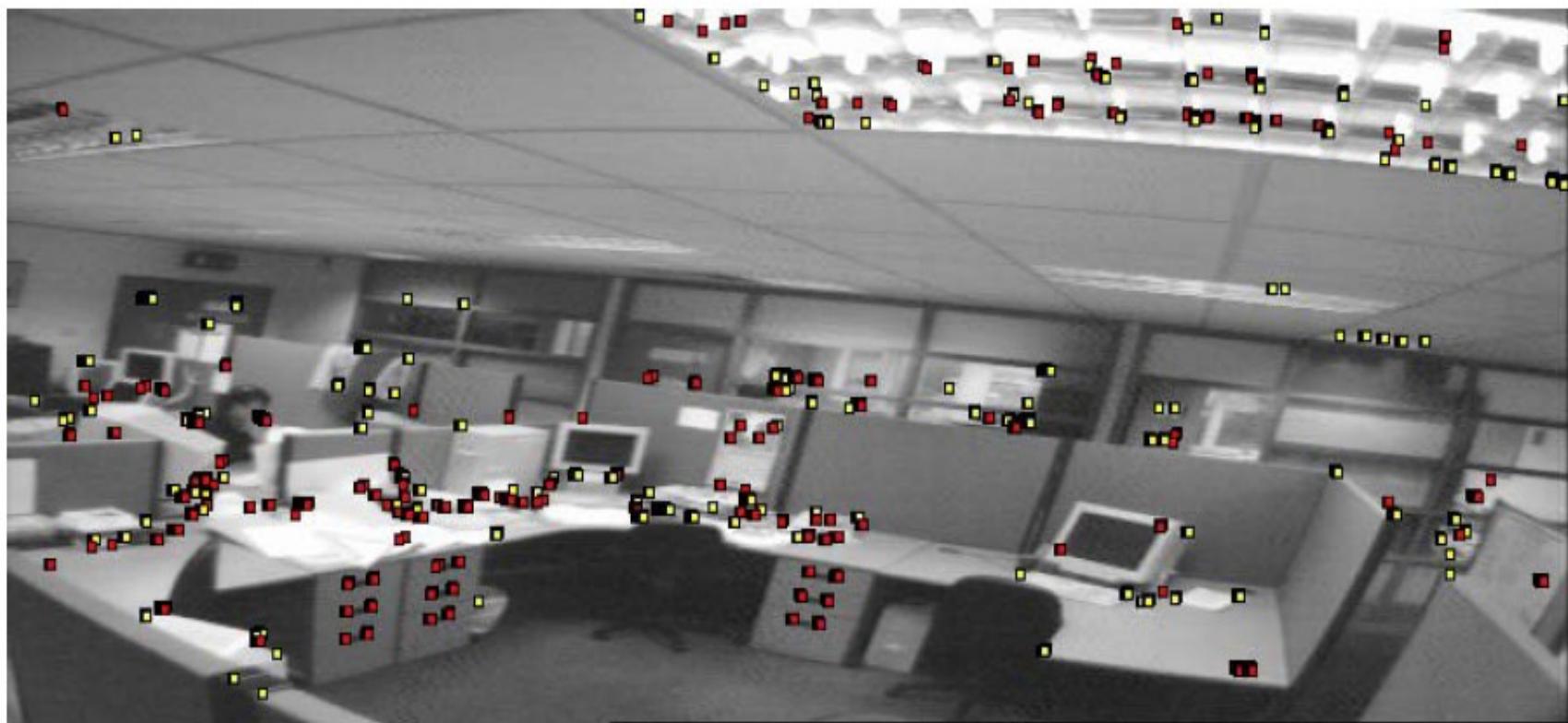
- Ilustración del método:



- El método es muy rápido (hasta 1 ms para una imagen tamaño PAL desentrelazada 768x576).

Operador FAST

- Ejemplo de funcionamiento:



Puntos rojos: Esquinas “positivas”

Puntos amarillos: Esquinas “negativas”

Apéndice: Extracción de segmentos (grupo PARP-MoVi, UMU)

- Preprocesamiento:
 - Color a BW, filtrado (mediana, gaussiano) , filtro paso-alto.
- Estimación de orientación de bordes
 - Conjunto de máscaras de orientación local.
 - Respuesta no lineal, efecto amortiguador del ruido.
- Agrupamiento:
 - Histéresis para mejorar respuesta (similar a Canny).
- Postprocesamiento:
 - Limpieza y marcado de la imagen de bordes.
 - Estimación robusta del color de los lados.

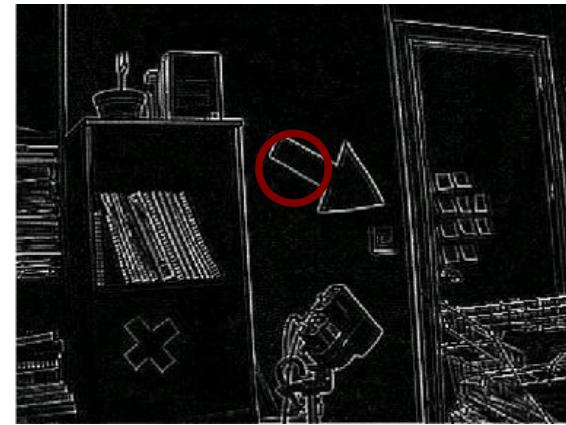
Esquema general de funcionamiento (1/2)



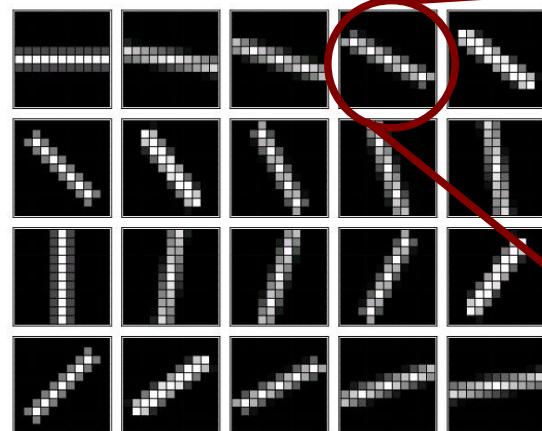
Imagen original

$$\begin{array}{ccc} -1 & 0 & +1 \\ -1 & -1 & -1 \\ 0 & -1 & +8 & -1 \\ +1 & -1 & -1 & -1 \end{array}$$

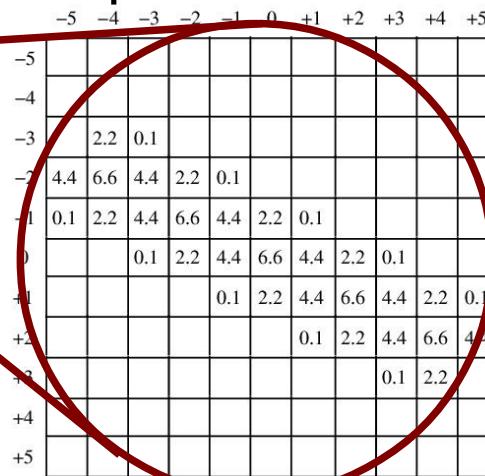
$$I^{brd} = |h * I^{gray}|$$



Filtro
paso-alto



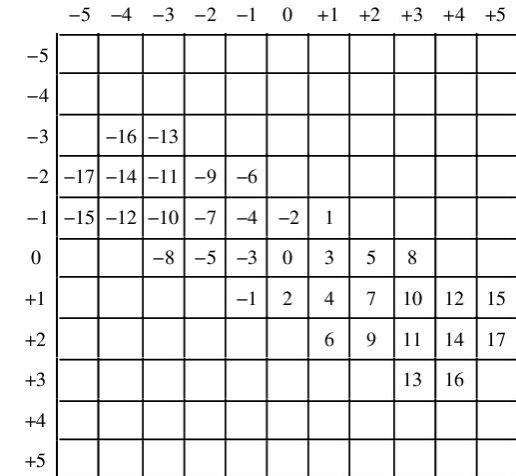
Máscaras de
orientación



A 13x13 grid of values representing a detailed orientation mask. The values range from -16 to 15. A red circle highlights the central value of 1.0 at position (0,0).

-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5
-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5
-4	-3	2.2	0.1							
-3	4.4	6.6	4.4	2.2	0.1					
-2	0.1	2.2	4.4	6.6	4.4	2.2	0.1			
-1			0.1	2.2	4.4	6.6	4.4	2.2	0.1	
0				0.1	2.2	4.4	6.6	4.4	2.2	0.1
+1					0.1	2.2	4.4	6.6	4.4	2.2
+2						0.1	2.2	4.4	6.6	4.4
+3							0.1	2.2	4.4	6.6
+4								0.1	2.2	4.4
+5									0.1	2.2

Máscaras de orienta-
ción (detalle)



A 13x13 grid of values representing a grouping matrix. The values range from -16 to 15. A red circle highlights the central value of 1.0 at position (0,0).

-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5
-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5
-4	-3	-16	-13							
-3	-16	-14	-11	-9	-6					
-2	-14	-17	-11	-9	-6					
-1	-11	-15	-12	-10	-7	-4	-2	1		
0	-7	-8	-5	-3	0	3	5	8		
+1			-1	2	4	7	10	12	15	
+2				6	9	11	14	17		
+3					13	16				
+4										
+5										

Agrupamiento
(histéresis)

Esquema general de funcionamiento (2/2)

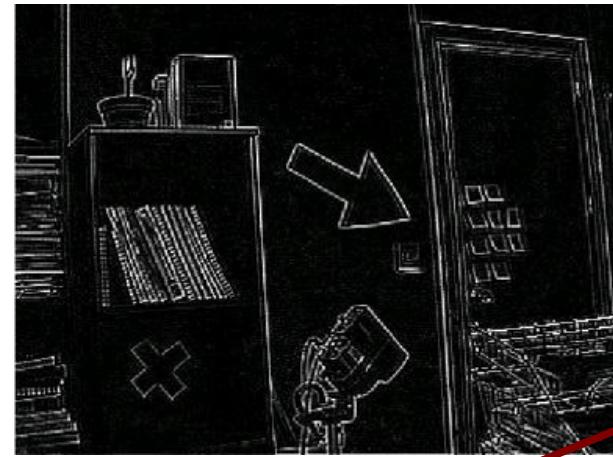
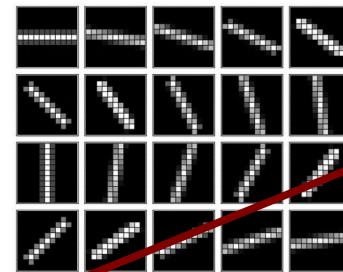
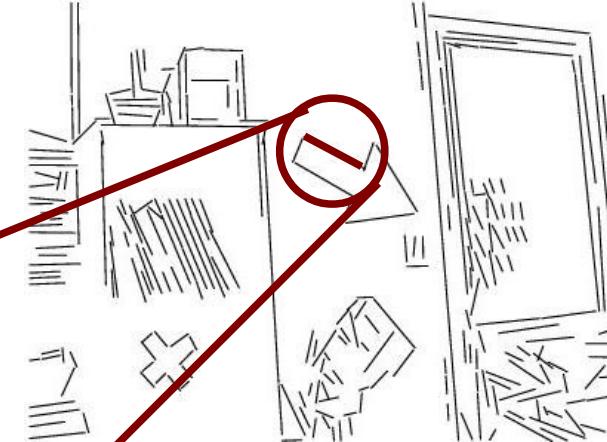


Imagen de bordes



Saliente y agrupamiento

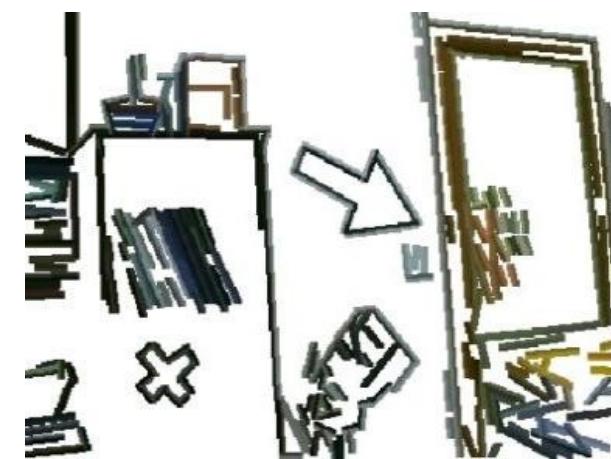


Segmentos (información geométrica)

A 12x12 grid representing a color histogram or sampling matrix. The x-axis is labeled from 0 to 13. The y-axis is labeled from 0 to 11. A red circle highlights a peak in the distribution at approximately (5, 5).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0,0	0,1												
1	2,0	1,0	1,1	0,2	0,3									
2	3,0	2,1	2,2	1,2	1,3	0,4	0,5							
3	5,0	4,0	3,1	3,2	2,3	2,4	1,4	1,5	0,6	0,7				
4	6,0	5,1	4,1	4,2	3,3	3,4	2,5	2,6	1,6	1,7	0,8			
5	6,1	5,2	5,3	4,3	4,4	3,5	3,6	2,7	2,8	1,8	0,9			
6	6,2	6,3	5,4	5,5	4,5	4,6	3,7	3,8	2,9	1,9				
7			6,4	6,5	5,6	5,7	4,7	4,8	3,9					
8				6,6	6,7	5,8	5,9	4,9						
9					6,8	6,9								
10														
11														

Limpieza y muestreado de color



Segmentos con información de color

Resultados

Original:



Segmentos:

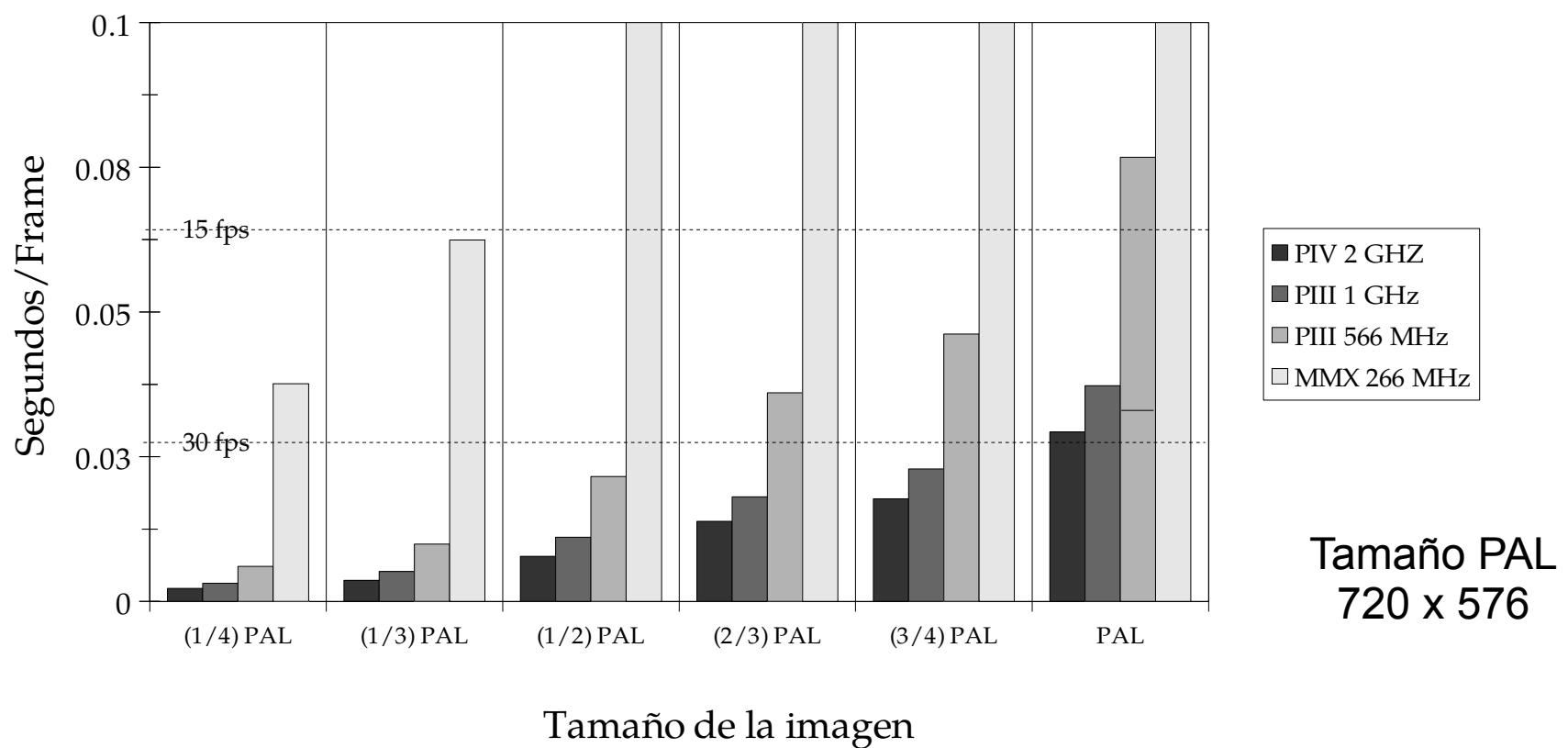


Reconstrucción
(proceso difusión):



Rendimiento

- Eficiencia de cómputo (tiempo real, año 2003):
 - Complejidad lineal con el tamaño de la imagen, dependiente de la cantidad de detalles



Referencias

- Algunos notebooks de Python+OpenCV:
 - Bordes: gradiente, operador de Canny, difusión de color: [←](#) [←](#)
 - Covarianza, direcciones principales: [←](#)
- Tutorial Python+OpenCV (secciones relevantes):
 - Features (discusión general): [←](#)
 - Operador de Canny: [←](#)
 - Detector de Harris: [←](#)
 - Variante Shi-Tomasi: [←](#)
 - Speeded Up Robust Features (SURF): [←](#)
 - Features from Accelerated Segment Test (FAST): [←](#)