DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Procesamiento y Análisis de Imágenes

Violeta Chang

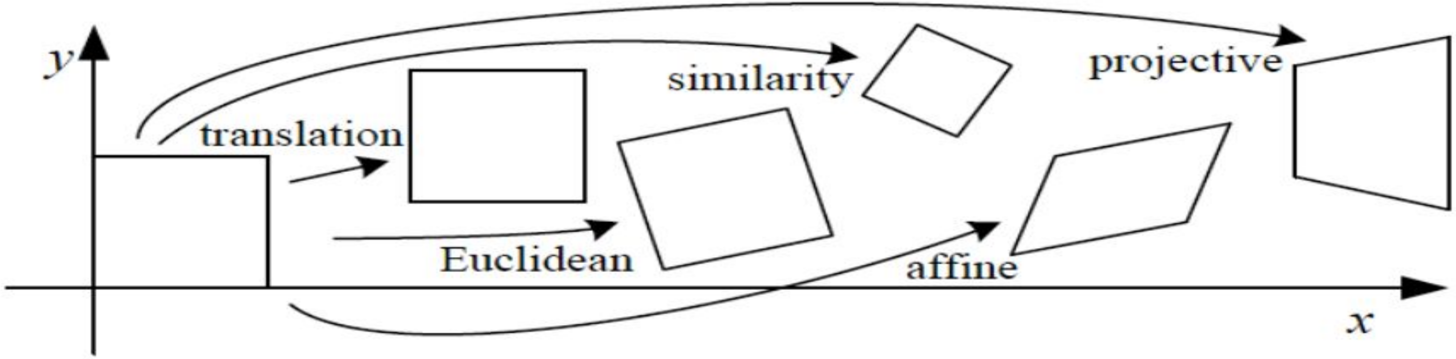
violeta.chang@usach.cl

Créditos por slides: José M. Saavedra, Aaron Bobick, Fei-Fei Li, Juan Carlos Niebles, Ranjay Krishna

Detección de puntos de interés

The basic image point matching problem

- Suppose I have two images related by some transformation. Or have two images of the same object in different positions.
- How to find the transformation of image 1 that would align it with image 2?



Detección de puntos de interés




Detección de puntos de interés

We want *Local*(1) *Features*(2)

- Goal: Find points in an image that can be:
  - Found in other images
  - Found precisely – well localized
  - Found reliably – well matched
- Why?
  - Want to compute a fundamental matrix to recover geometry
  - Robotics/Vision: See how a bunch of points move from one frame to another. Allows computation of how camera moved -> depth -> moving objects
  - Build a panorama…

Detección de puntos de interés

Suppose you want to build a panorama

Detección de puntos de interés

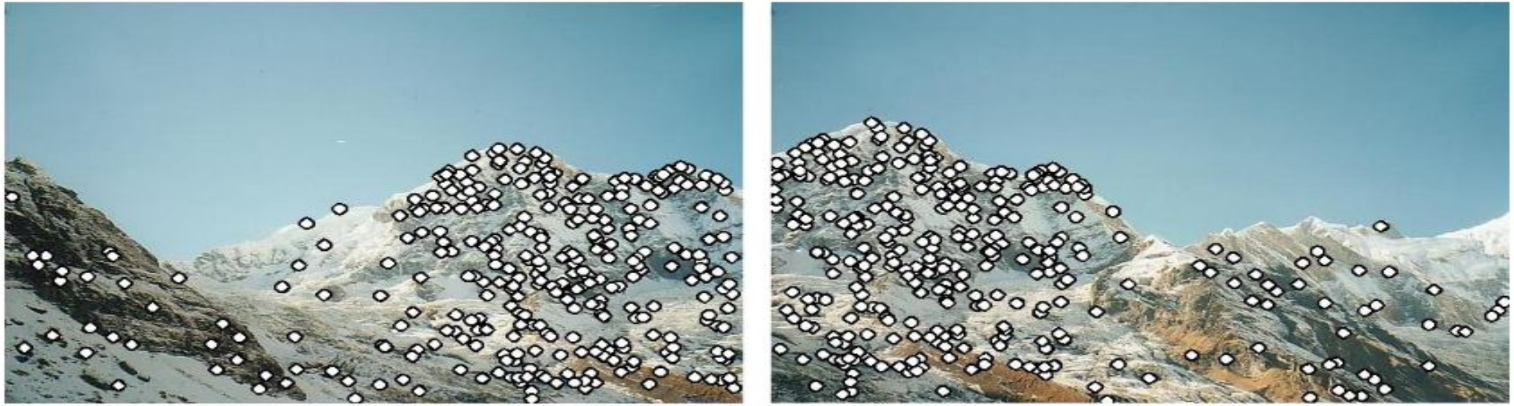
How do we build panorama?

- We need to match (align) images

Detección de puntos de interés

Matching with Features

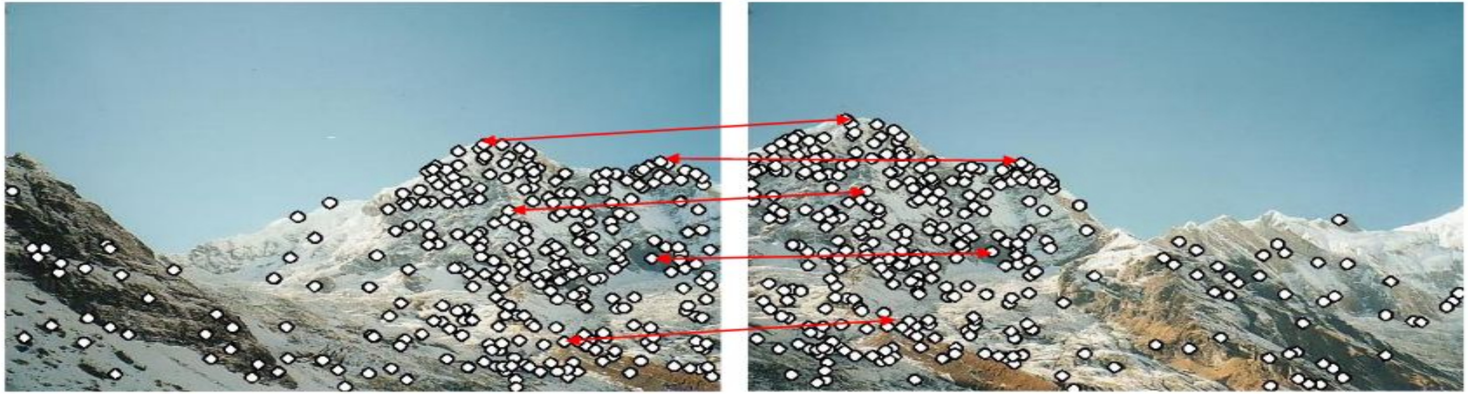
- Detect features (feature points) in both images



Detección de puntos de interés

Matching with Features

- Detect features (feature points) in both images
- Match features - find corresponding pairs



Detección de puntos de interés

Matching with Features

- Detect features (feature points) in both images
- Match features - find corresponding pairs
- Use these pairs to align images

Detección de puntos de interés

Matching with Features

- Problem 1:
  - Detect the *same* point *independently* in both images

no chance to match!

We need a repeatable detector

Detección de puntos de interés

Matching with Features

- Problem 2:
  - For each point correctly recognize the corresponding one

We need a reliable and distinctive ***descriptor***

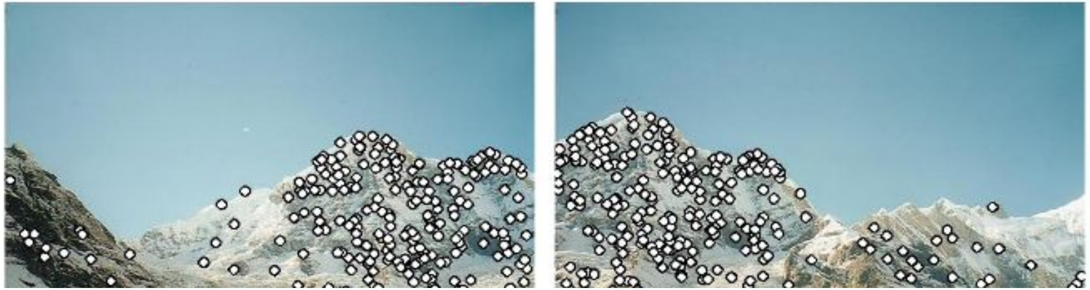
Detección de puntos de interés

More motivation…

- Feature points are used also for:
  - Image alignment (e.g. homography or *fundamental* matrix)
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation
  - … other

Detección de puntos de interés

Characteristics of good features

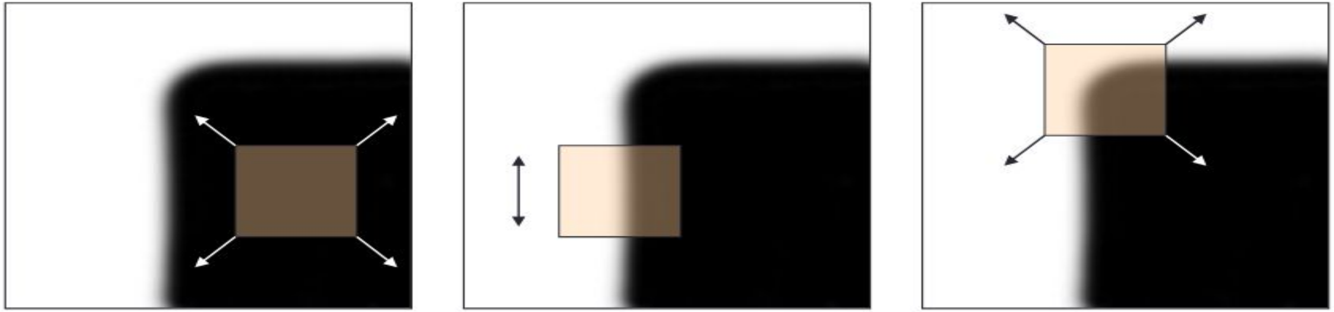


- Repetibilidad y Precisión
  - Invariante a traslación, rotación y cambios de escala.
  - Robusto a transformaciones afines y de perspectiva.
  - Robusto a variaciones de luminosidad, ruido, blur.
- Localidad: Calculados en forma local → robusto a oclusión.
- Cantidad: Gran cantidad de descriptores → objeto cubierto.
- Distintivo: Deben contener características discriminativas.
- Eficiencia: Para ser aplicado en tiempo real.

Detección de puntos de interés

Corner Detection: Basic Idea

- We should easily recognize the point by looking through a small window
- Shifting a window in any direction should give a large change in intensity



"flat" region:
no change in
all directions

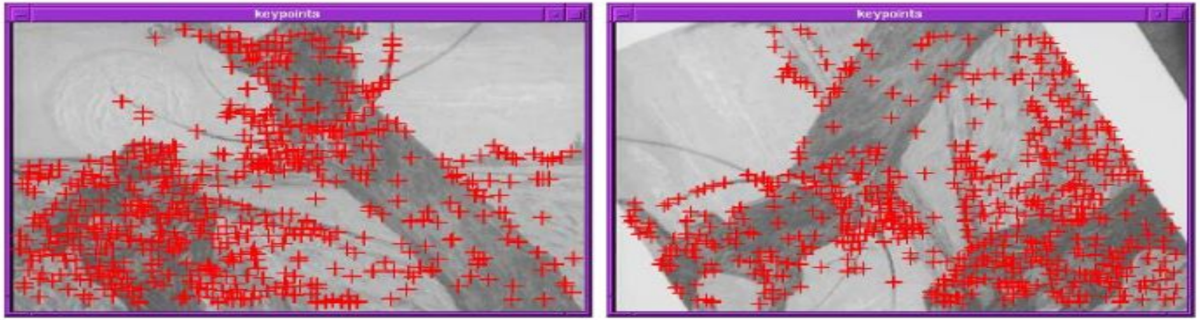
"edge":
no change
along the edge
direction

"corner":
significant change
in all directions with
small shift

Source: A. Efros

Detección de puntos de interés

Finding *Corners*



- Key property: in the region around a corner, image gradient has two or more dominant directions
- Corners are repeatable and distinctive

C.Harris and M.Stephens. "A Combined Corner and Edge Detector.“ *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, **1988**

Detección de puntos de interés

Corner Detection: Mathematics

Change in appearance for the shift $[u,v]$:

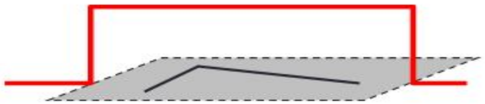
$$\boxed{E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2}$$

Window function

Shifted intensity

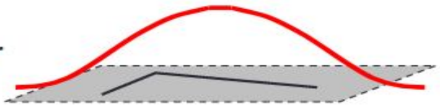
Intensity

Window function $w(x,y) =$



1 in window, 0 outside

or



Gaussian

Source: R. Szeliski

Detección de puntos de interés

Change of intensity for the shift $[u,v]$:

$$E(u,v) = \sum_{x,y} w(x,y) \left[I(x+u, y+v) - I(x,y) \right]^2$$

Window function

Shifted intensity

Intensity

For nearly constant patches, this will be near 0.
For very distinctive patches, this will be larger.
Hence... we want patches where E(u,v) is LARGE.

Detección de puntos de interés

Harris Detector Formulation

- This measure of change can be approximated by:

$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

where M is a 2×2 matrix computed from image derivatives:

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

Gradient with respect to x, times gradient with respect to y

Sum over image region – the area we are checking for corner

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y]$$

Detección de puntos de interés

Harris Detector Formulation

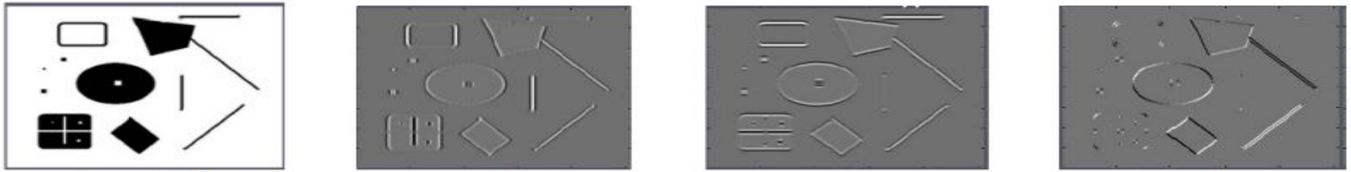


Image I $\quad I_x \quad I_y \quad I_x I_y$

where M is a 2×2 matrix computed from image derivatives:

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

Gradient with respect to x, times gradient with respect to y

Sum over image region – the area we are checking for corner

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y]$$

Detección de puntos de interés

Interpreting the second moment matrix

First, consider the axis-aligned case where gradients are either horizontal or vertical

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

If either λ is close to 0, then this is **not** a corner, so look for locations where both are large.

Detección de puntos de interés

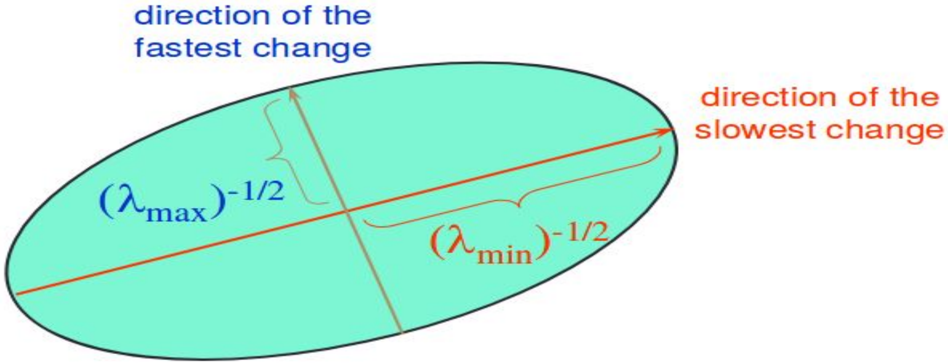
Interpreting the second moment matrix

Consider a horizontal "slice" of $E(u, v)$: $[u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$

This is the equation of an ellipse.

Diagonalization of M: $M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$

The axis lengths of the ellipse are determined by the eigenvalues and the orientation is determined by *R*



Detección de puntos de interés

Intuitive way to understand Harris

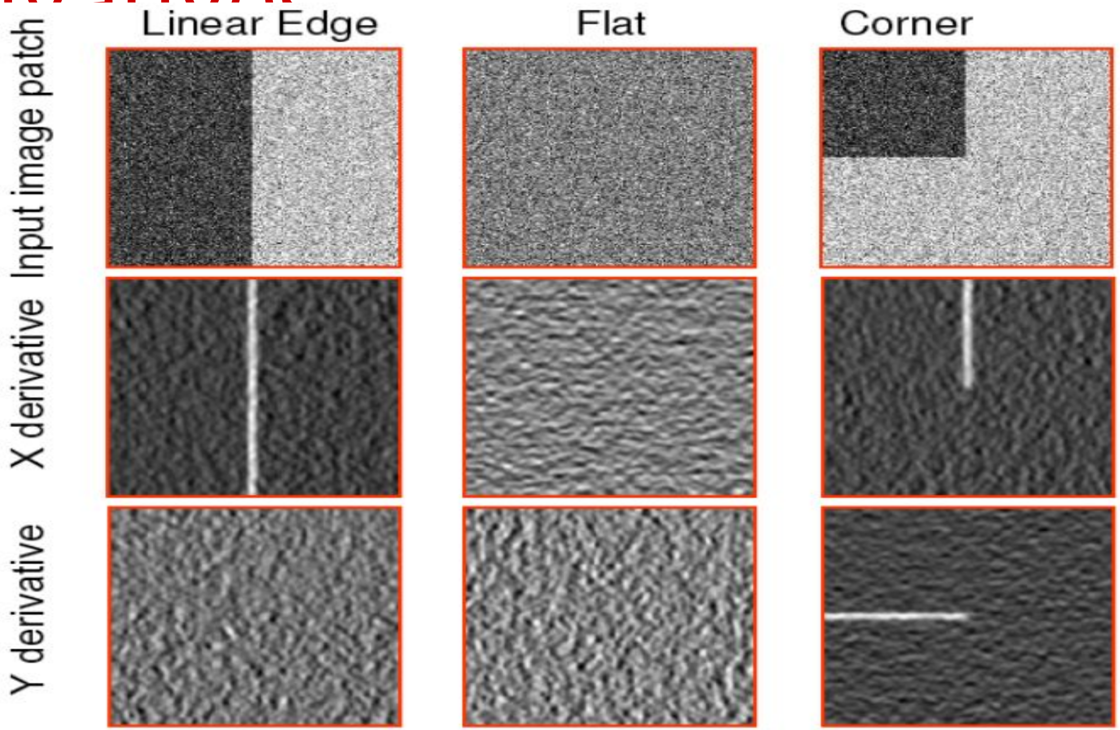
Treat gradient vectors as a set of (dx,dy) points with a center of mass defined as being at (0,0).

Fit an ellipse to that set of points via scatter matrix

Analyze ellipse parameters for varying cases…

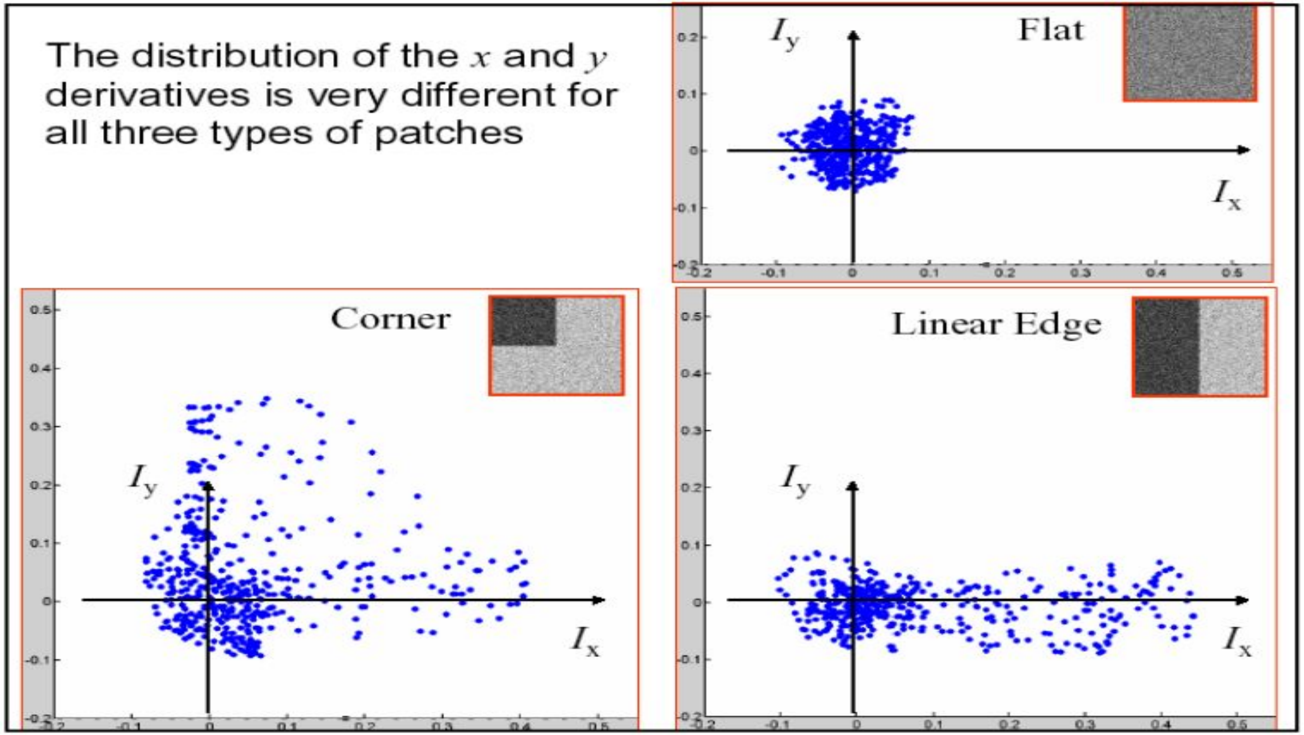
Detección de puntos de interés

Example: Cases and 2D derivatives



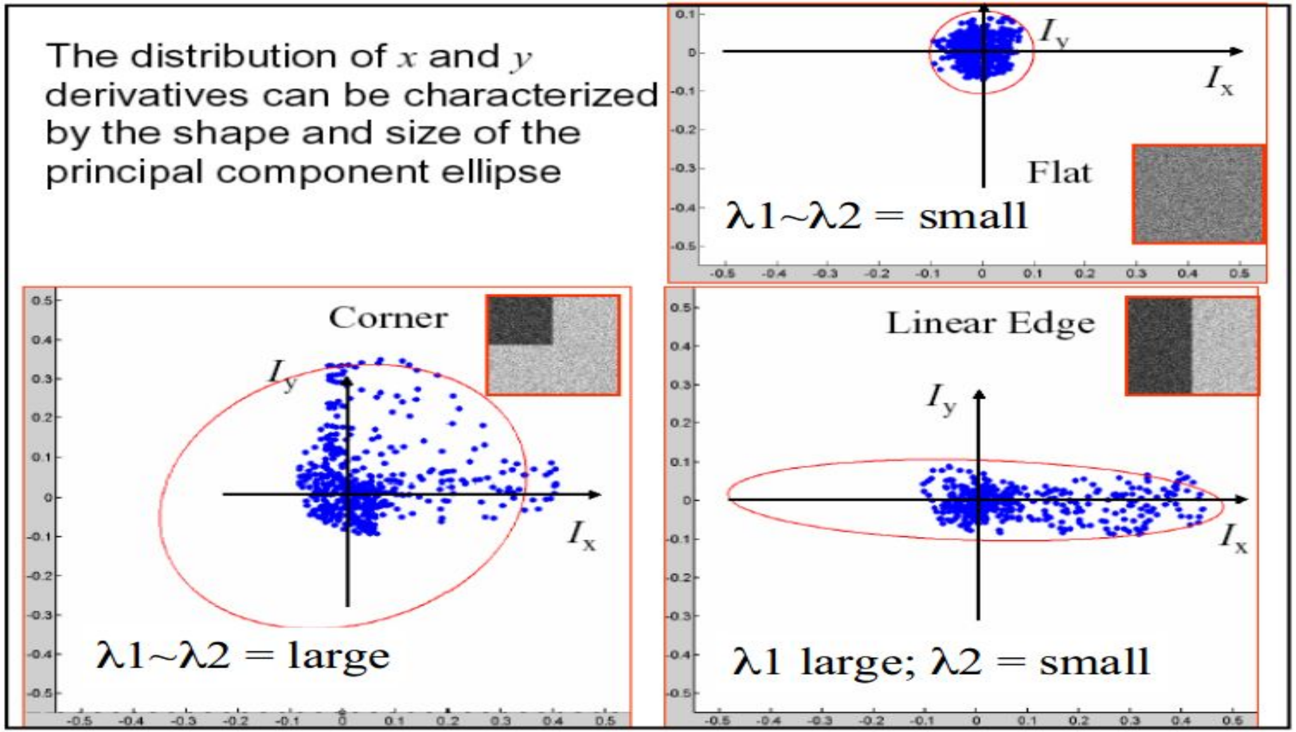
Detección de puntos de interés

Plotting derivatives as 2D points



Detección de puntos de interés

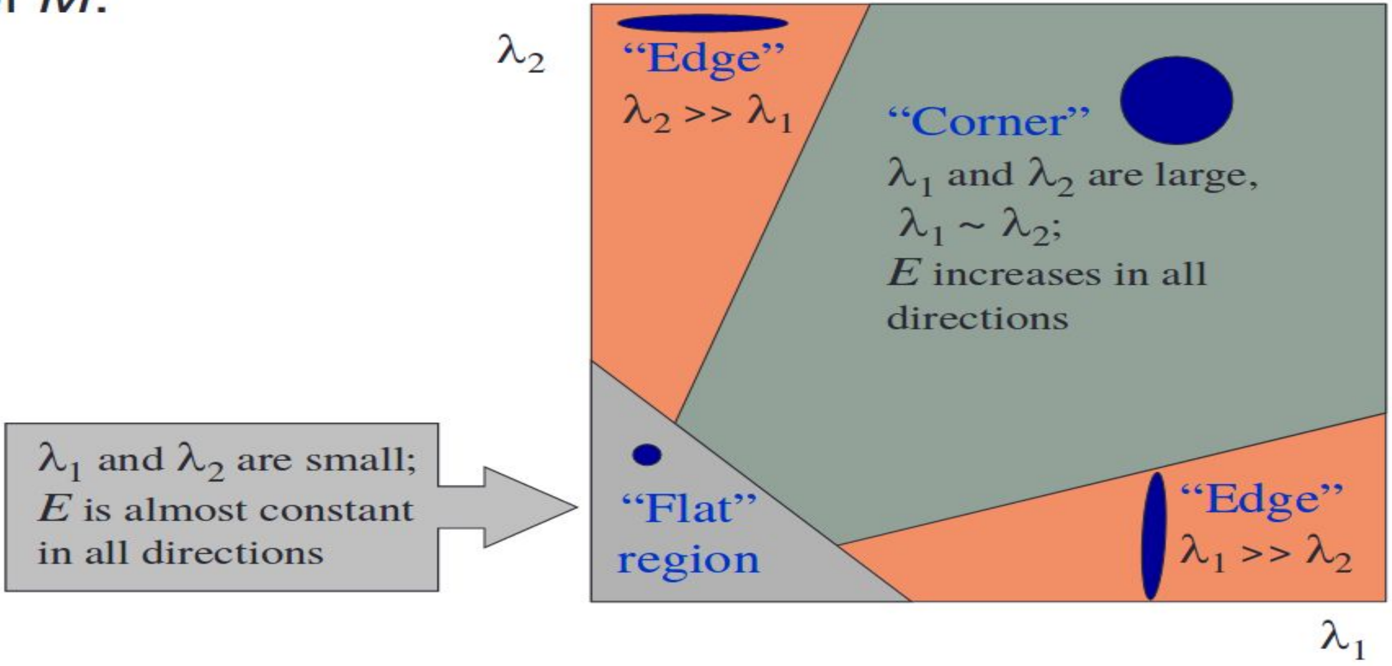
Plotting derivatives as 2D points



Detección de puntos de interés

Interpreting the eigenvalues

Classification of image points using eigenvalues of *M*:



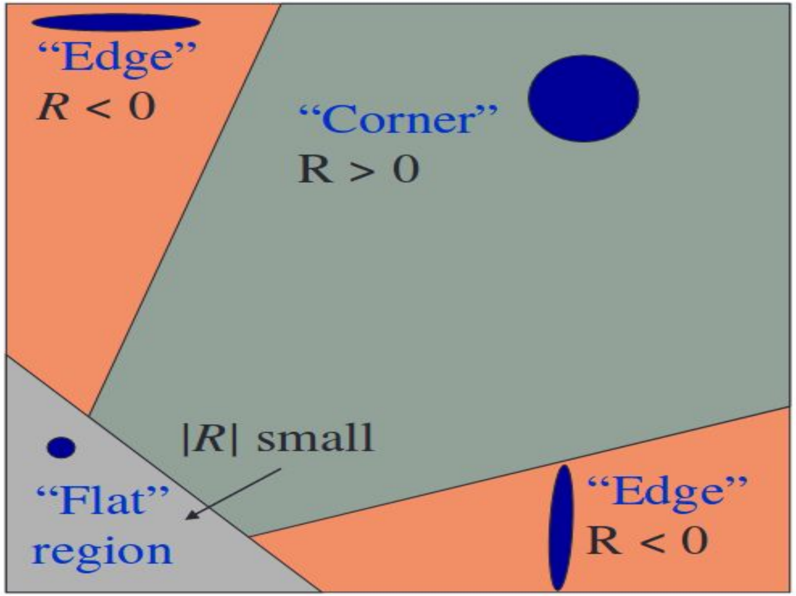
Detección de puntos de interés

Harris corner response function

$$R = \det(M) - \alpha \, \mathrm{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

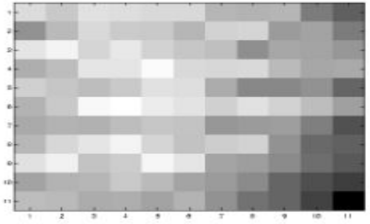
α: constant (0.04 to 0.06)

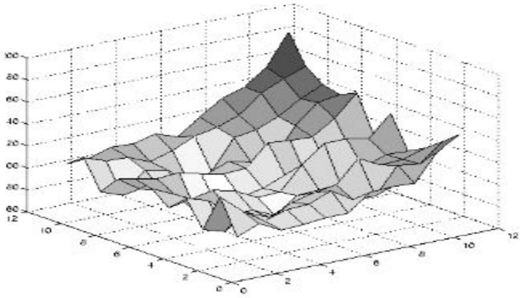
- R depends only on eigenvalues of M, but don't compute them (no sqrt, so really fast!
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region



Detección de puntos de interés

Low texture region





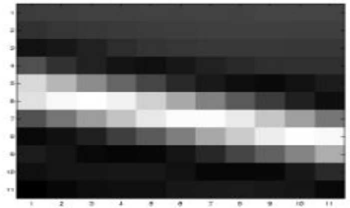
$$\sum \nabla I (\nabla I)^T$$

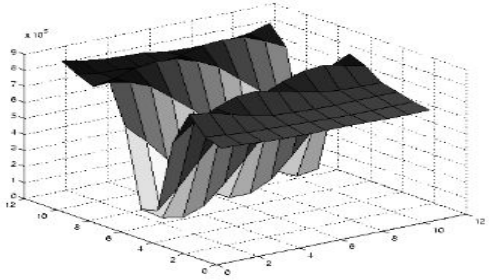
– gradients have small magnitude

– small λ_1, small λ_2

Detección de puntos de interés

Edge





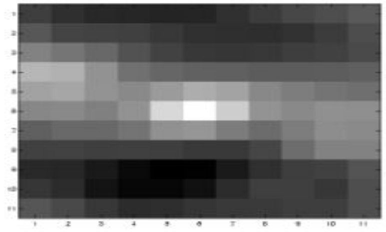
$$\sum \nabla I (\nabla I)^T$$

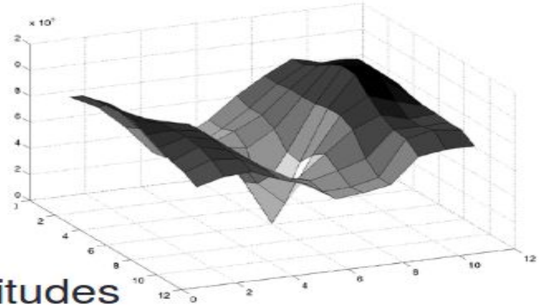
– large gradients, all the same

– large λ_1, small λ_2

Detección de puntos de interés

High textured region





$$\sum \nabla I (\nabla I)^T$$

– gradients are different, large magnitudes
– large λ_1, large λ_2

Detección de puntos de interés

Harris detector: Algorithm

1. Compute Gaussian derivatives at each pixel
2. Compute second moment matrix M in a Gaussian window around each pixel
3. Compute corner response function R
4. Threshold R
5. Find local maxima of response function (nonmaximum suppression)

C.Harris and M.Stephens. "A Combined Corner and Edge Detector.“ *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

Detección de puntos de interés

- Aplique el algoritmo de Harris para determinar si la siguiente región corresponde una zona plana, a un borde o a una esquina.

Input Image:

0	0	1	4	9
1	0	5	7	11
1	4	9	12	16
3	8	11	14	16
8	10	15	16	20

differentiation kernels:

-1	0	1	d/dx

-1	
0	d/dy
1	

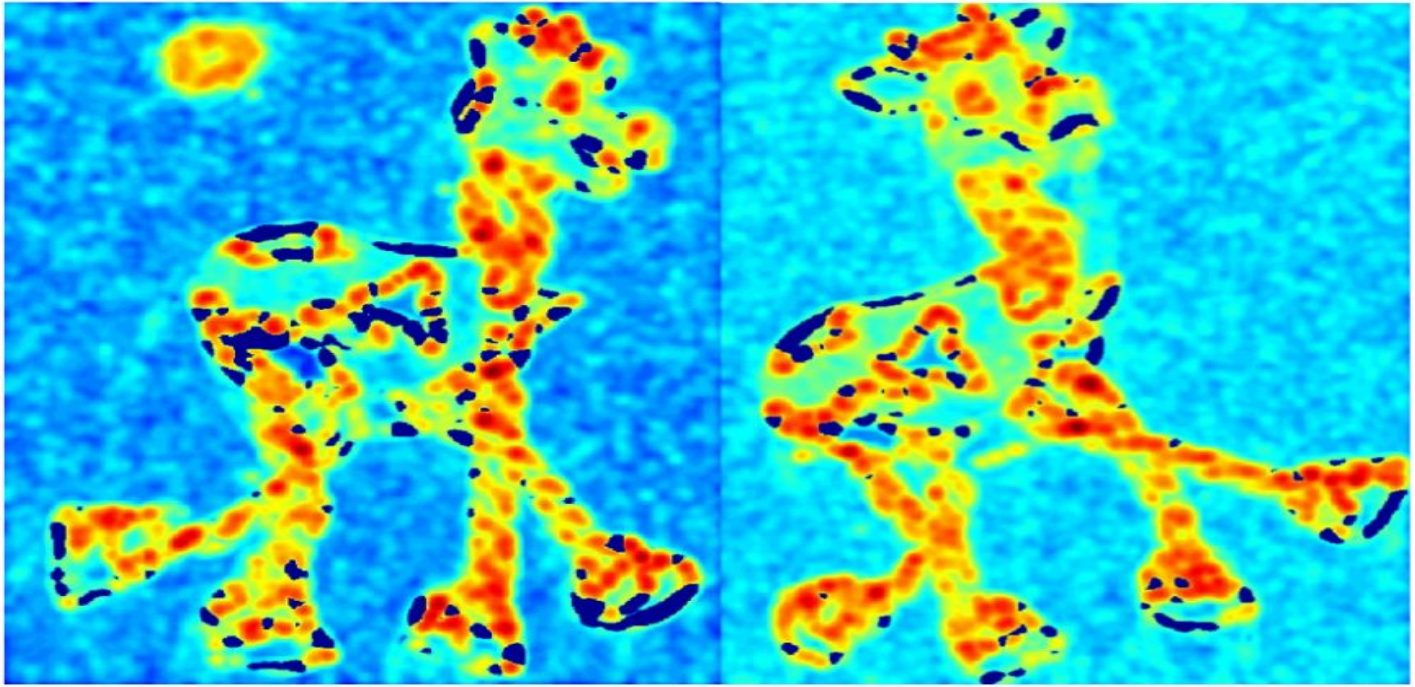
Detección de puntos de interés

Harris Detector: Workflow

Detección de puntos de interés

Harris Detector: Workflow

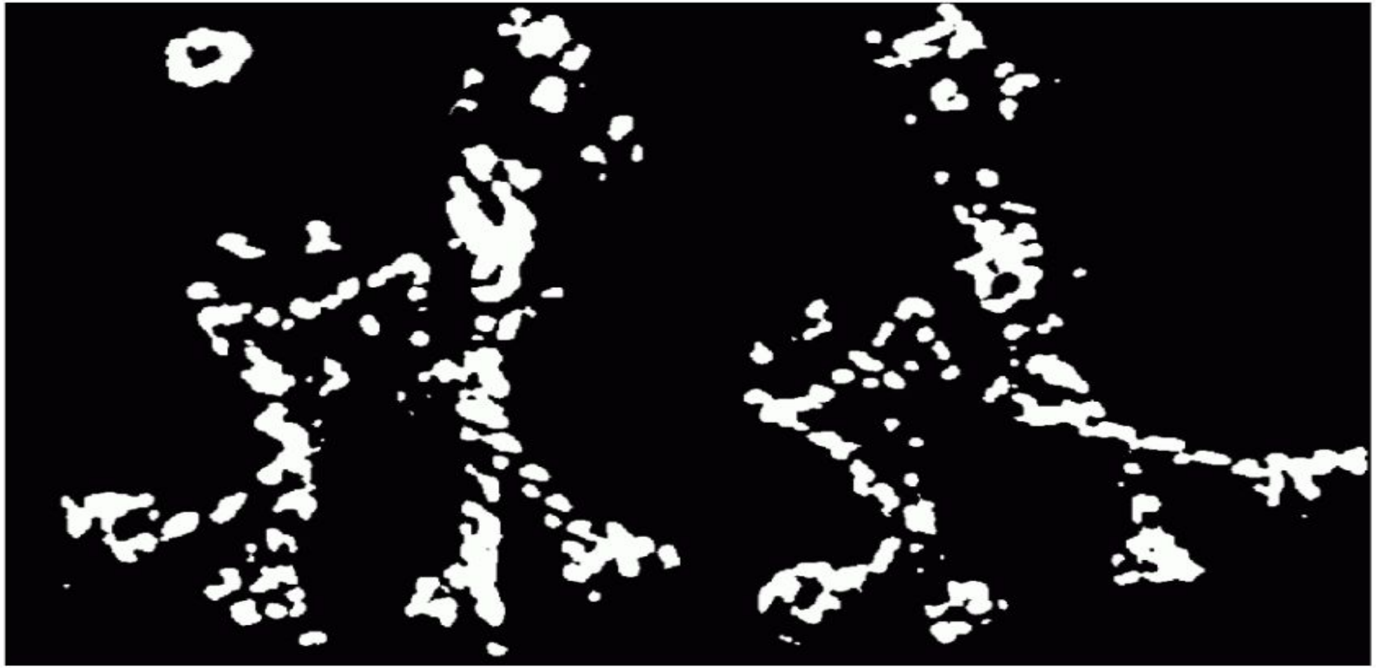
Compute corner response R



Detección de puntos de interés

Harris Detector: Workflow

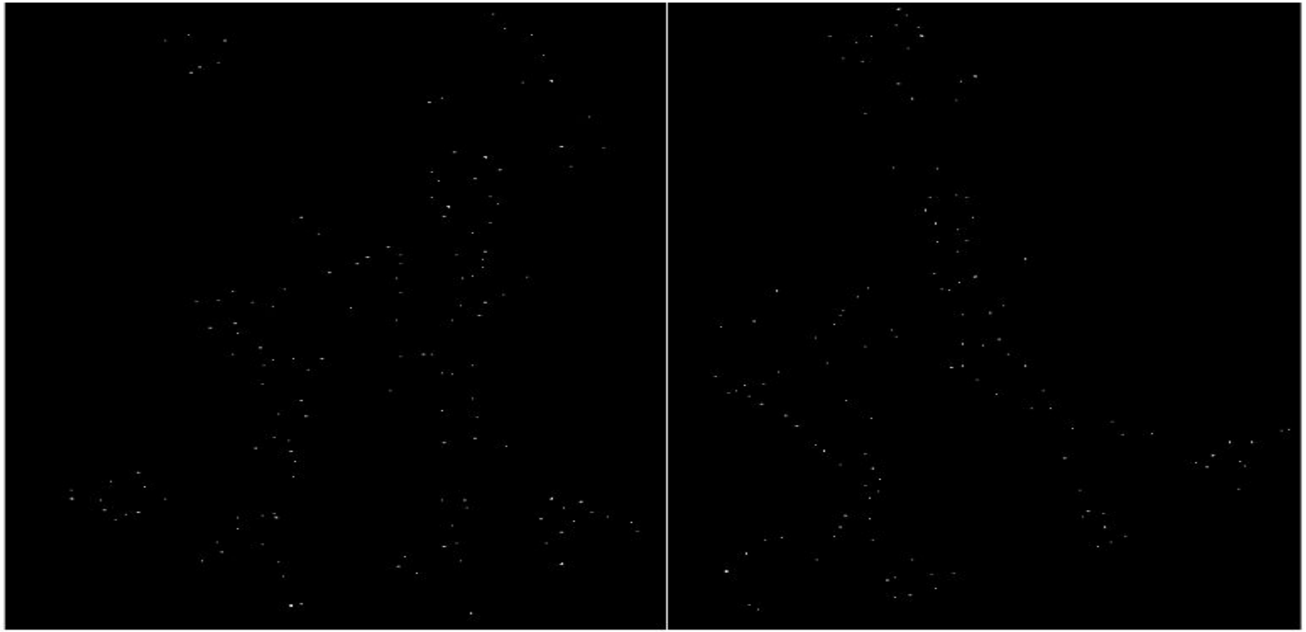
Find points with large corner response: R>threshold



Detección de puntos de interés

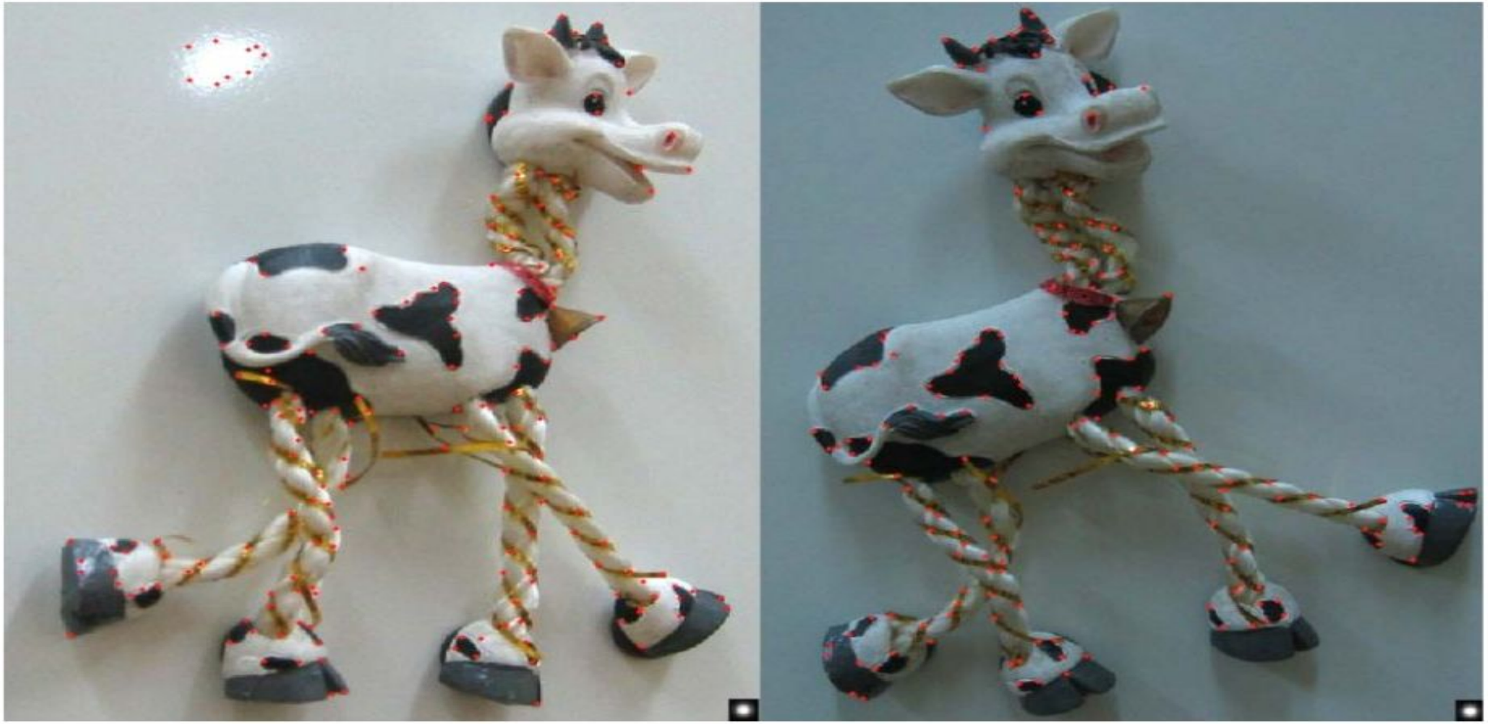
Harris Detector: Workflow

Take only the points of local maxima of R



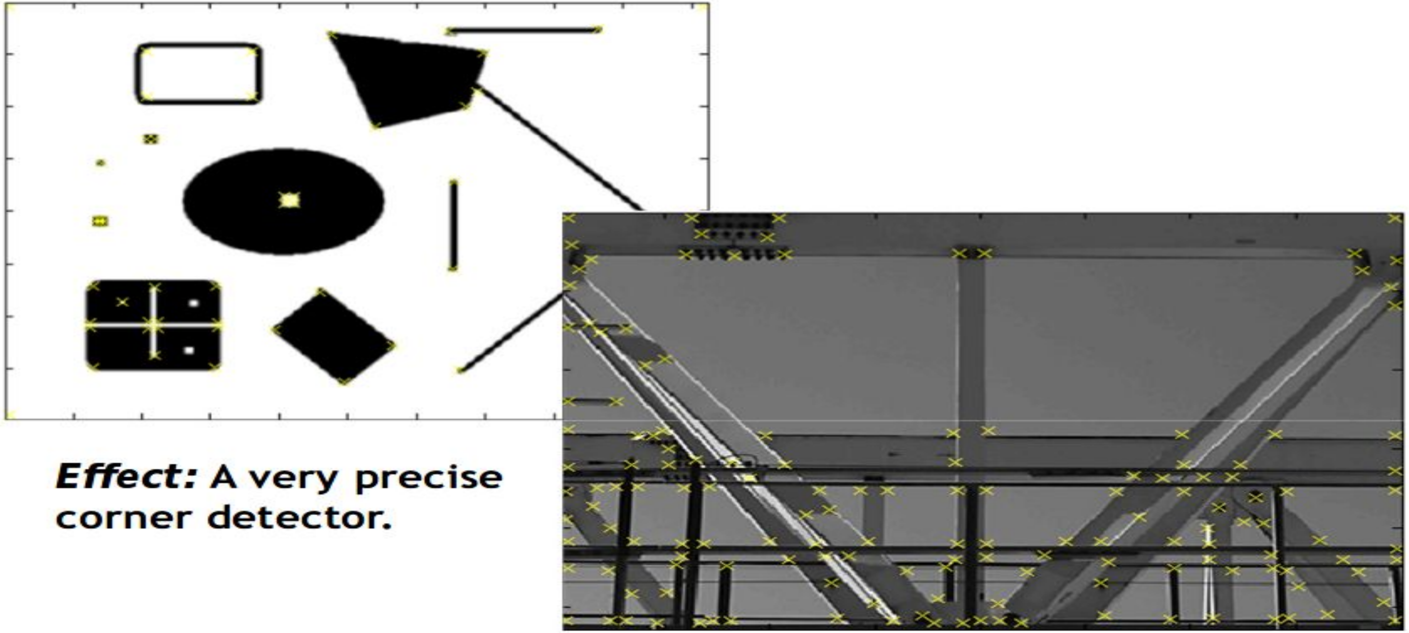
Detección de puntos de interés

Harris Detector: Workflow



Detección de puntos de interés

Harris Detector – Responses [Harris88]



Effect: A very precise corner detector.

Detección de puntos de interés

Harris Detector – Responses [Harris88]

Detección de puntos de interés

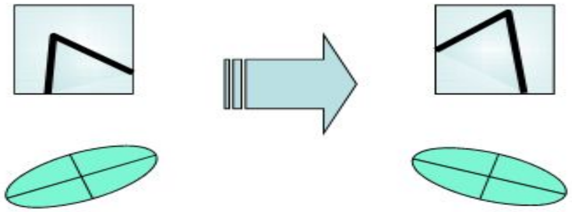
Harris Detector – Responses [Harris88]

- Results are well suited for finding stereo correspondences

Detección de puntos de interés

Harris Detector: Properties

- Rotation invariance?



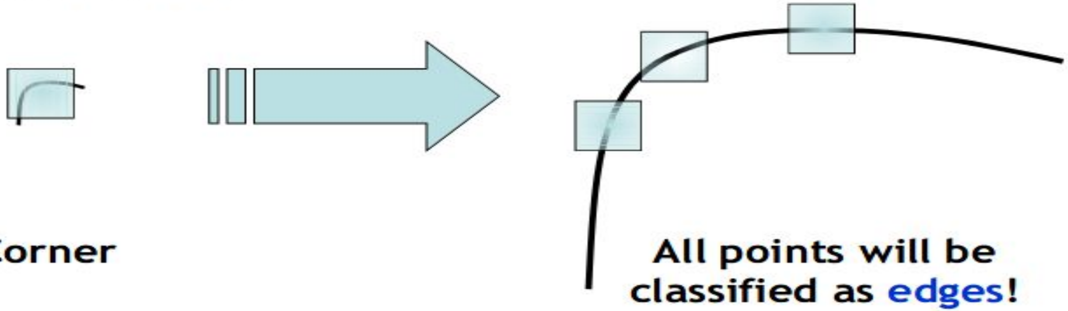
Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner response R **is invariant to image rotation**

Detección de puntos de interés

Harris Detector: Properties

- Rotation invariance
- Scale invariance?



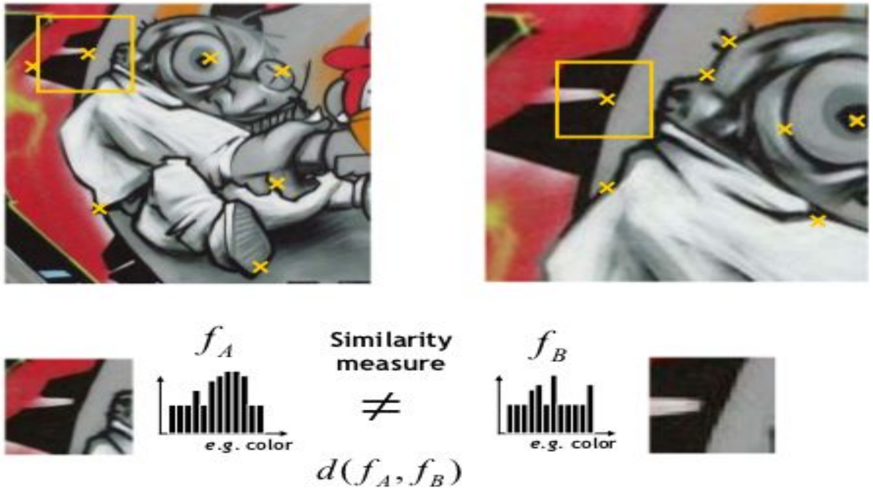
Not invariant to image scale!

Detección de regiones de interés

- Cuando se detectan puntos de interés (esquinas), la escala de la ventana usada puede cambiar las esquinas que se detecten

Detección de regiones de interés

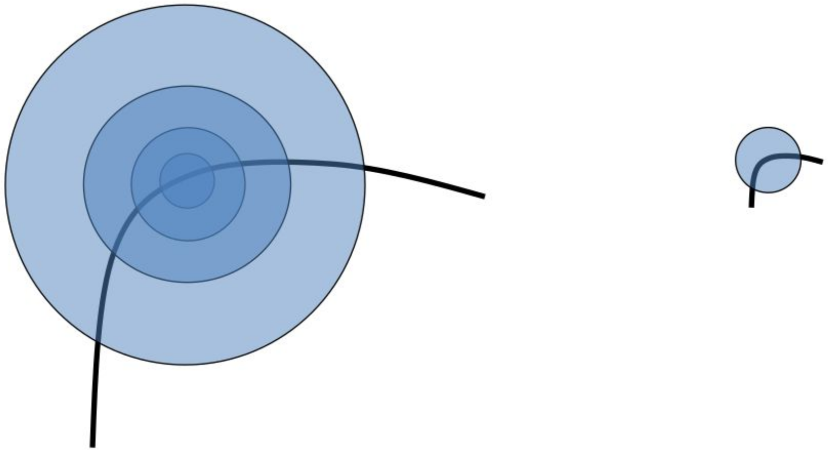
Necesidad de detectar regiones de interés invariantes a escala



Detección de regiones de interés

Scale Invariant Detection

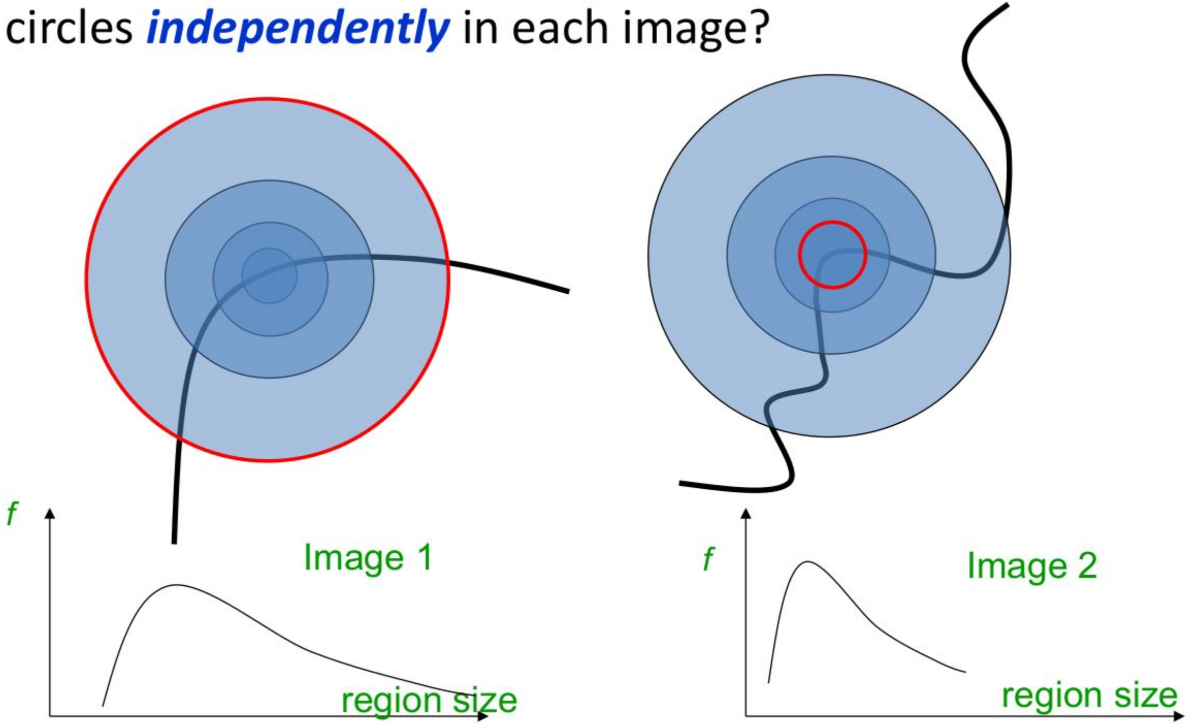
- Consider regions (e.g. circles) of different sizes around a point
- What region size do we choose, so that the regions look the same in both images?



Detección de regiones de interés

Scale Invariant Detection

- The problem: how do we choose corresponding circles ***independently*** in each image?



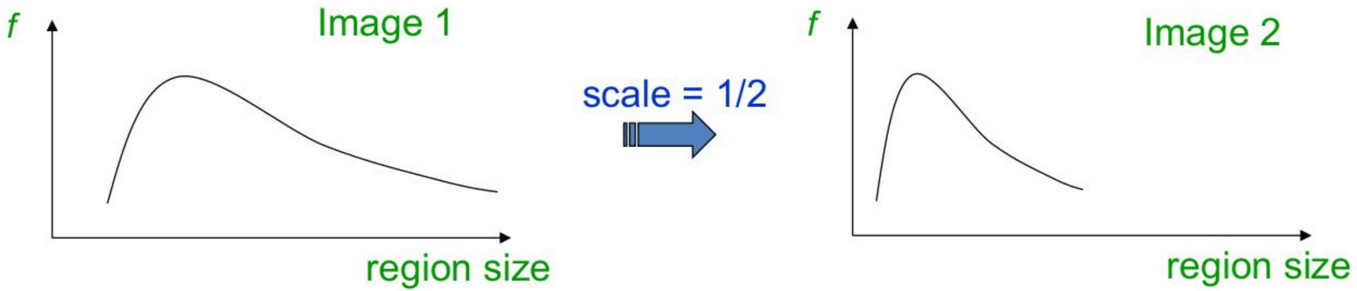
Detección de regiones de interés

Scale Invariant Detection

- Solution:
  - Design a function on the region (circle), which is "scale invariant" (the same for corresponding regions, even if they are at different scales)

    Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

  - For a point in one image, we can consider it as a function of region size (circle radius)



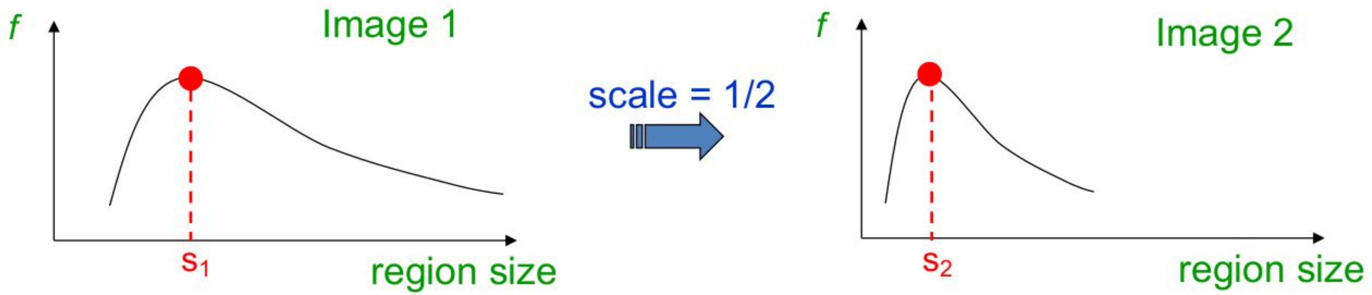
Detección de regiones de interés

Scale Invariant Detection

- Common approach:
  Take a local maximum of this function

- Observation: region size, for which the maximum is achieved, should be *co-variant* with image scale.

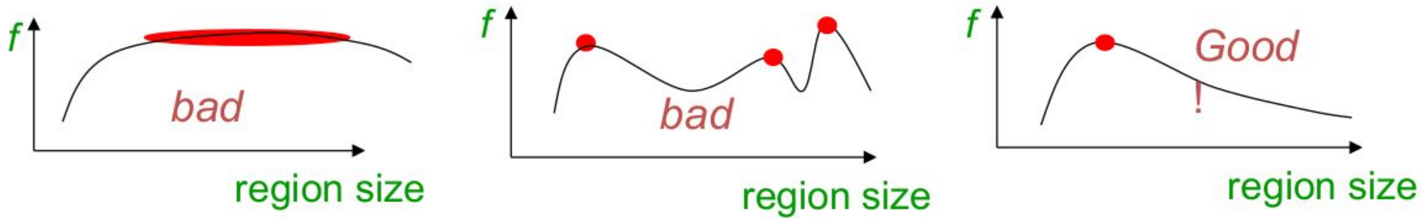
Important: this scale invariant region size is found in each image independently!



Detección de regiones de interés

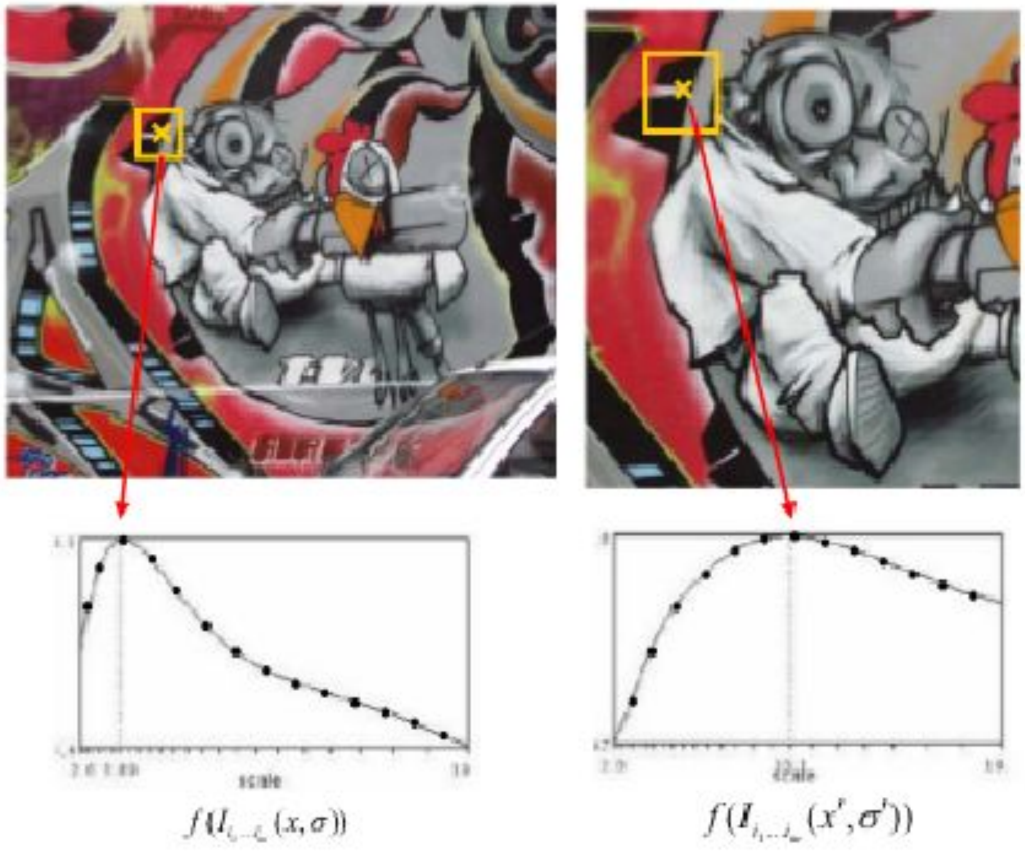
Scale Invariant Detection

- A "good" function for scale detection:
  has one stable sharp peak



- For usual images: a good function would be one which responds to contrast (sharp local intensity change)

Detección de regiones de interés



Detección de regiones de interés

Scale Invariant Detection

- Functions for determining scale $f = \text{Kernel} * \text{Image}$

Kernels:

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

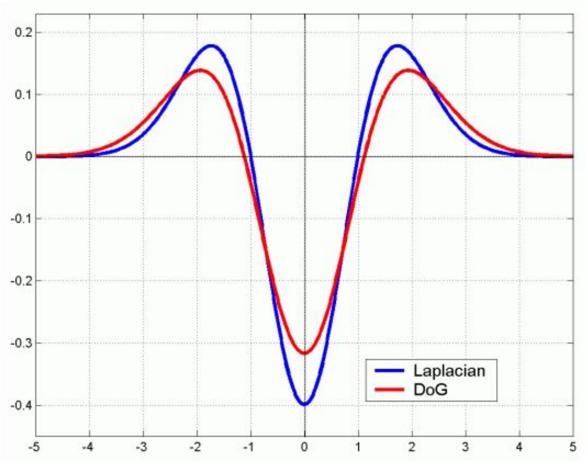
(Laplacian)

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

(Difference of Gaussians)

where Gaussian

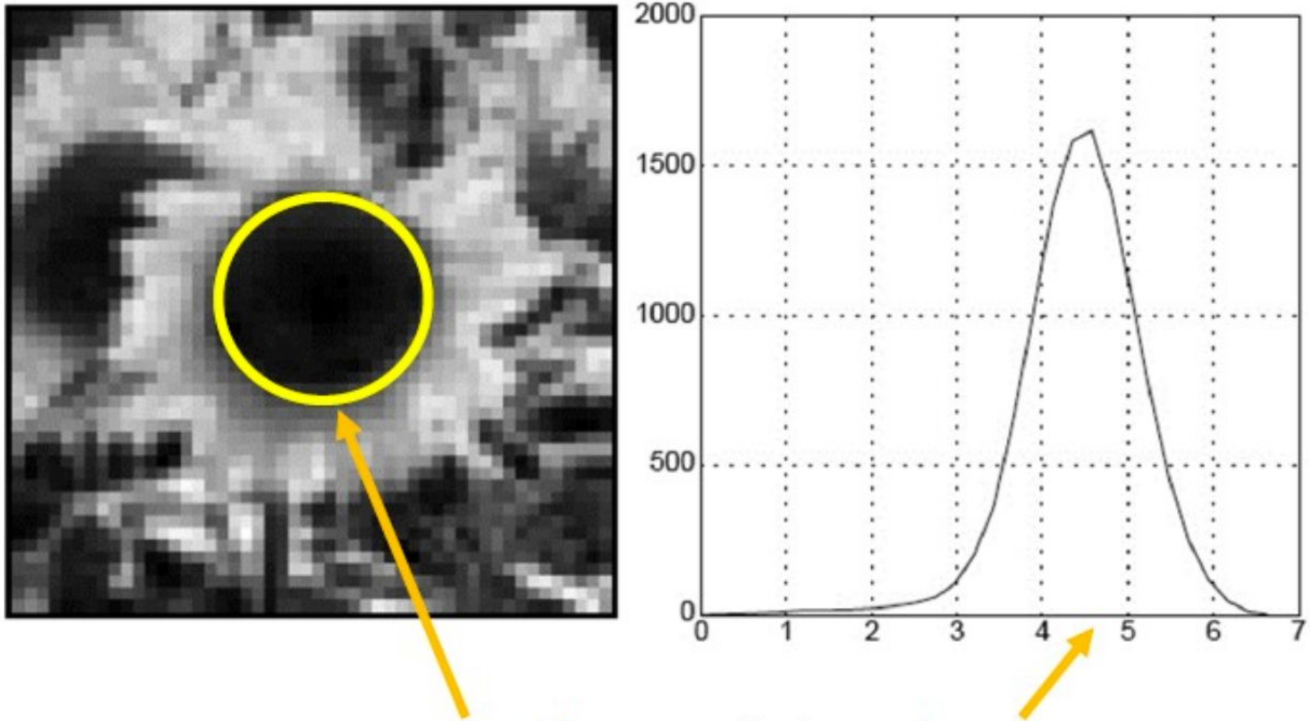
$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Note: both kernels are invariant to *scale* and *rotation*

Detección de regiones de interés

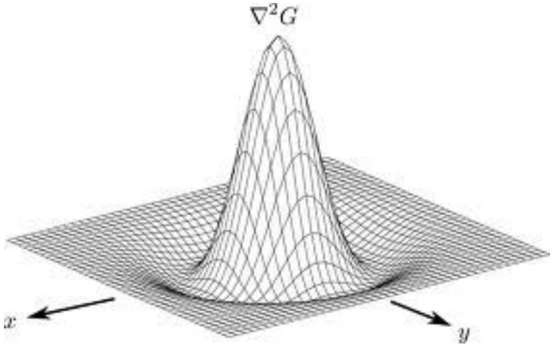
Laplacian

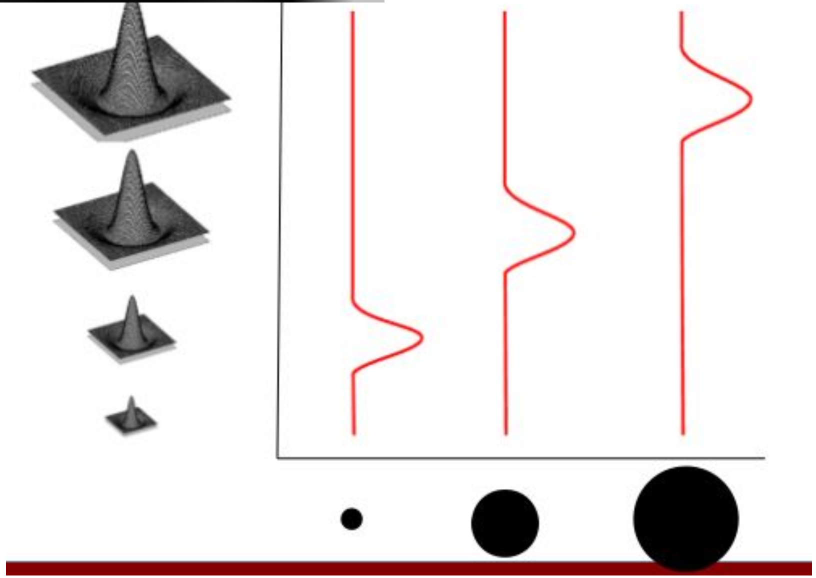


Characteristic scale

Detección de regiones de interés

Laplacian of Gaussian (LoG) = “Blob detector”

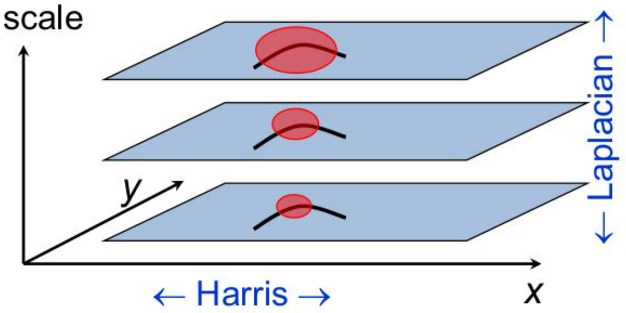




Detección de regiones de interés

Scale Invariant Detectors

- Harris-Laplacian[1]
  Find local maximum of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale

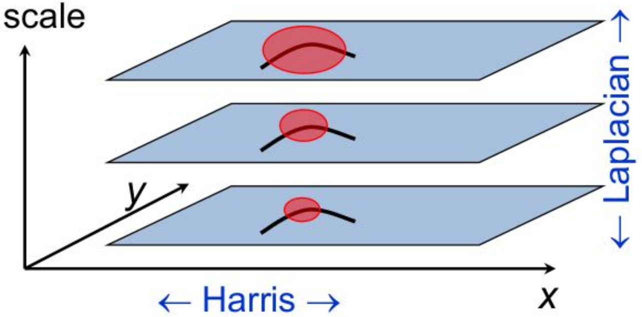


[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

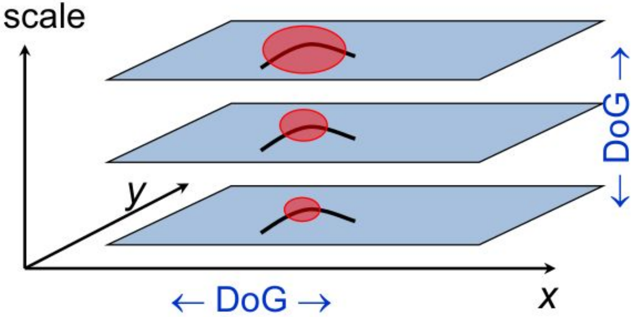
Detección de regiones de interés

Scale Invariant Detectors

- Harris-Laplacian[1]
  Find local maximum of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale



- DoG (from SIFT by Lowe)[2]
  Find local maximum of:
  - Difference of Gaussians in space and scale



[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

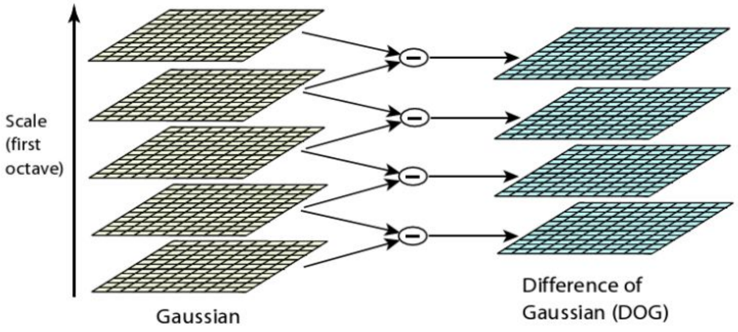
Detección de regiones de interés

Difference-of-Gaussians

$G(k^2\sigma)*I$

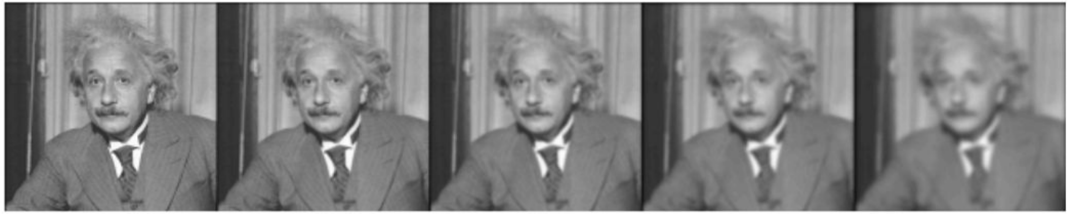
$G(k\sigma)*I$

$G(\sigma)*I$



$$D(\sigma) \equiv (G(k\sigma) - G(\sigma)) * I$$

Gaussian:

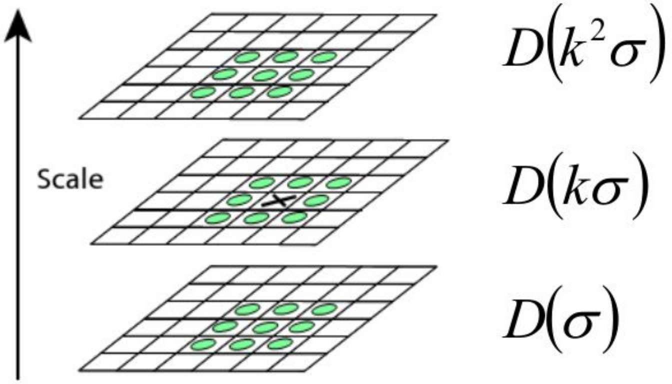


DoG:

Detección de regiones de interés

Scale-Space Extrema

- Choose all extrema within 3x3x3 neighborhood.

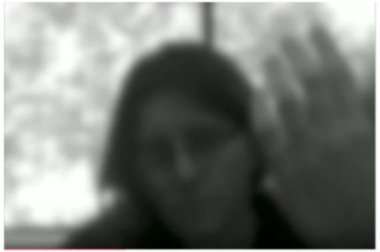


X is selected if it is larger or smaller than all 26 neighbors

Detección de regiones de interés

Difference of Gaussians (DoG)

Original video



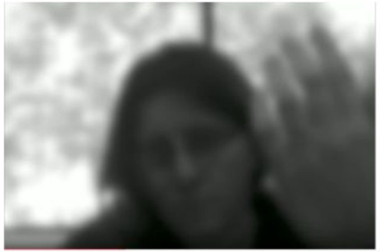
Blurred with a gaussian kernel: k_1

Blurred with a different gaussian kernel: k_2

Detección de regiones de interés

Difference of Gaussians (DoG)

Original video



Blurred with a gaussian kernel: k_1

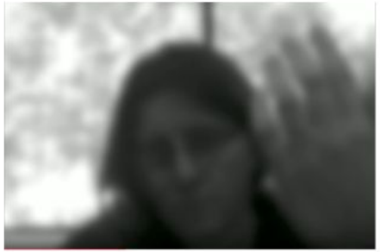
Blurred with a different gaussian kernel: k_2

What happens if you subtract one blurred image from another?

Detección de regiones de interés

Difference of Gaussians (DoG)

Original video



Blurred with a gaussian kernel: k_1

Blurred with a different gaussian kernel: k_2

DoG: $k_1 - k_2$

DoG: $k_1 - k_3$

DoG: $k_1 - k_4$

Detección de regiones de interés

Difference of Gaussians (DoG)

At different resolutions of kernel size, we see different fine details of the image. In other words, we can capture keypoints at varying scales.

DoG: $k_1 - k_2$ DoG: $k_1 - k_3$ DoG: $k_1 - k_4$