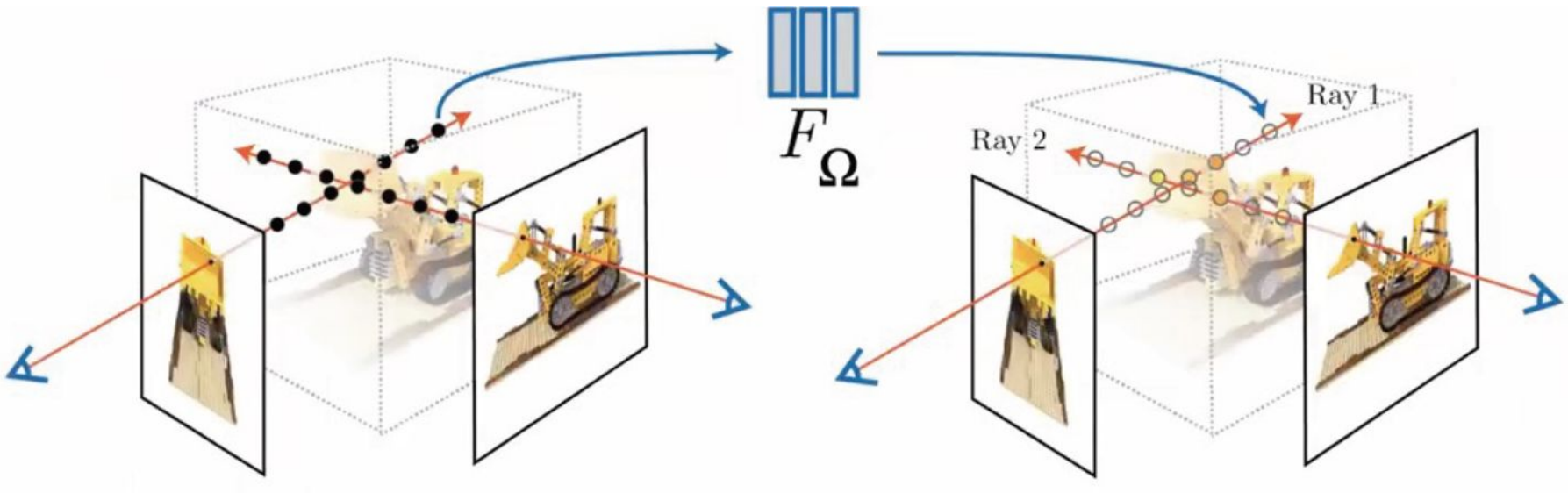
Visión Computacional
para imágenes y videos

Módulo 3:
Detección de puntos de interés



Gilberto Ochoa Ruiz, PhD
Associate Professor
Researcher in Computer Vision

Computer Science Dept.
Advanced AI Group
gilberto.ochoa@tec.mx

Tecnológico de Monterrey

TODAY'S AGENDA

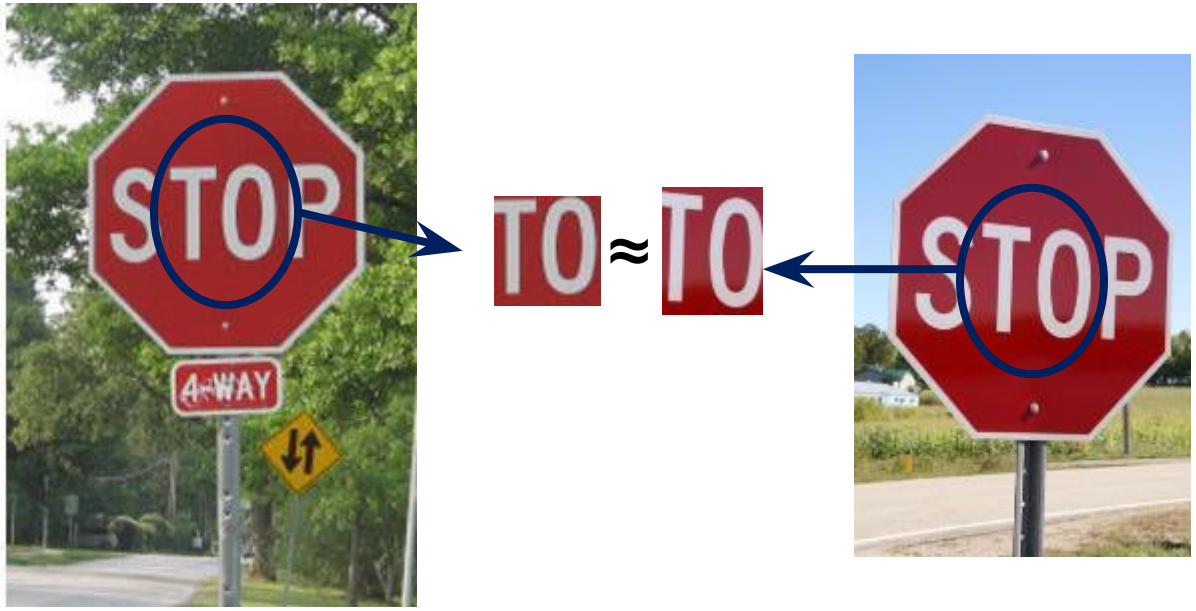
introduction to our course

1. Today class goal
2. About us
3. Course policies
4. Evaluation schema
5. Summary and conclusions

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Matching images
A challenge for computer vision systems!



Motivation

It enables to map 3D points to images

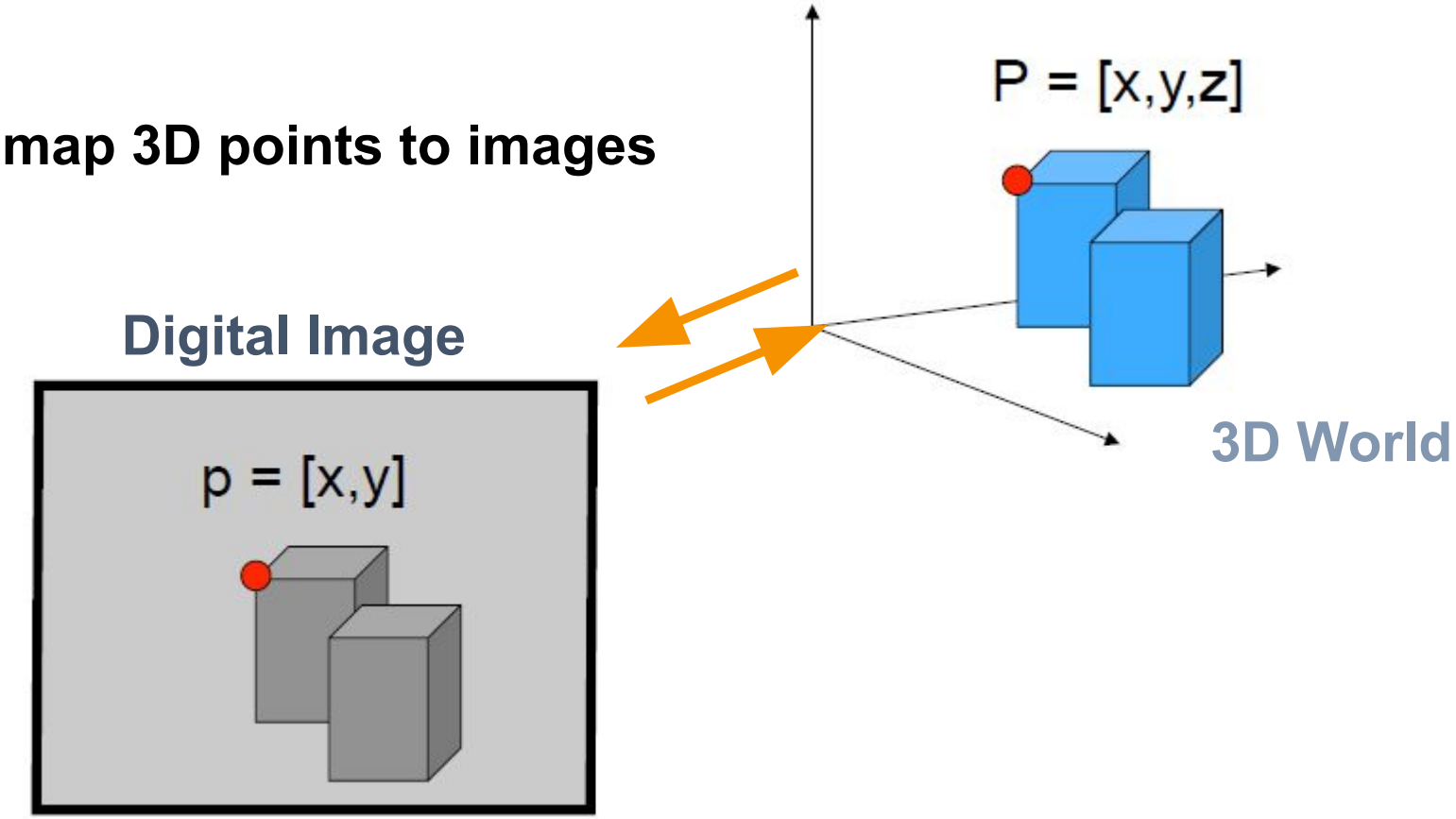


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Motivation

Image alignment

Image Stitching

3D Reconstruction

Object tracking in video

Drone and Robot Navigation

Image Retrieval

Object Recognition

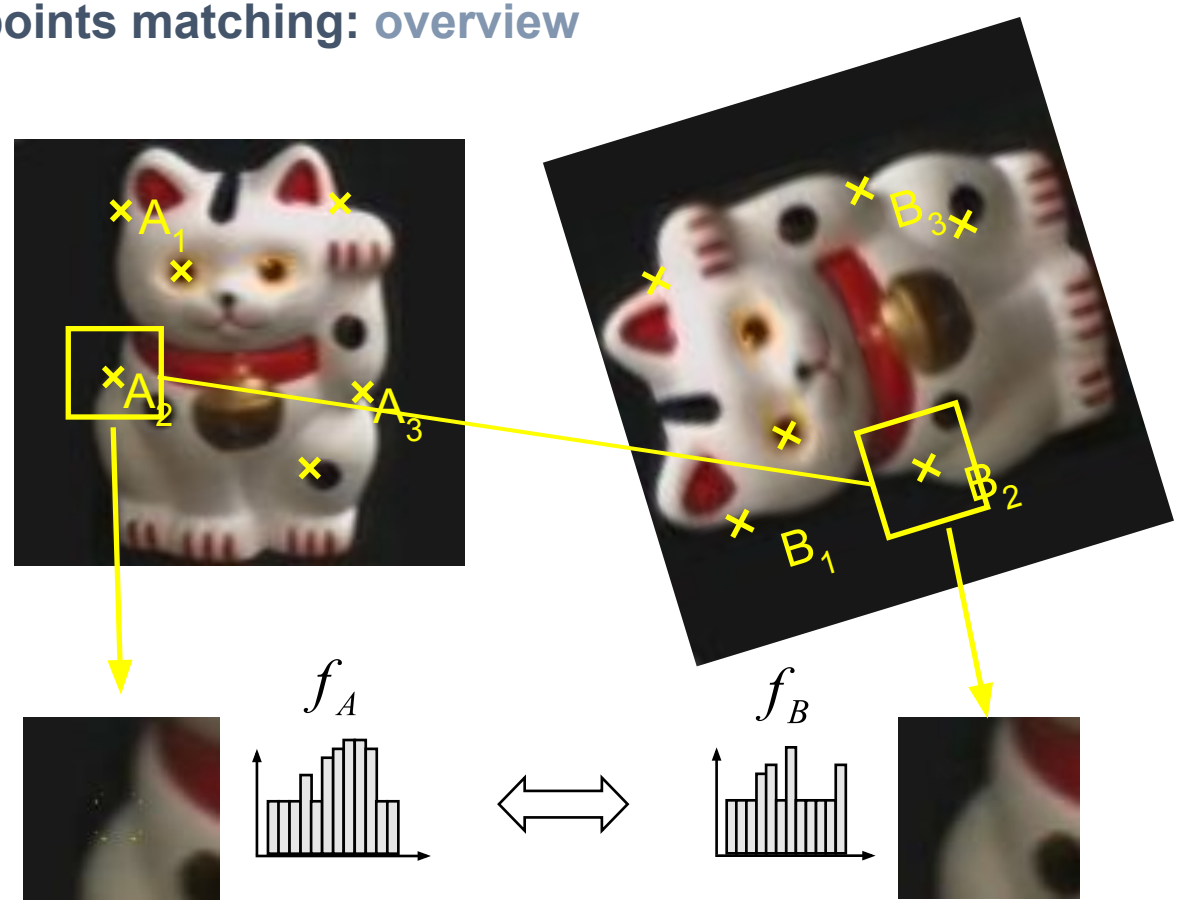
Biometrics Applications

Note: “interest points” = “keypoints”, aka “features points”

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Keypoints matching: overview



$$d(f_A, f_B) < T$$

1. Find a set of **distinctive keypoints**

2. Define a **region** around each keypoint

3. Calculate a **local descriptor** from the normalized region

4. Find **correspondences** between descriptors

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

El contenido de la imagen es transformado en coordenadas de POIs que son invariantes a traslación, rotación y cambios de escala, etc

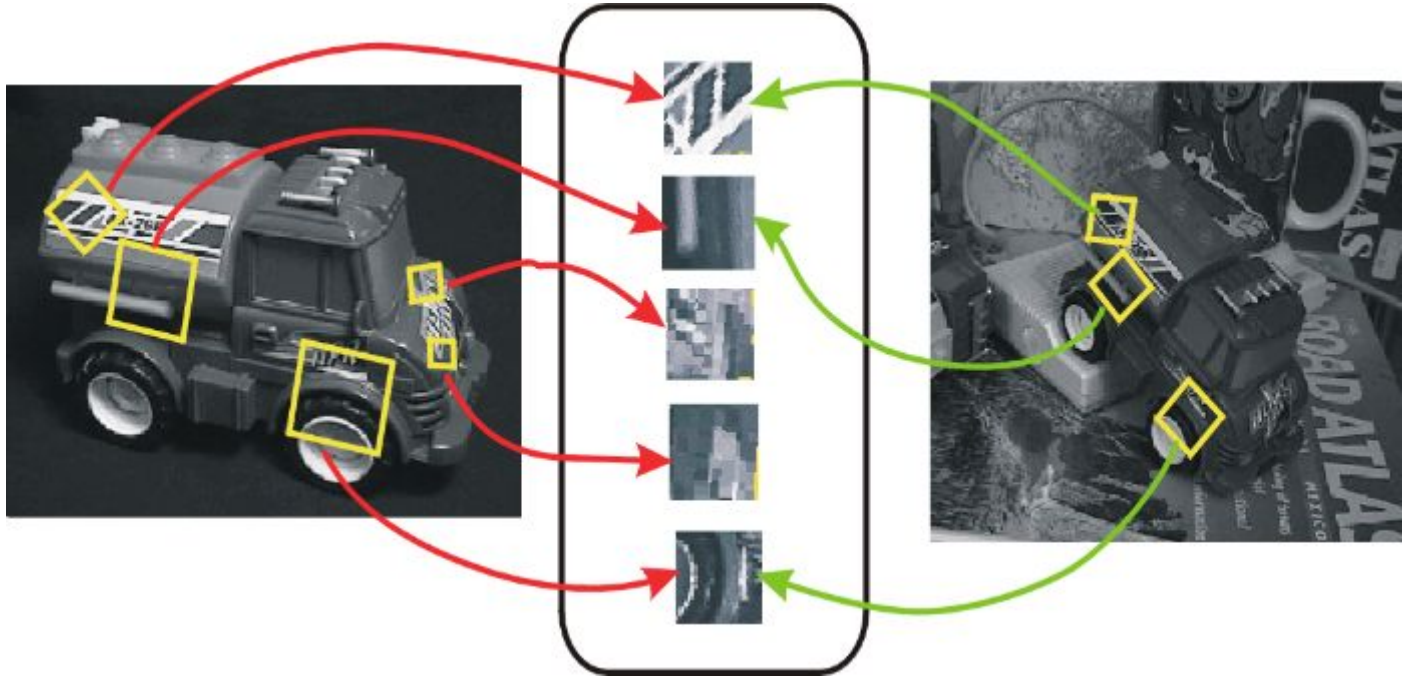
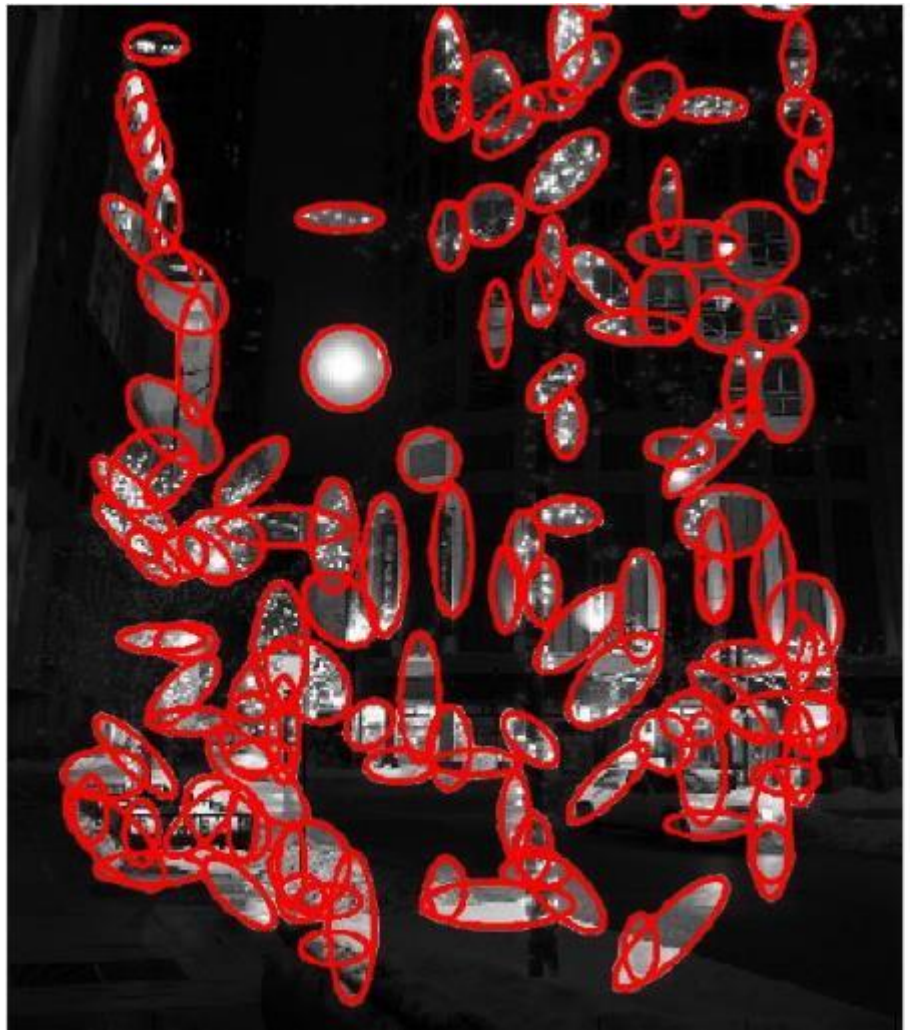


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

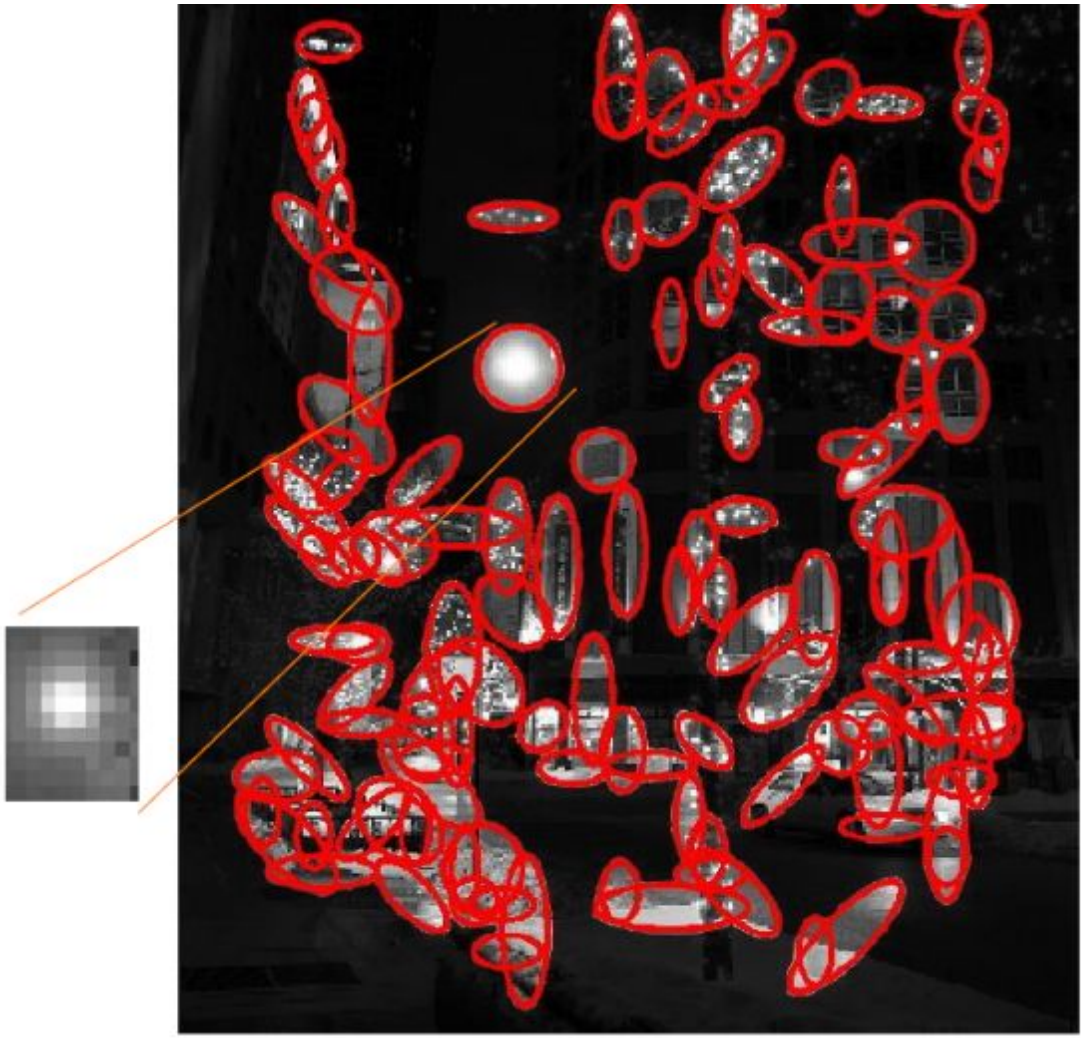


Feature Detection

e.g. DoG

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction



Feature Detection

e.g. DoG

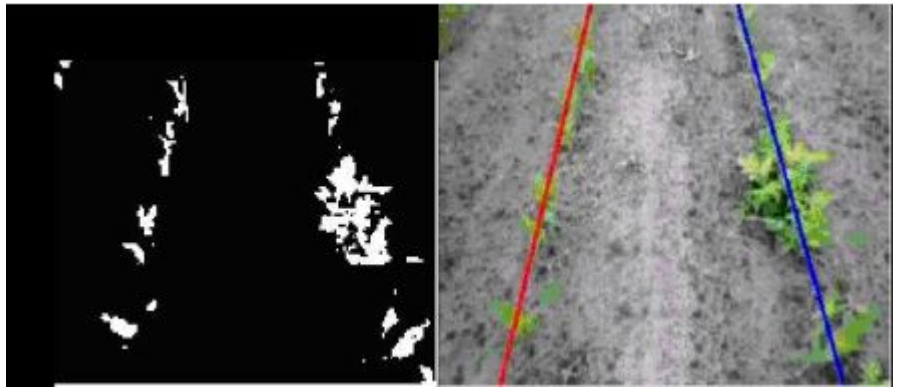
Feature Description

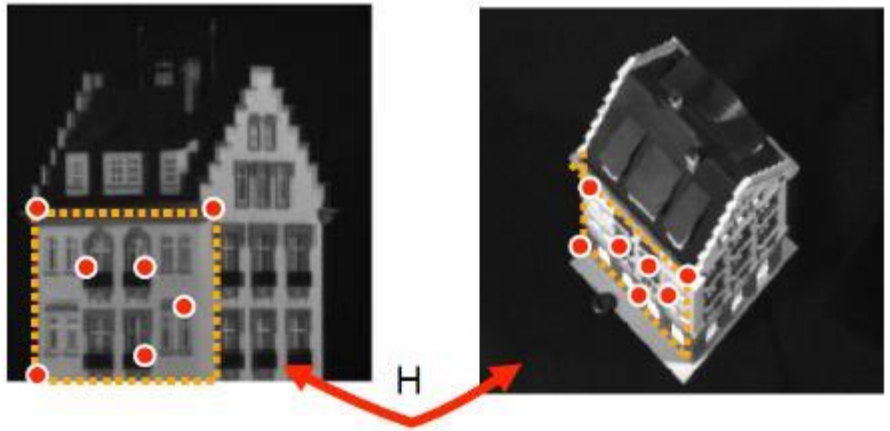
e.g. SIFT

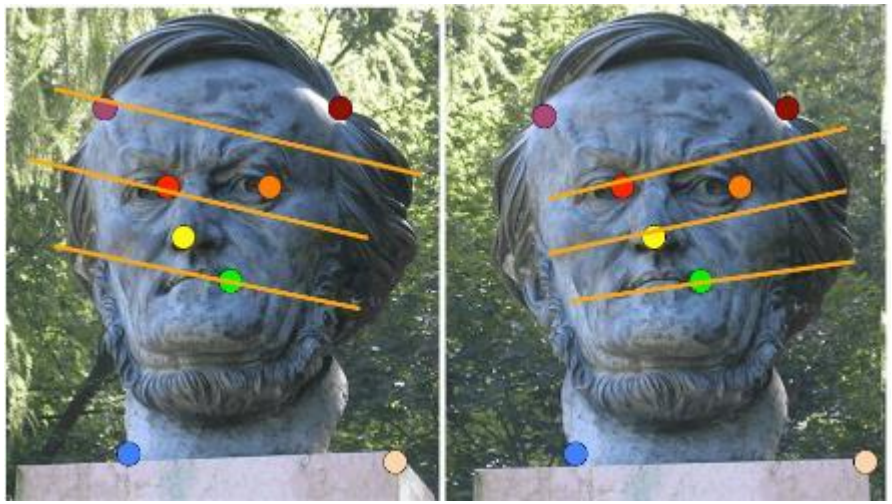
- Estimation
- Matching
- Indexing
- Detection

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction







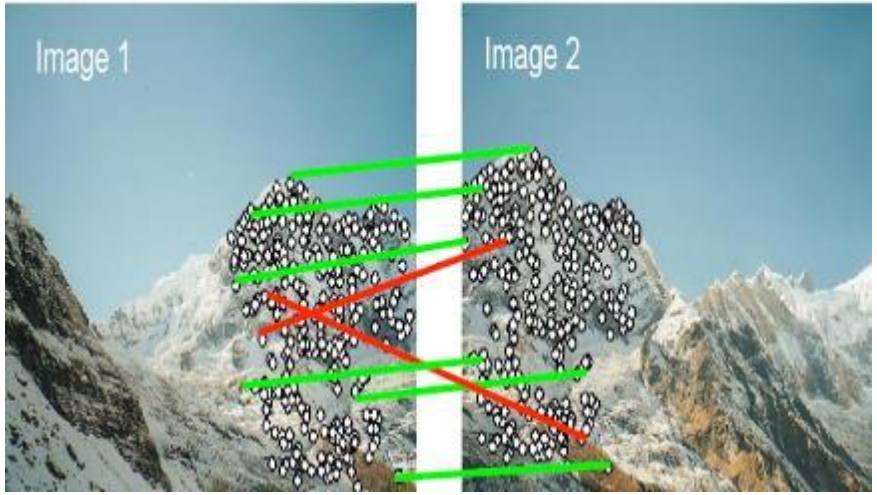
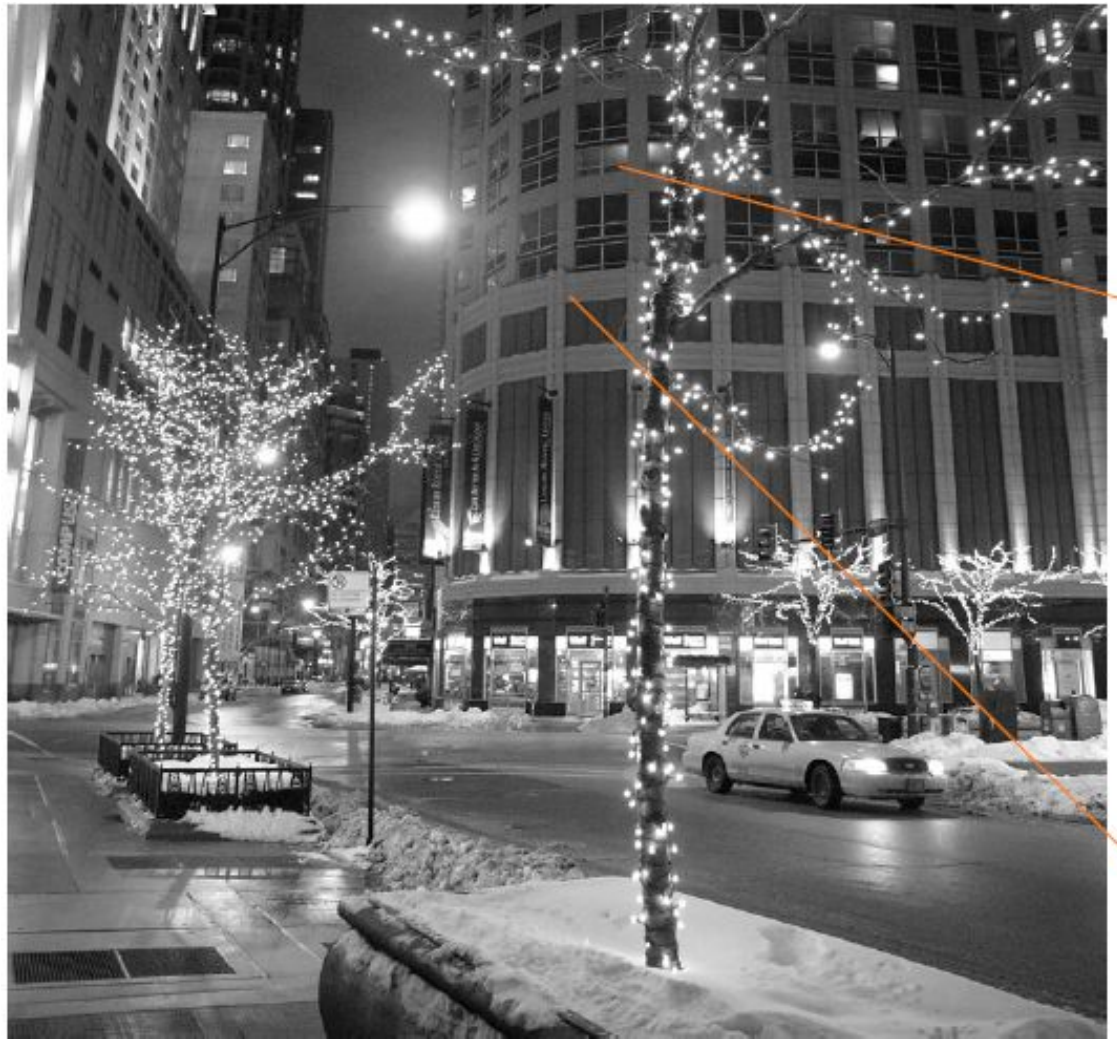


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction



What produces an edge?

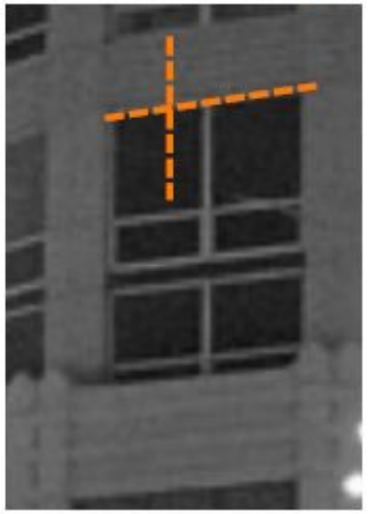


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Depth discontinuities

Identify sudden changes in the image

Changes or discontinuities
In the **orientation of surfaces**

Discontinuity in **reflectance**
(i.e., changes in the properties
of the mareriales)

Discontinuity in **ilumination**
(e.g., bright, shadows)

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Edge Detection

original

smoothed (5x5 Gaussian)

smoothed – original
(scaled by 4, offset +128)

J. Canny, A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Examples

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Criteria for **optimal edge detection** (Canny 86):

– Good precisión in detection

• Minimize the probability of false positives and false negatives(not finding edges)

– Good edge localization

• Edges must be detected as close as posible to the true edged

– Constraints in response

• Minimize the number of local maxima around a true edge

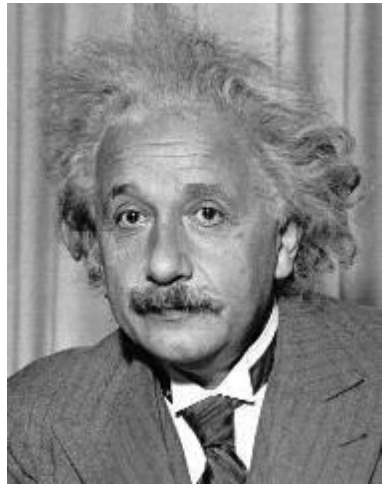
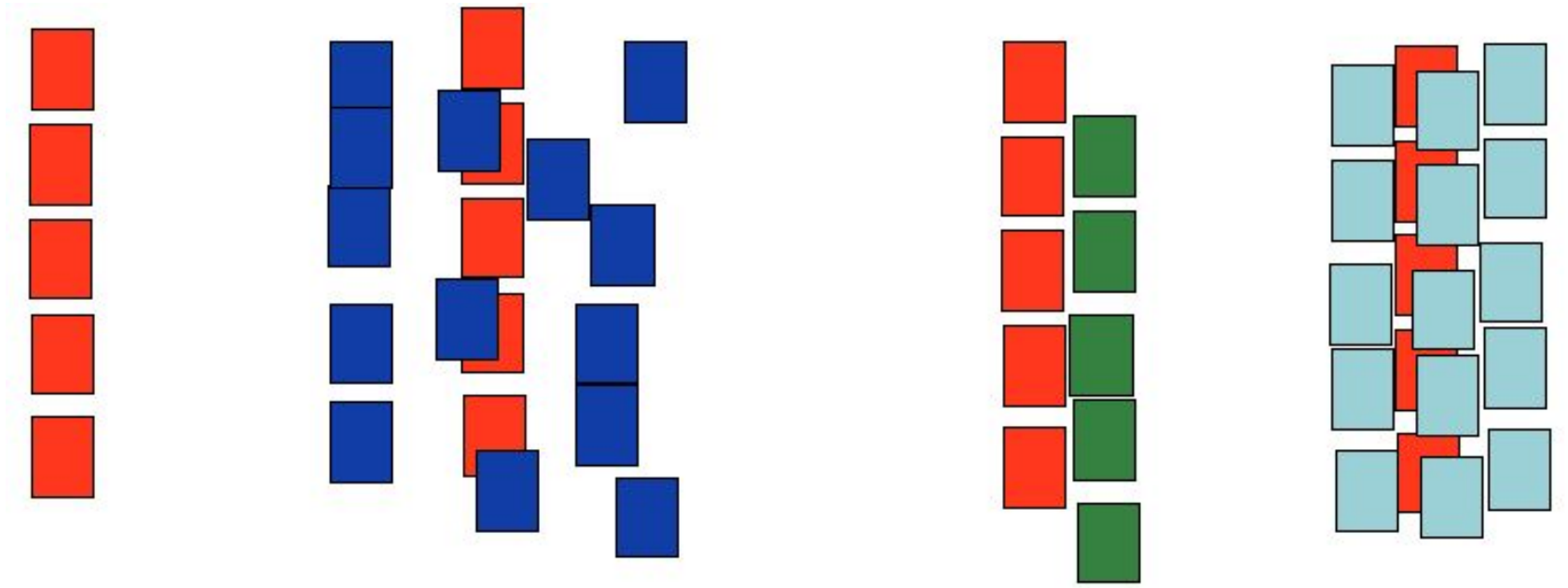


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Criteria for optimal **edge detection** (Canny 86):



True Edge

Poor Robustness to noise

Poor Localization

Too many responses

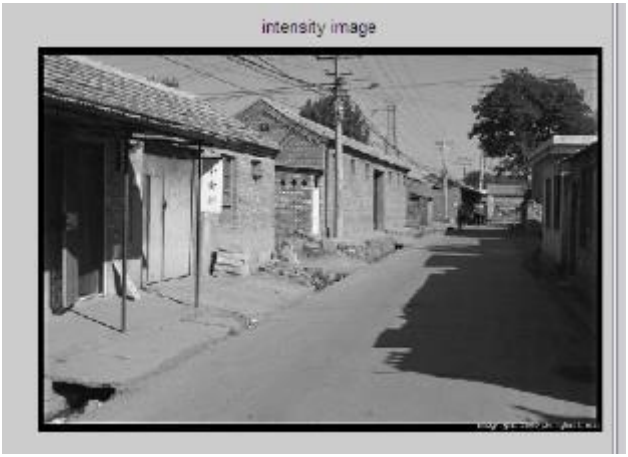
Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Criteria for **optimal edge detection** (Canny 86):

- Designing a good edge detector, two ingredients :

1. Use derivatives (x-y) direction to **find zonea with strong gradient**
2. To use a filter to **smooth the signal** before finding discreet derivative



*

=

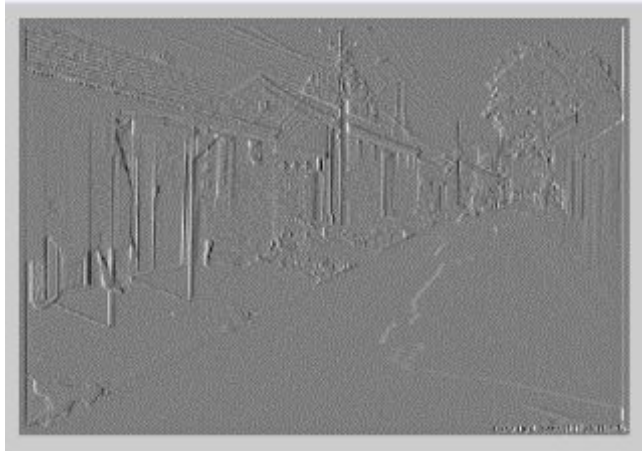
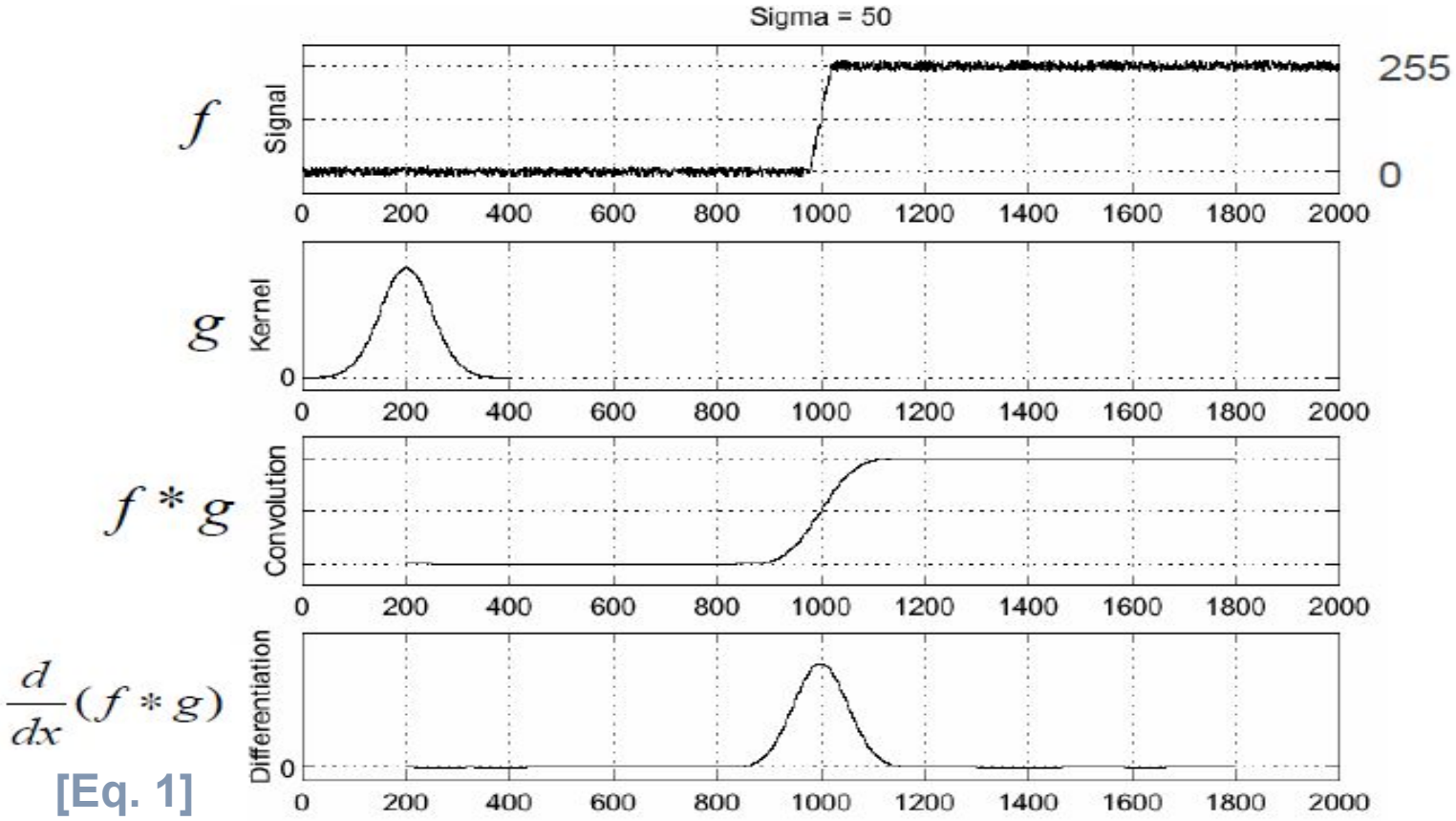


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction



[Eq. 1]

$$= \frac{dg}{dx} * f =$$ "Derivativa of the Gaussian Filter" [Eq. 2]

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Smoothing

$$I' = g(x,y) * I \quad \text{[Eq. 3]}$$

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{[Eq. 4]}$$

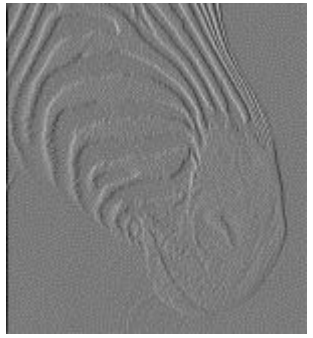
Derivatuve or gradient extraction

$$S = \nabla(g * I) = (\nabla g) * I = \begin{bmatrix} g_x \\ g_y \end{bmatrix} * I = \begin{bmatrix} g_x * I \\ g_y * I \end{bmatrix} = \begin{bmatrix} S_x & S_y \end{bmatrix} \quad \text{[Eq. 5]}$$

= gradient vector

$$\nabla g = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

[Eq. 6]



Edge detection with Canny

original | Canny with $\sigma = 1$ | Canny with $\sigma = 2$

- The choice of **σ** depends on the desired behavior
  - **large σ** detects lines at large at big scales
  - **small σ** detects finer lines or at lower scales

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Keypont detection:

it depends on edge detection

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Keypoint detection: It depends on edge detection
Relavant information but not sufficient
Image patches are very similar to each other

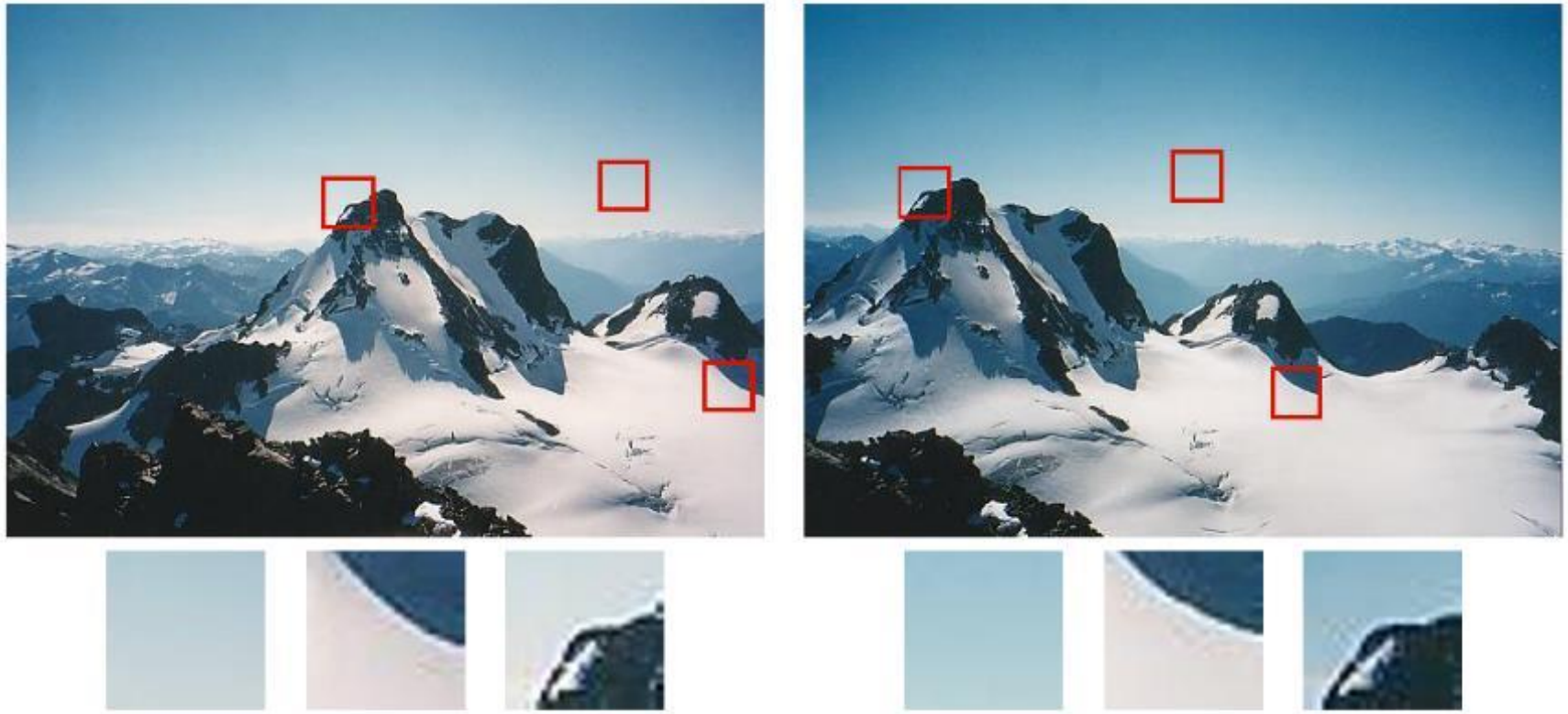


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

- **Motivation: panorama stitching**

We have two images, with traslation and rotation

– How to combine them?

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

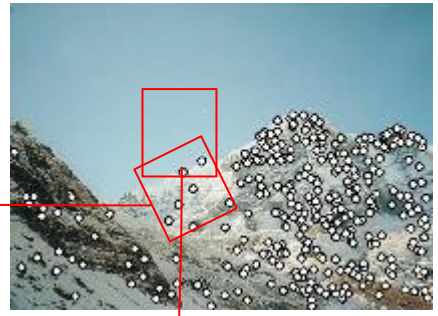
Keypoint detection

1) **Detection:** Identify keypoints or features

2) **Description:** Extract a vector or descriptor around each keypoint

3) **Matching:** Find corresondences between each of the views

4) **Readjust:** Mix images



$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$

$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

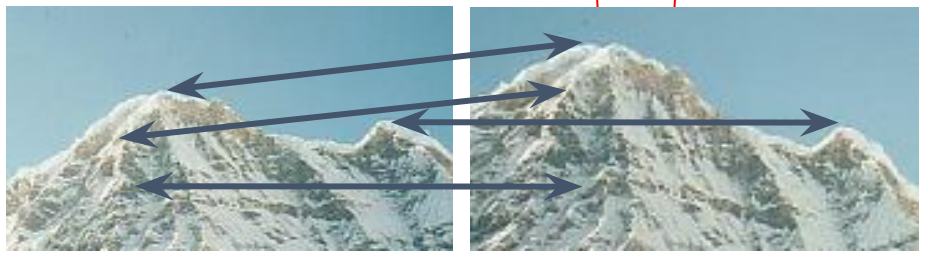


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Several detectors have been proposed through the years

Hessian & Harris	[Beaudet '78], [Harris '88]
Laplacian, DoG	[Lindeberg '98], [Lowe 1999]
Harris-/Hessian-Laplace	[Mikolajczyk & Schmid '01]
Harris-/Hessian-Affine	[Mikolajczyk & Schmid '04]
EBR and IBR	[Tuytelaars & Van Gool '04]
MSER	[Matas '02]
Salient Regions	[Kadir & Brady '01]

Tuytelaars, T, and Mikolajczyk, **Local Invariant Feature Detection: A Survey**, Foundations and Trends in Computer Vision, Vol. 3. Num. 3, pag. 177-280, 2007.

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

- **Repeatibility**

– The same features should be found in various images despite photometric or geometric transformations

Invariant to traslation, rotation, etc

- **Prominency (saliency)**

– Each “feature” should be found within “regions of interest” of the image

- **Localization**

– Each feature occupies a “relatively small área” of the image

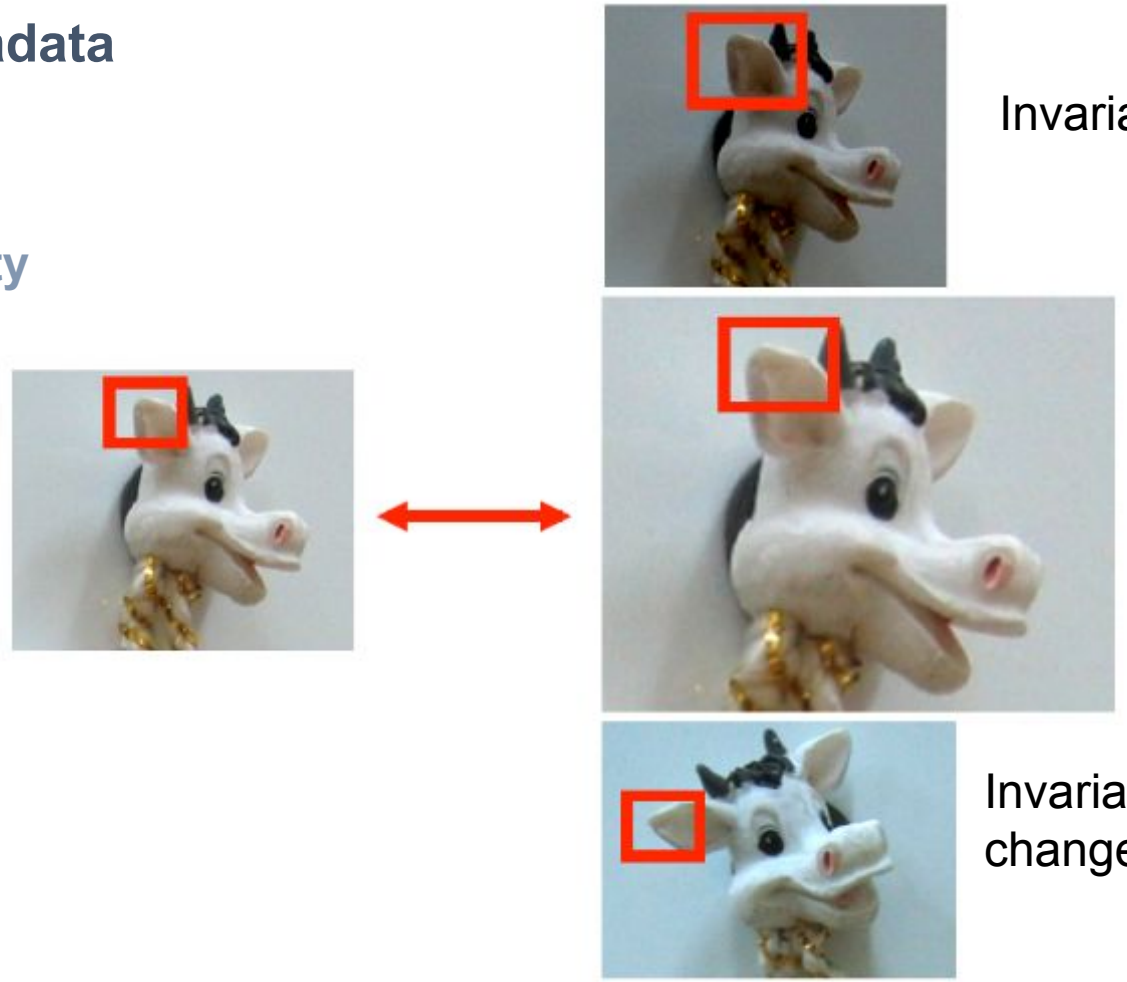
Desiradata

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Desiradata

Repeatibility



Invariance to **ilumination**

Invariance to **scale changes**

Invariancie to **position/pose** changes: **rotation / affine**

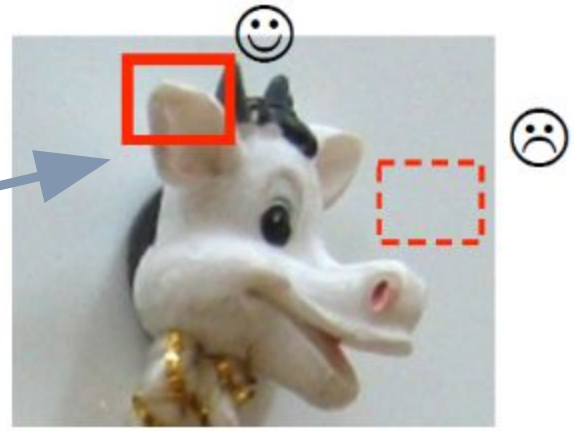
Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Desiradata

Prominency (saliency)

Each “feature” should be found within “regions of interest” of the image



Localización

Each feature occupies a “relatively small área” of the image

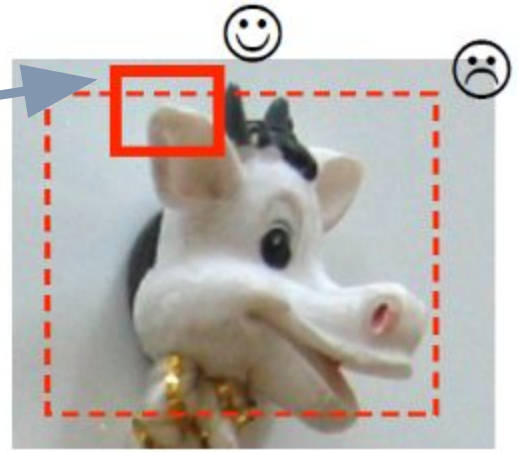


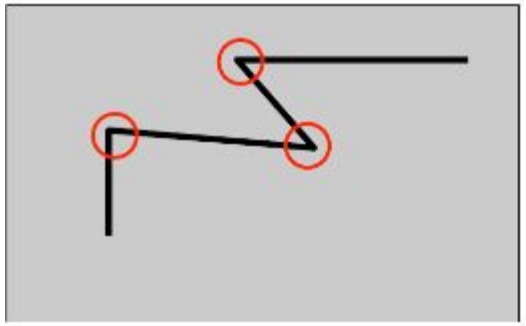
Image Analysis for Computer vision

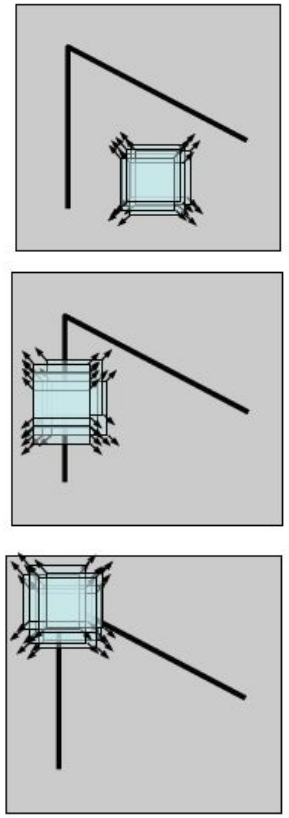
Keypoint Detection and Feature Extraction

Harris corner detector

Basic idea:

Explore changes in intensity within a windo,w varying the location the window





"flat" region: no changes in any direction

"edge": no changes in the line direction

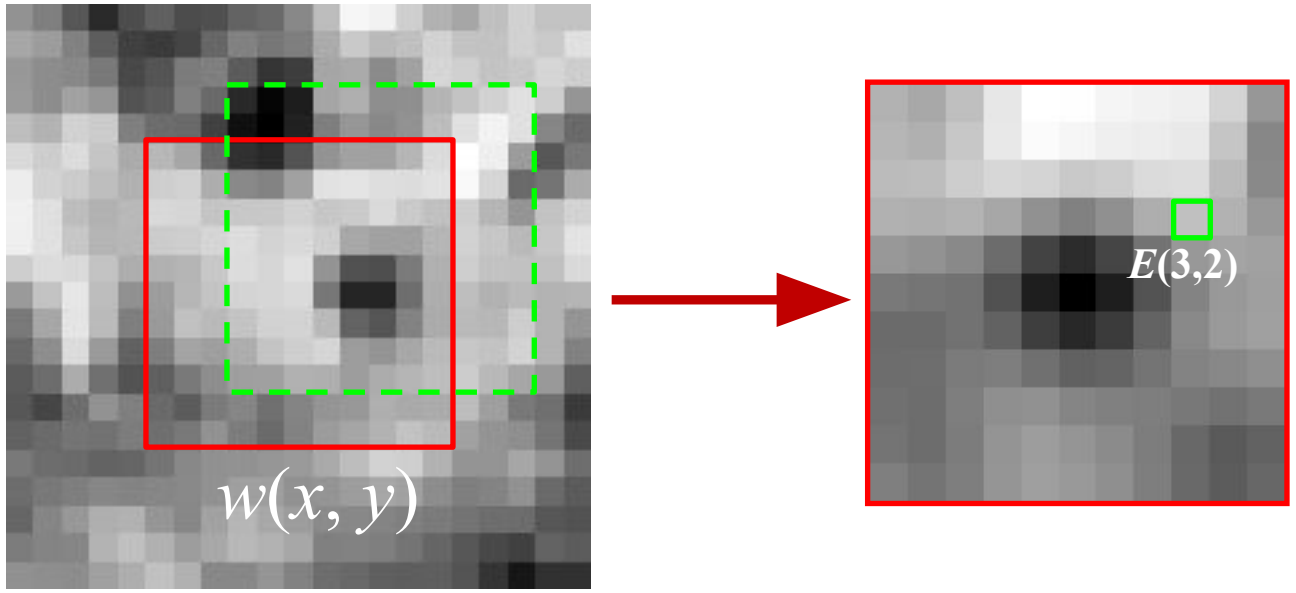
"corner": high variations or significative in all directions

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

Changes in appearance in the window ***w(x,y)*** given a **[*u,v*]** displacement:

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

$I(x, y)$



$E(u, v)$

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Changes in appearance in the window ***w(x,y)*** given a **[*u,v*]** displacement:

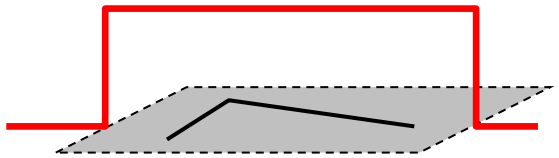
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Función ventana

Shift de intensidad

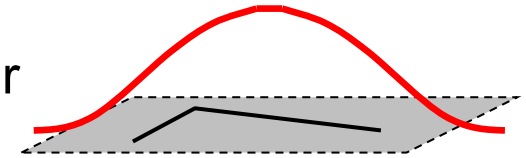
Intensidad

Función kernel $w(x,y) =$



1 in window, 0 outside

or



Gaussian

The quadratic approximation can be simplified to

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

M is the ***second moment matrix***, calculated from the image derivatives

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

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \;\; I_y] = \sum \nabla I (\nabla I)^T$$

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

The surface ***E*(*u*,*v*)** is locally approximated by a quadratic in the form

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

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

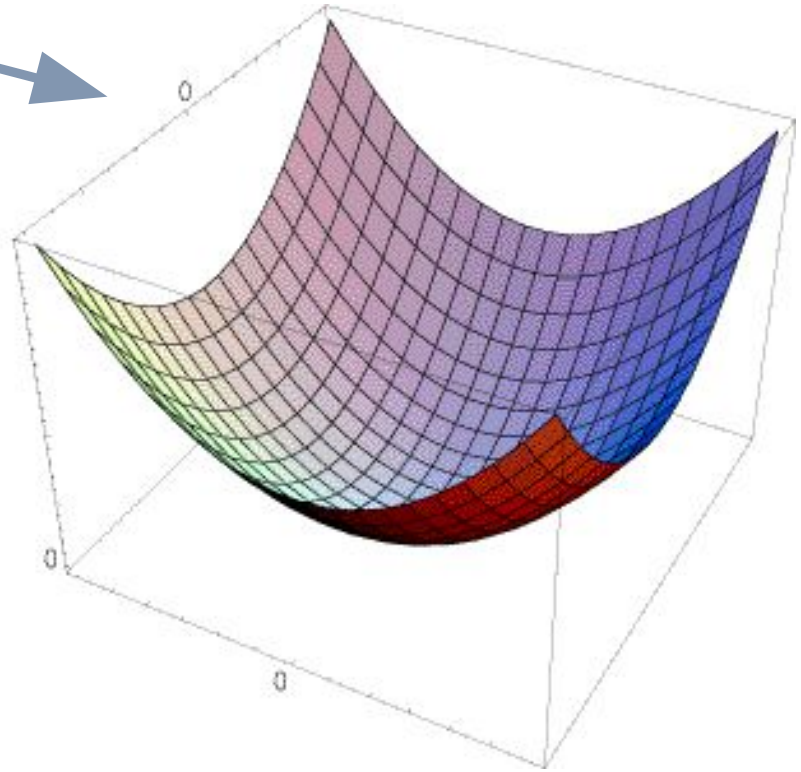
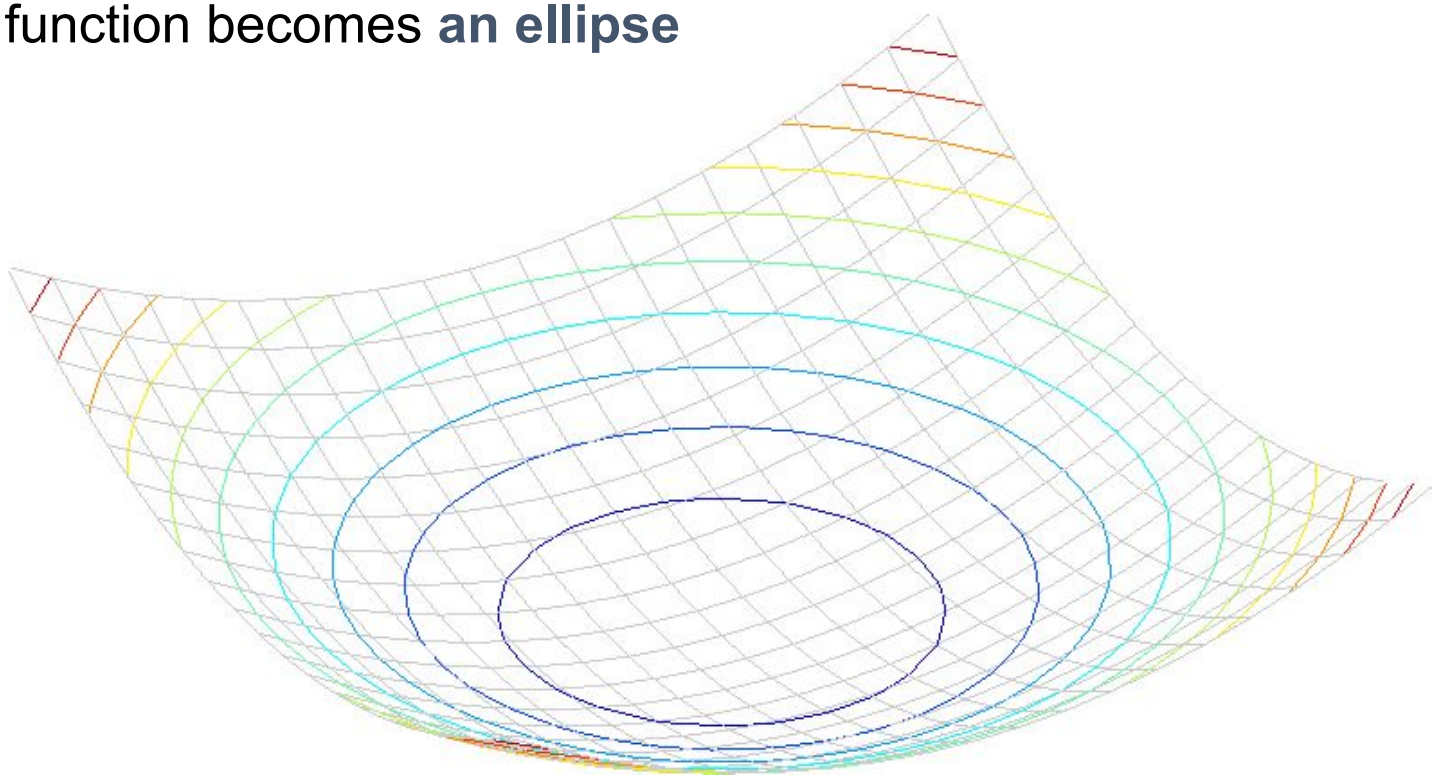


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Let's consider a "slice" of ***E*(*u*, *v*)** $\quad [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{constante}$

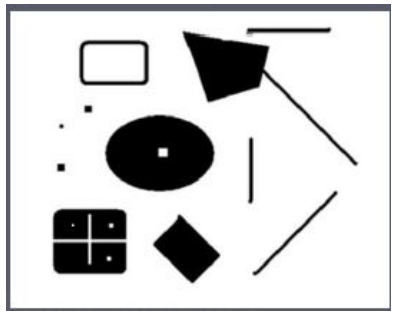
In 2D, the function becomes **an ellipse**



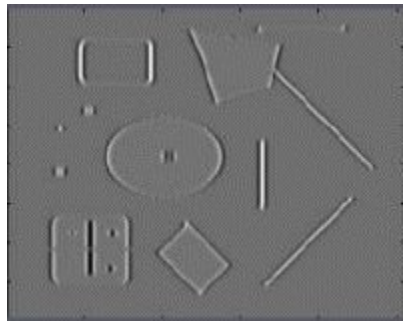
Harris corner detector

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

M is a 2 x 2 matrix that contains derivatives around a point

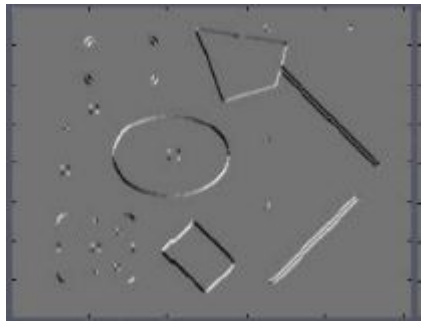


Input Image



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

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



$$I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$$

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Let's consider a "slice" of ***E*(*u*, *v*)** □ $[u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{constante}$

In 2D, the function becomes **an ellipse**

Diagonalizing the matrix M □ $M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$

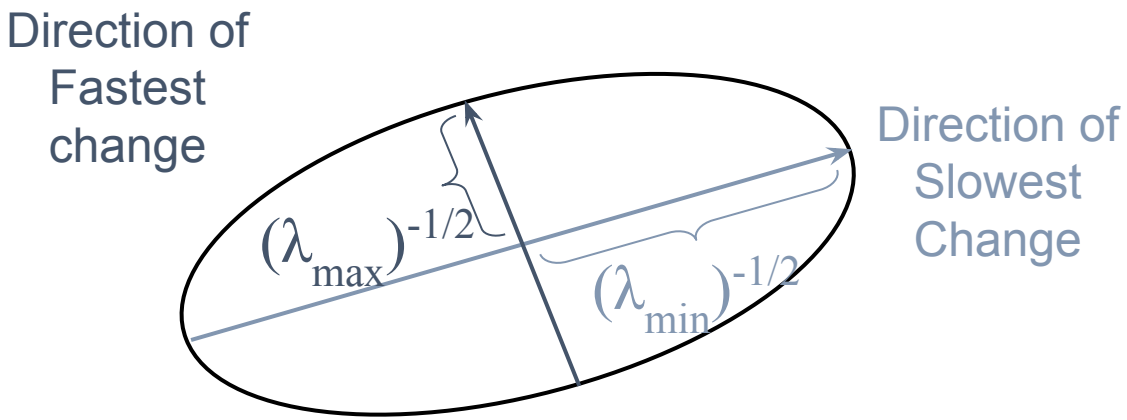
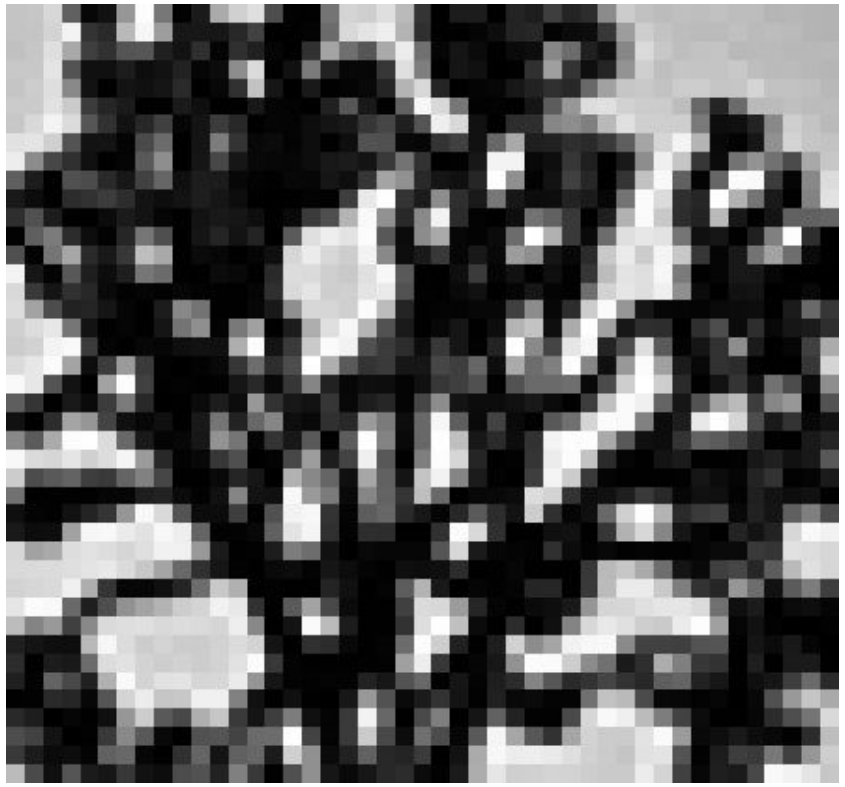


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Harris corner detector



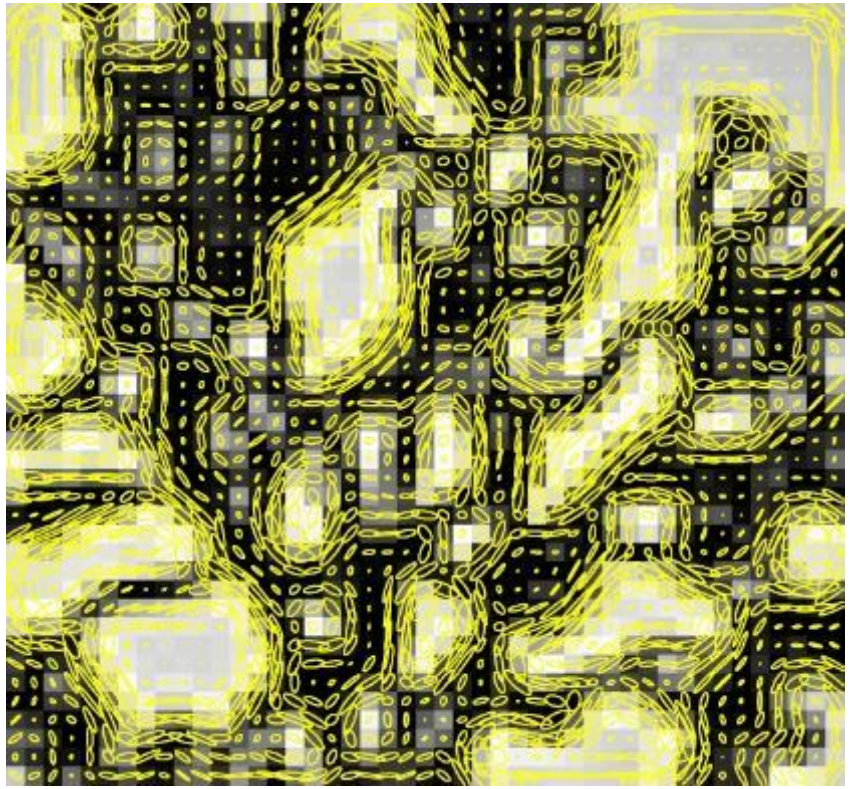


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

- **Eigenvalues** gice a measure of the "**cornerness**" of a given pixel

- But, computing them the values is computationally expensive

- We will see an alternative (in the next slide)

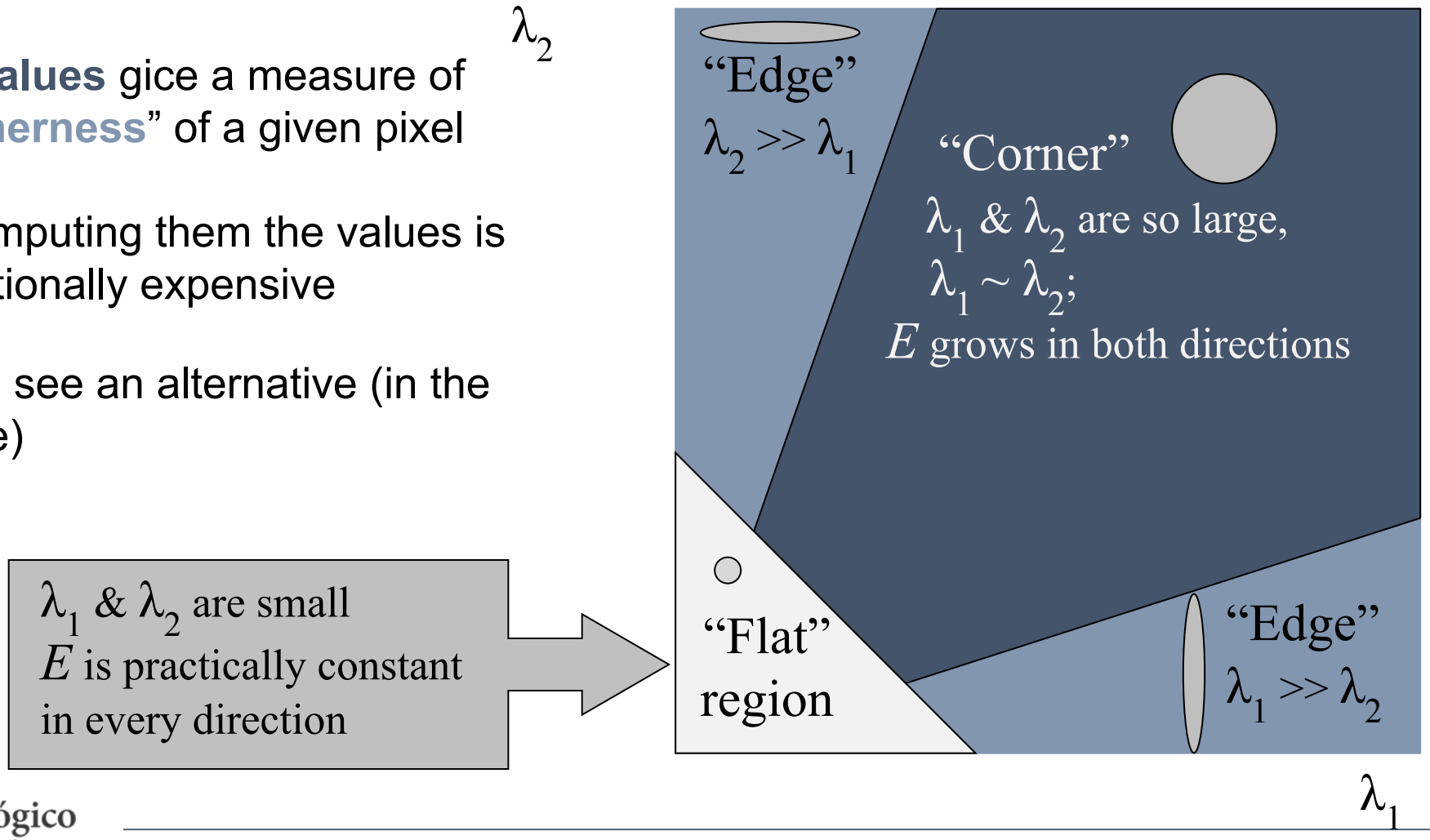


Image Analysis for Computer vision

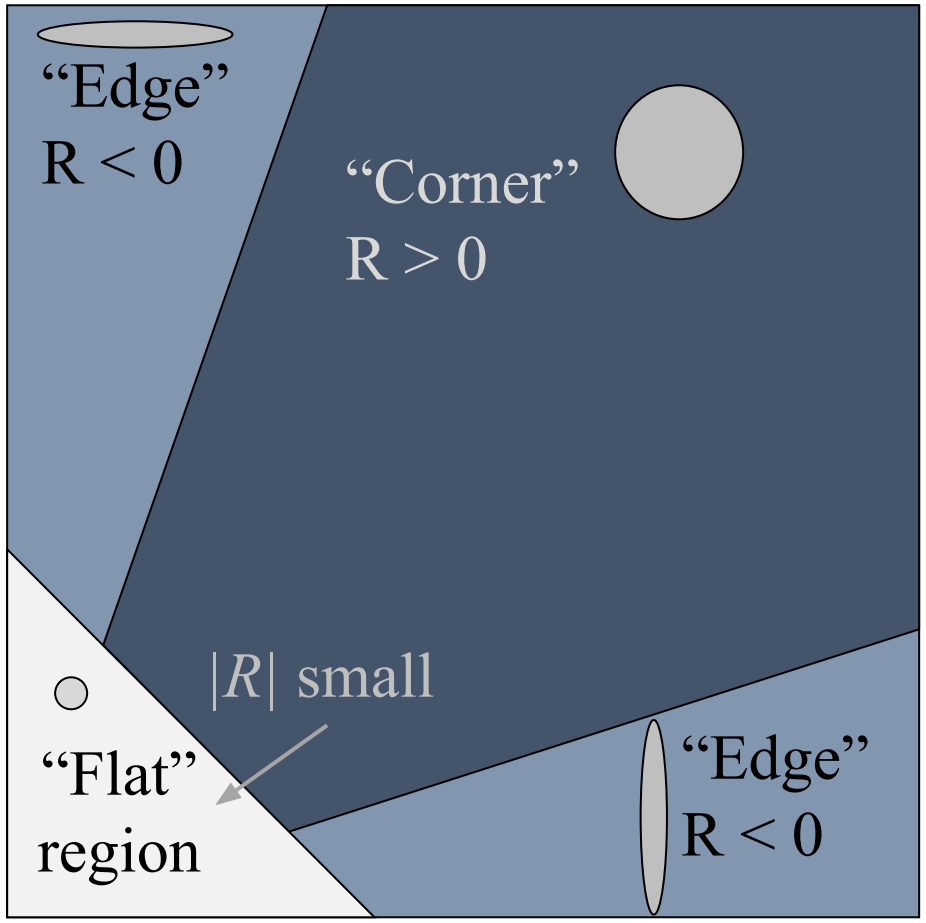
Keypoint Detection and Feature Extraction

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

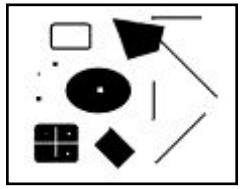
α: constant (0.04 a 0.06)

$$det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$$

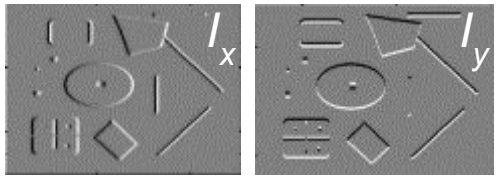
$$trace\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = a + d$$



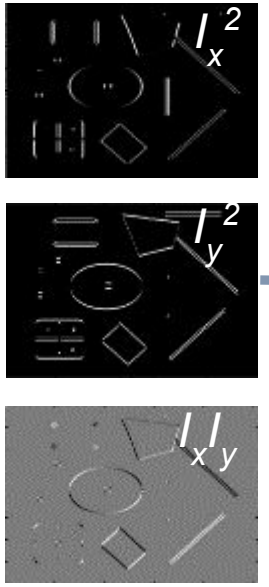
Harris corner detector



1. Image derivatives



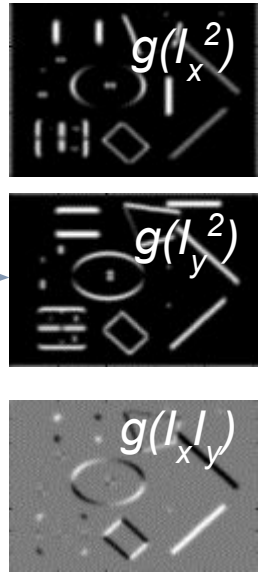
2. Squares of the derivatives



$$\det M = \lambda_1 \lambda_2$$

$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

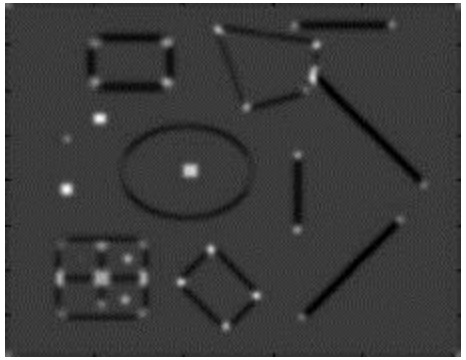
3. Gaussian filter $g(\sigma_I)$



4. "Cornerness" function

$$har = \det[M(\sigma_I, \sigma_D)] - \alpha[\operatorname{trace}(M(\sigma_I, \sigma_D))^2]$$

$$= g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$



5. Non-maxima supression (NMS)

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Harris corner detector → Input Image

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Harris corner detector → Cornerness R

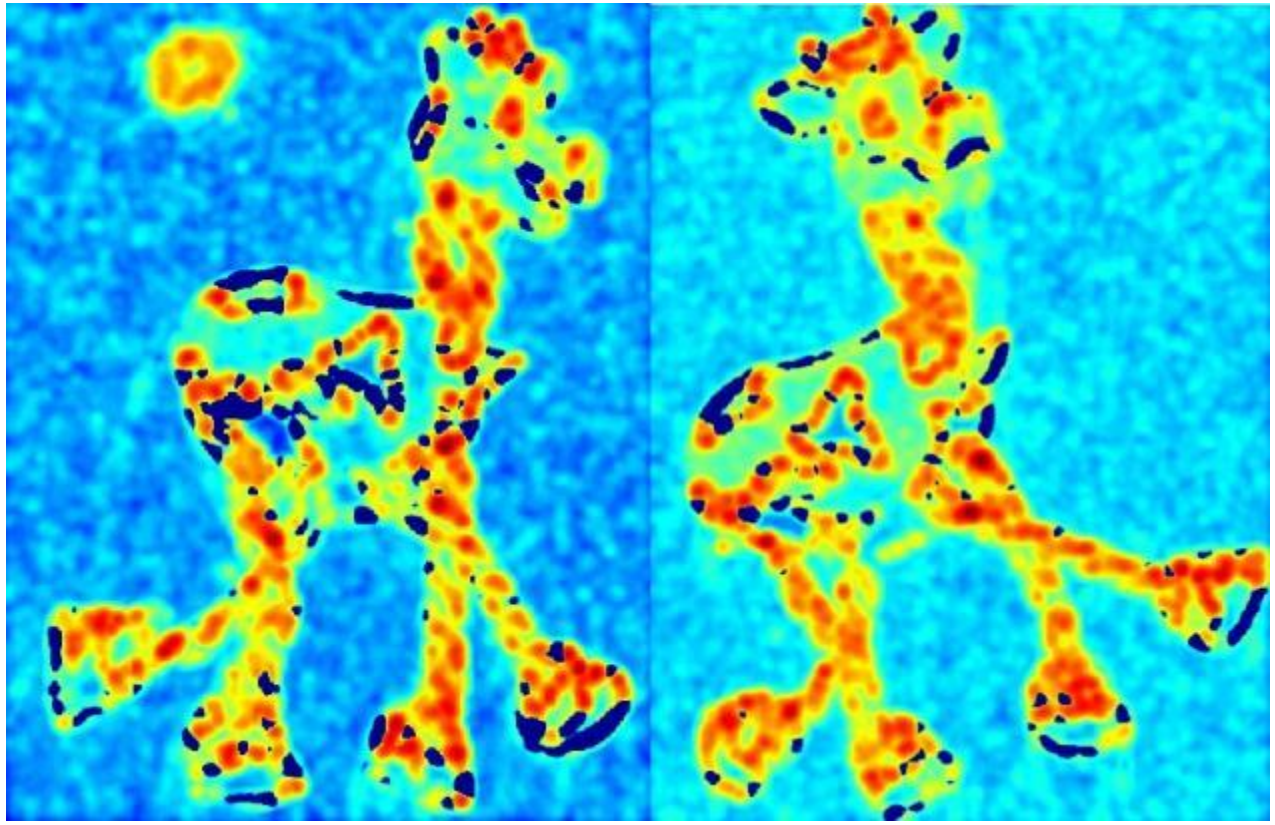


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Harris corner detector → Points with R > threshold

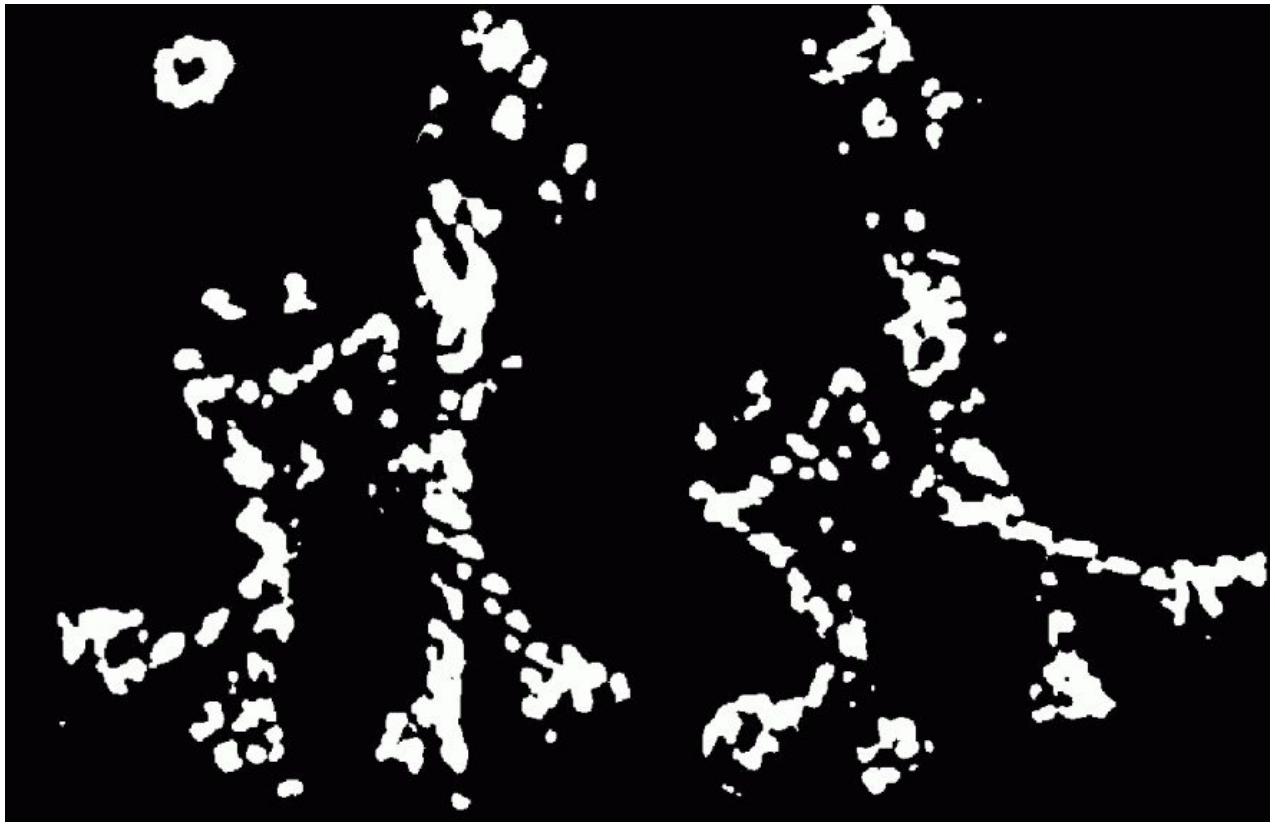


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Harris corner detector → Apply NMS

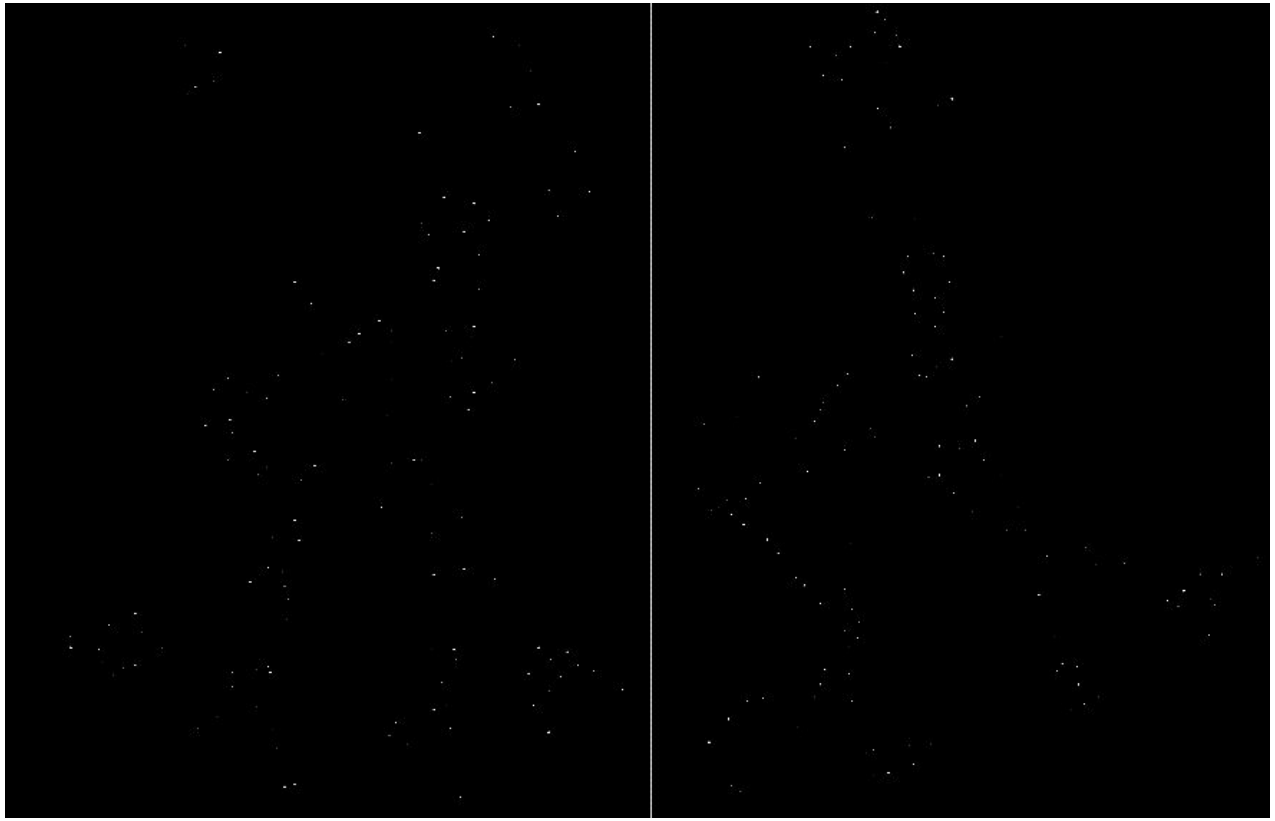


Image Analysis for Computer vision

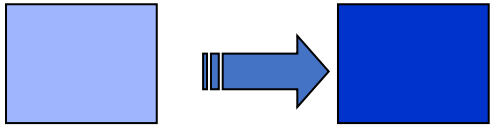
Keypoint Detection and Feature Extraction

Harris corner detector → Super/imposed points

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

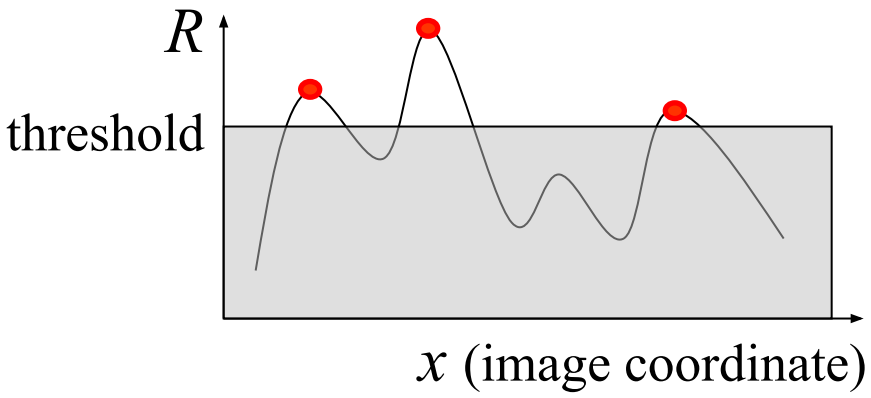
Harris corner detector → Partially invariant to changes in light **intensity**



$$I \rightarrow a\,I + b$$

Derivatives are used, this □ **Invariance to shifts in intensity**

I* → *I* + *b *or in scales of intensity*: ***I* → *a I***



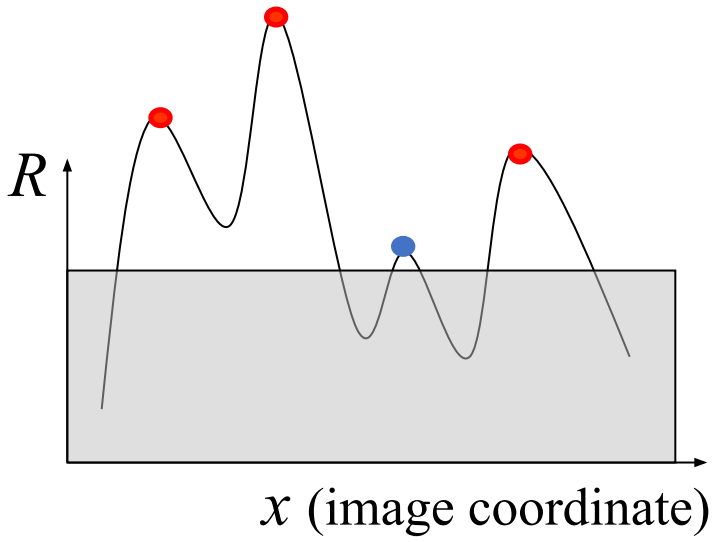


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Harris corner detector

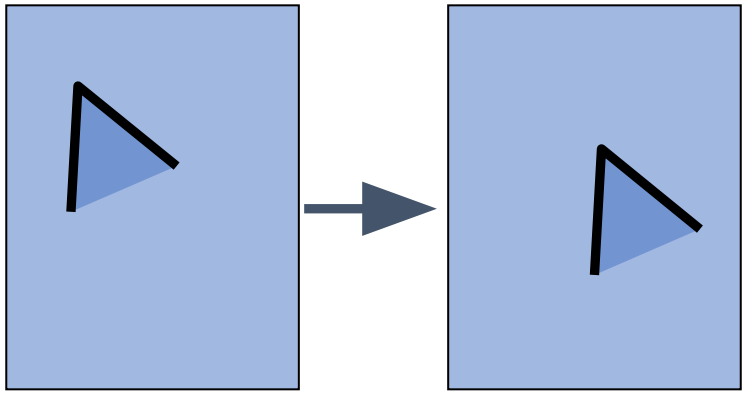


Image Translation

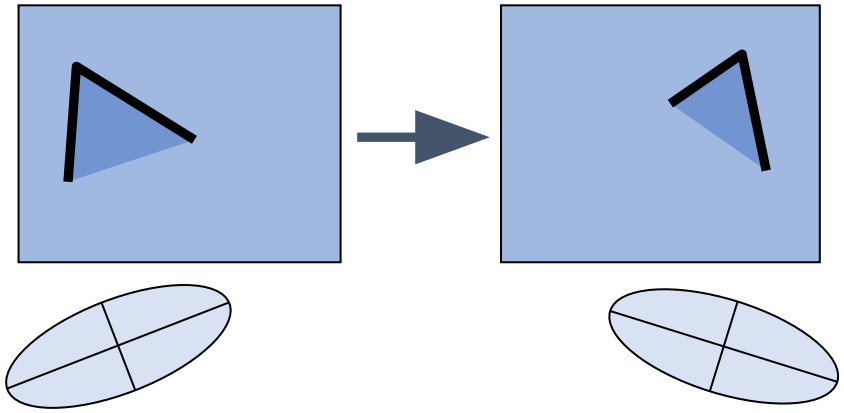


Image Rotation

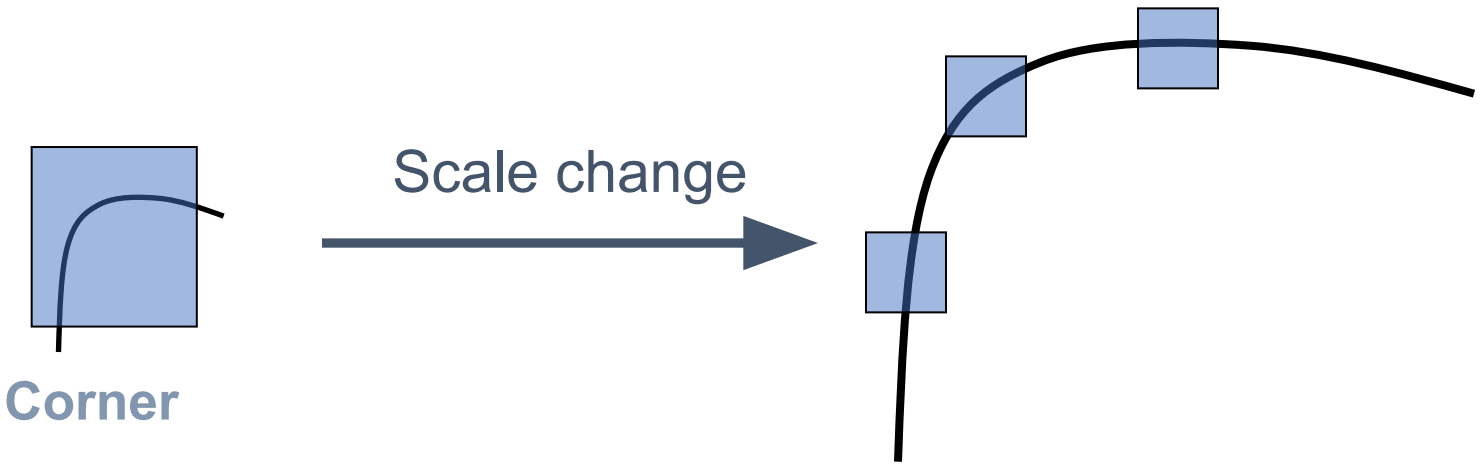
Derivatives and widndow kernerls are “shift-invariant”

La location of a corner is covariant w.r.t. traslation

The slops of moment M rotates but its shape (i.e. eigenvalues) are connstant

La location of a corner is covariant w.r.t. to rotation

Harris corner detector



Every corner or **keypoint** us classified as an **edge**

The location of a corner is not invariant to scale!

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Solución: blob detectors

T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* **30** (2): pp 77--116.

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

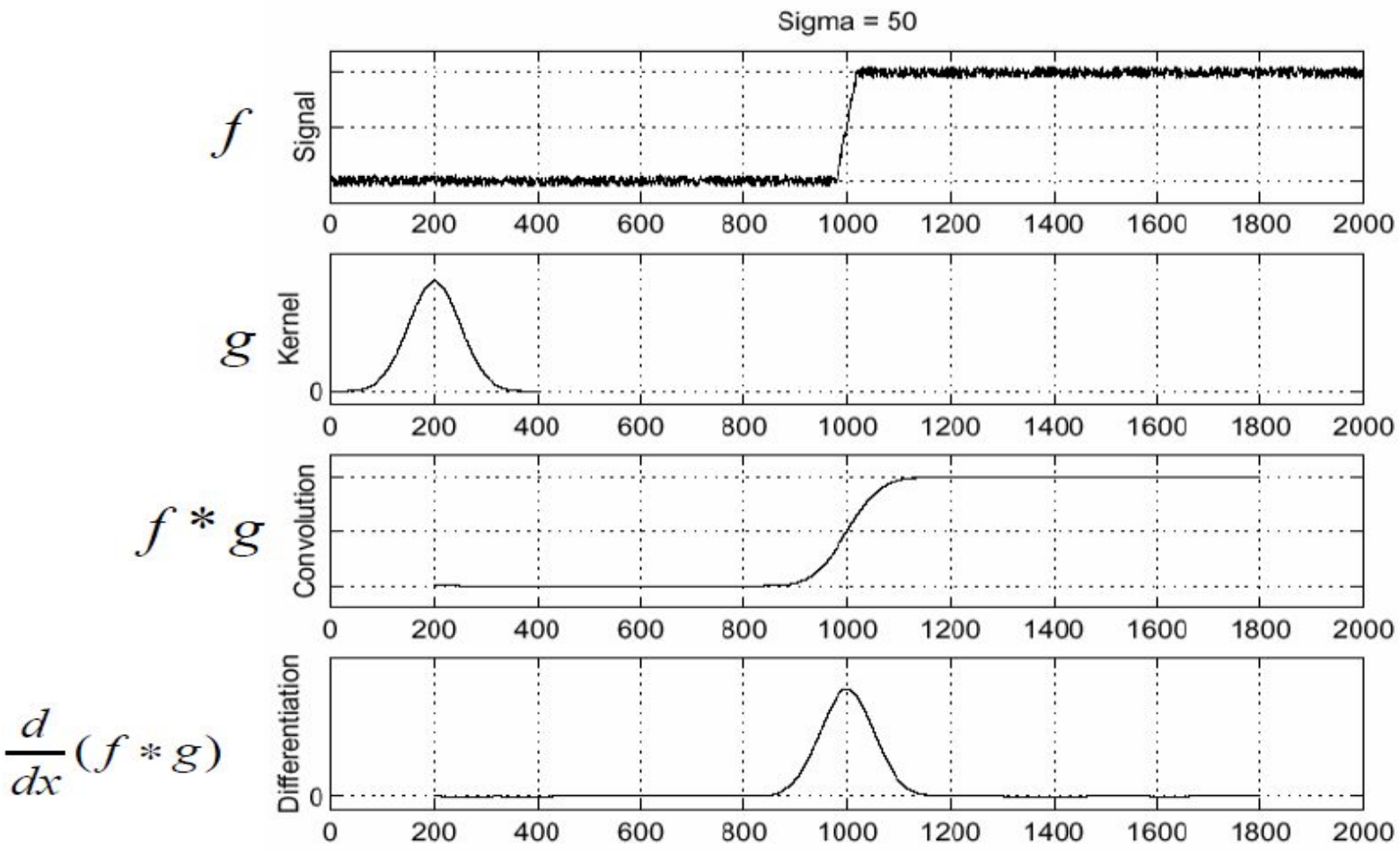
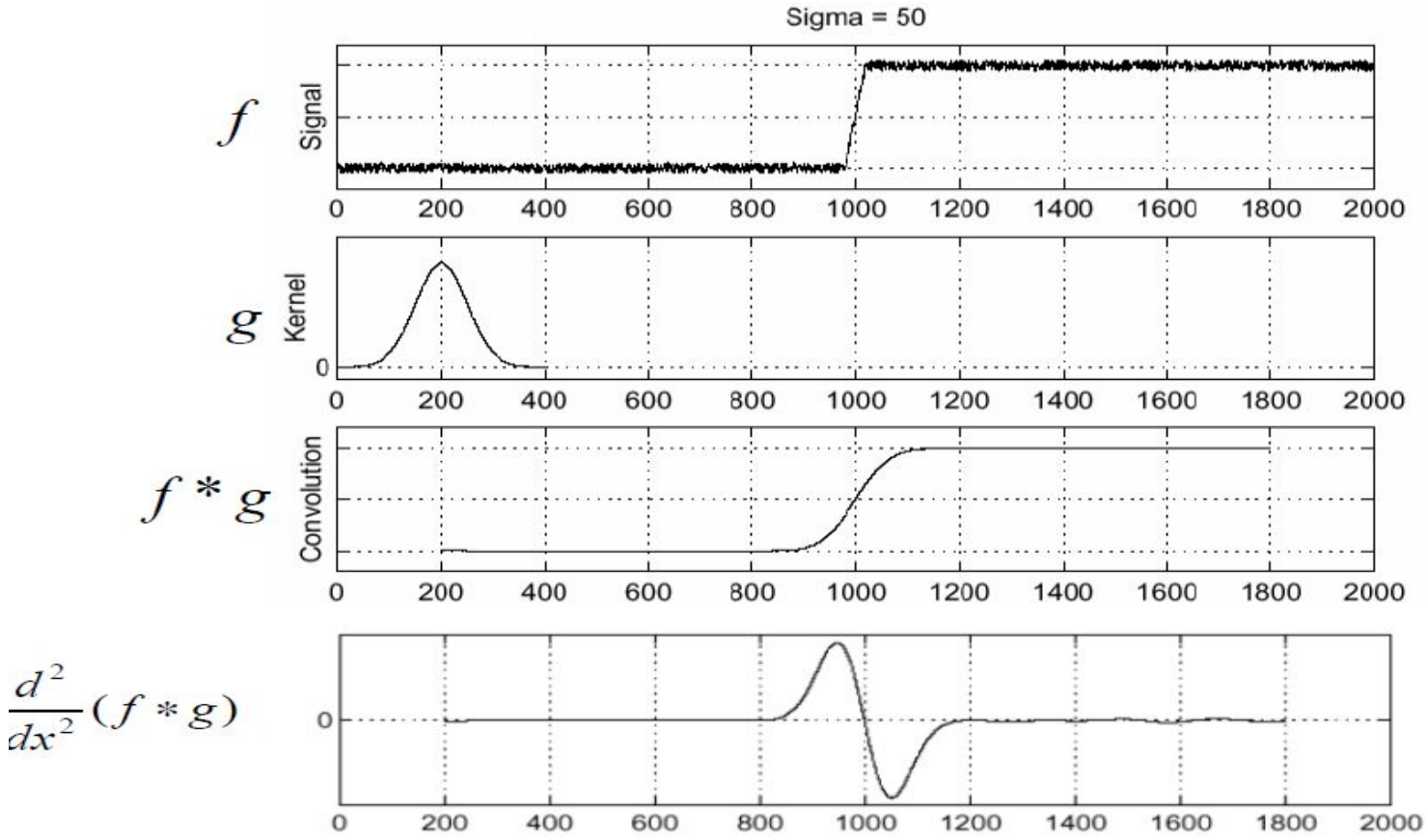


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

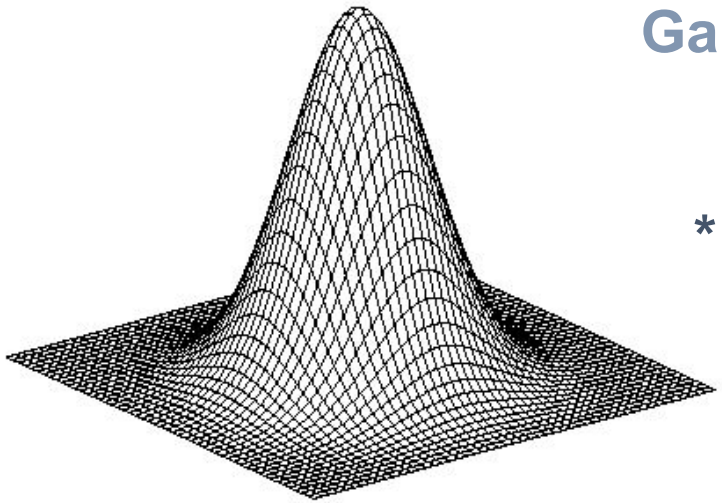


$f * \frac{d^2}{dx^2} g$ = “second derivative of Gaussian” filter = Laplacian of the Gaussian

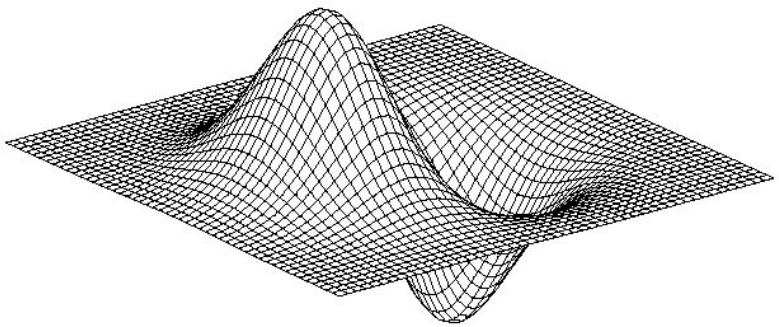
Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Derivative of Gaussian filter



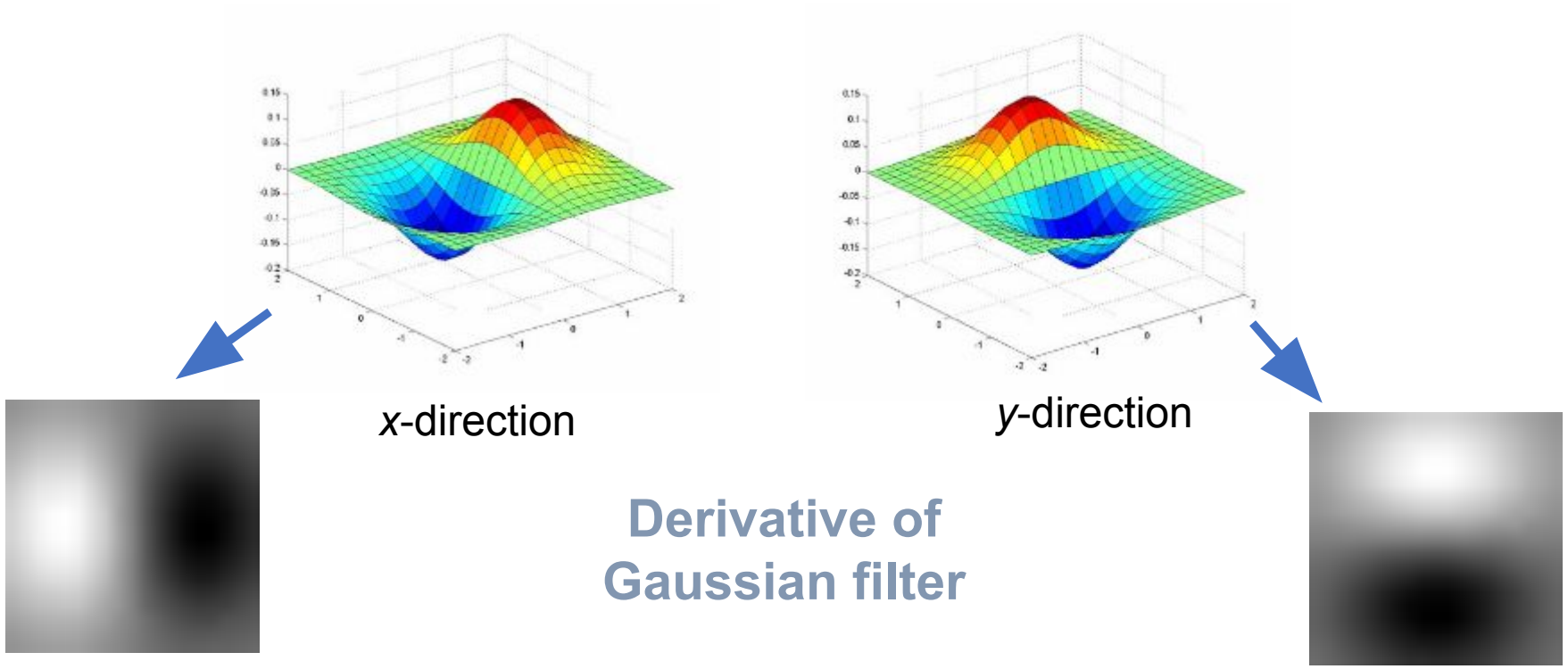
*** [1 -1] =**



J. Canny, A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction



J. Canny, ***A Computational Approach To Edge Detection***, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

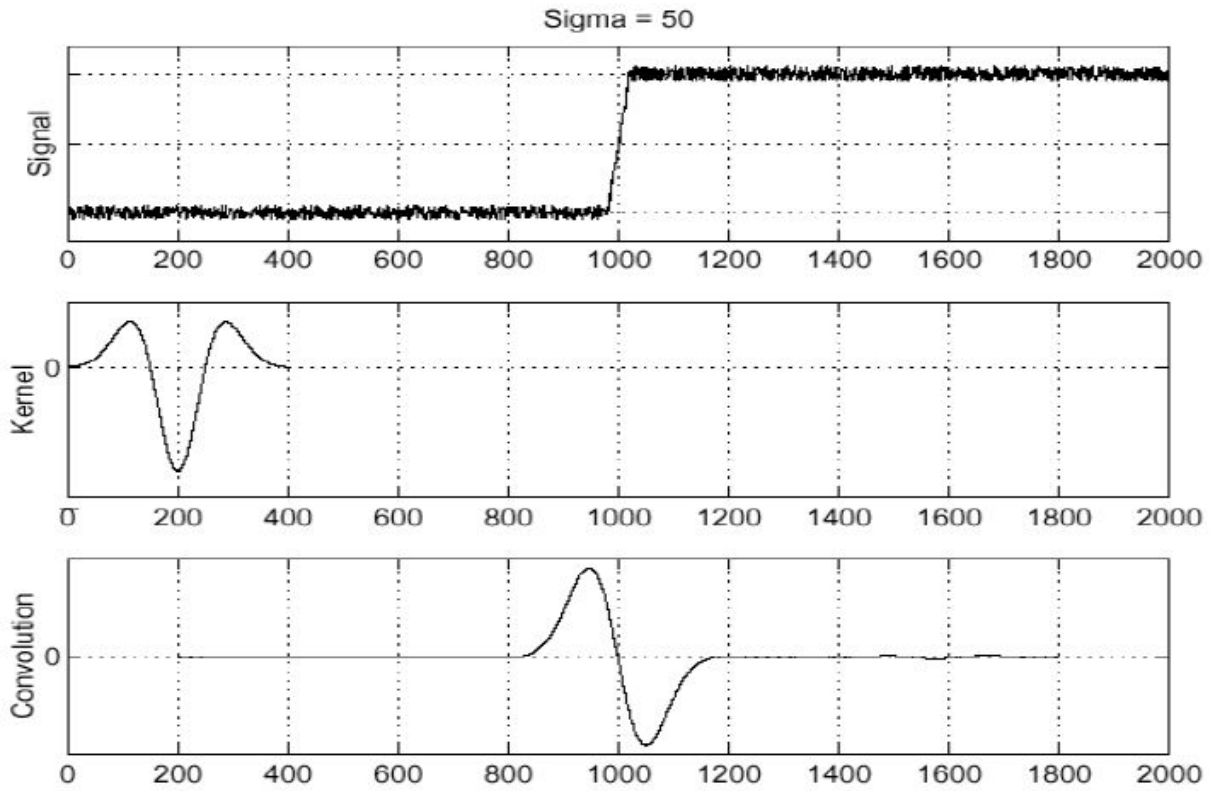
Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

f

$$\frac{d^2}{dx^2} g$$

$$f * \frac{d^2}{dx^2} g$$



Edge

Laplacian

Edge
zero crossing

Edge = zero crossing of the second derivative

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Podemos el Laplaciano para encontrar un “blob”(function rectangular)?

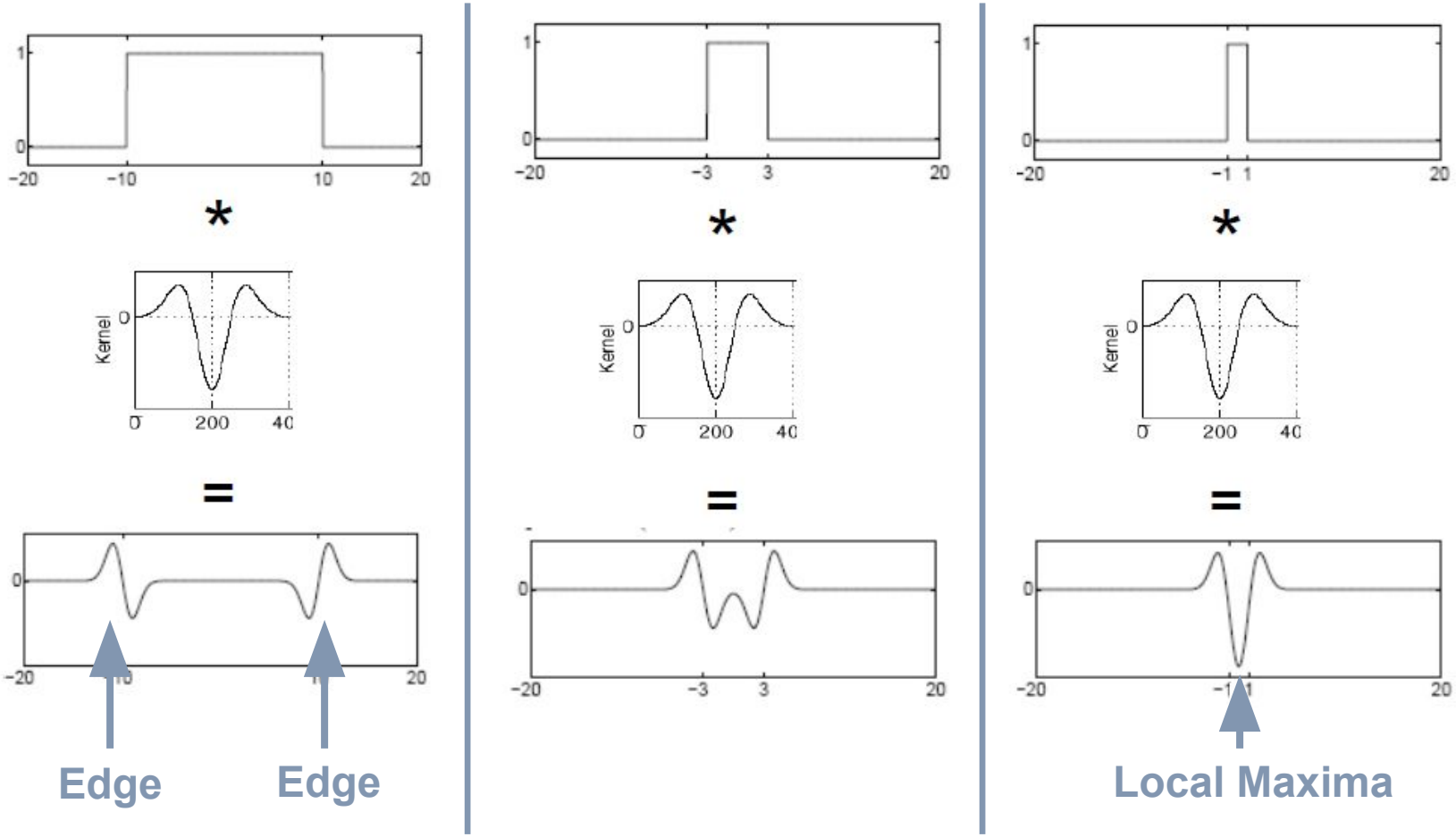
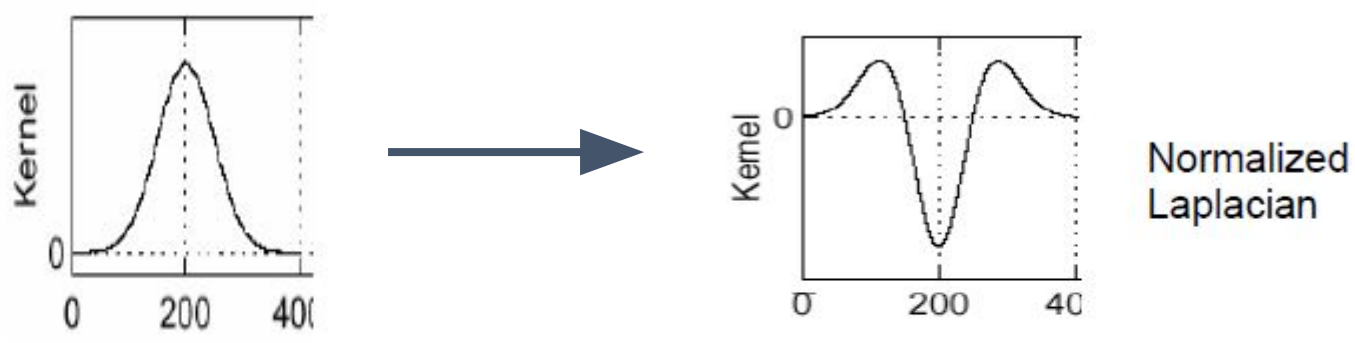


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

What happens as we change the width of the function? □ **Scale selection**

Idea: Convolve the signal with Laplacians at different scales and look for the máxima, how we do that? **Incrementing σ**



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

$$\sigma^2 \frac{d^2}{dx^2} g_n$$

To maintain the energy response constant, the kernel must be multiplied by σ □ **Scale normalization**

- Laplacian is the second derivative of the Gaussian, it must be multiplied by σ^2

Image Analysis for Computer vision

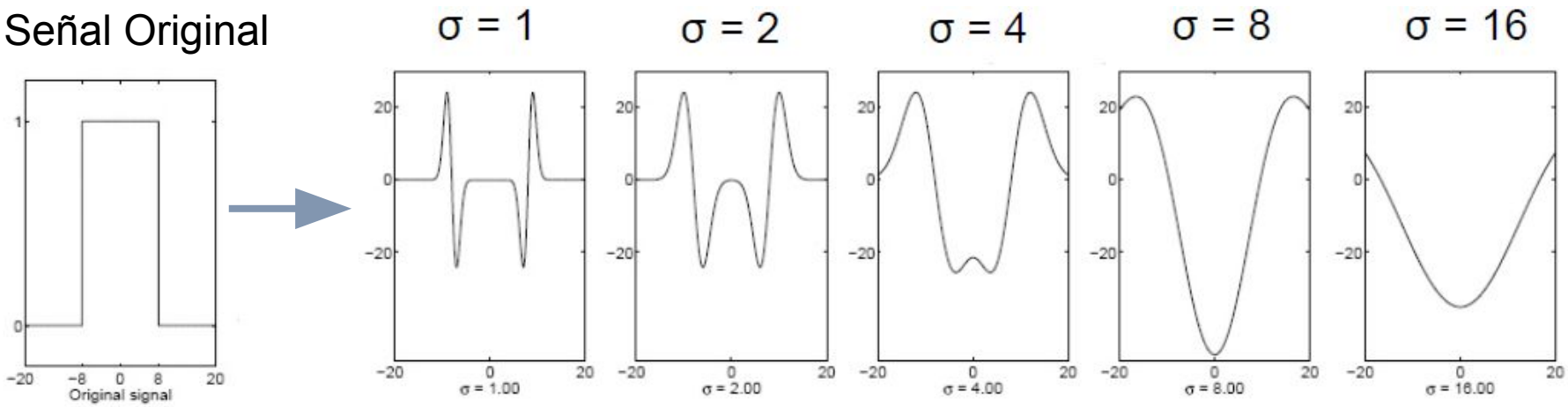
Keypoint Detection and Feature Extraction

Concept: characteristic scale

Local maxima

Response of the **"Scale-normalized Laplacian"**

Señal Original



The **characteristic scale** is that the produces a max in the responses of the Laplacian.
This procedure allows us to

1) Detect un **blob**
2) Estimate the size of a **blob**!

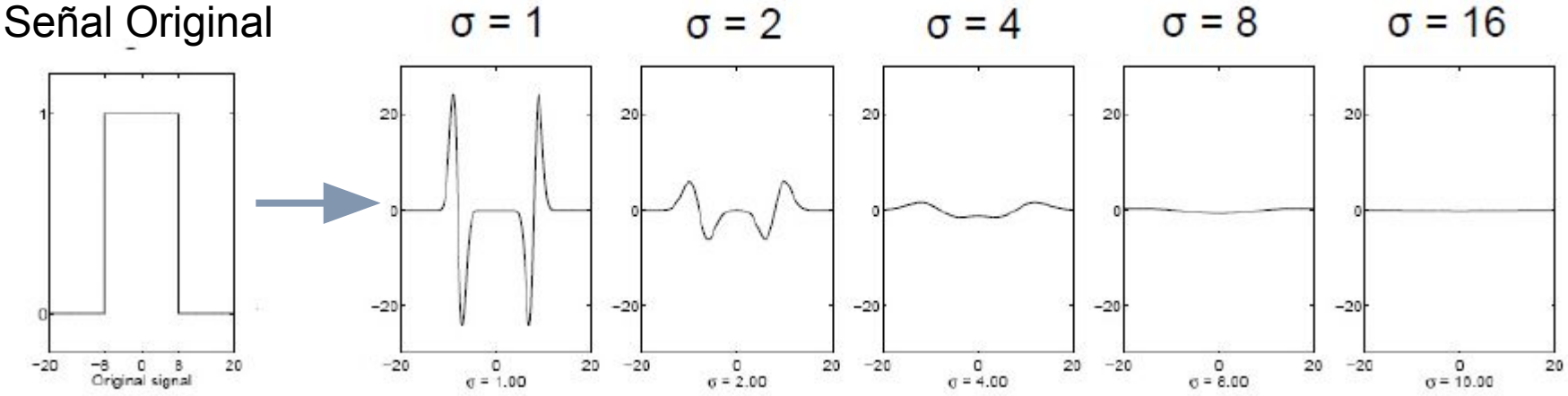
Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Concept: Charactestisc scale

What happens when the Lapalcian is not normalized?

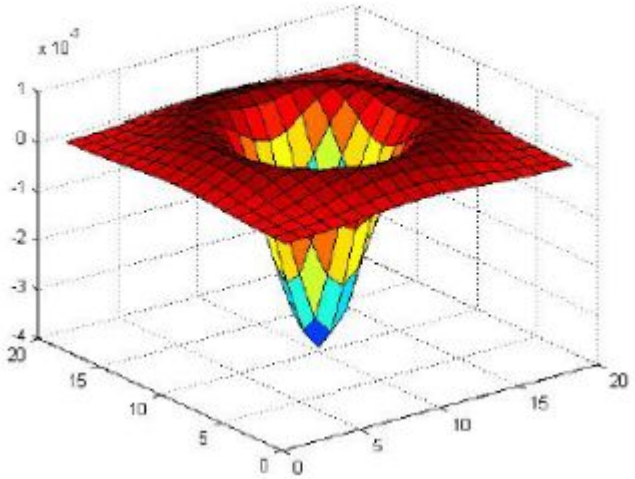
Señal Original

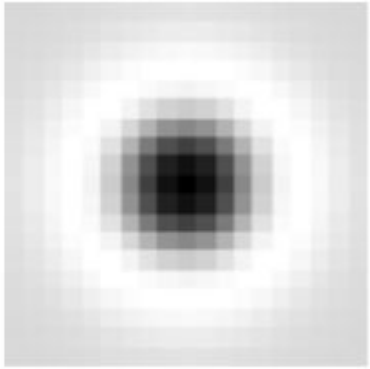


No local maxima

Blob detection in 2D

Laplacian of Gaussian: Symetric Circular Operator, it is used to detect blobs in 2D





Normalized in scale →

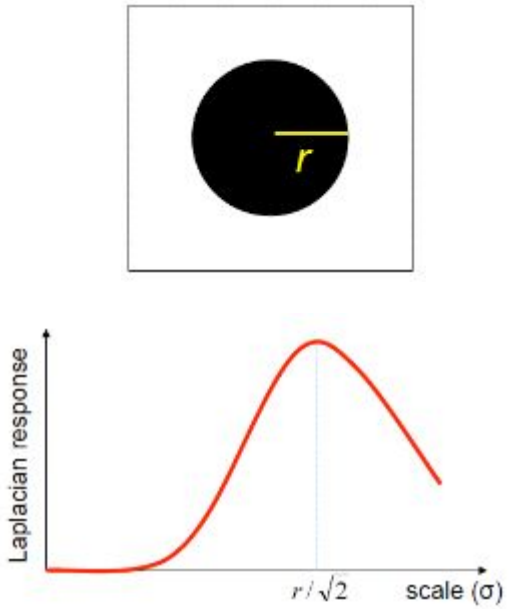
$$\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right) = \sigma^2 (g_{xx} + g_{yy})$$

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Blob detection in 2D → **Scale-space blob detector**

For a binary circle of ratio r, the Laplacian attains a máxima at $\sigma = r / \mathrm{sqrt}(2)$



1. Convolving the image with the Laplacian at different scales, modifying **σ**
2. Find a máxima in the Laplacian response in the space scale

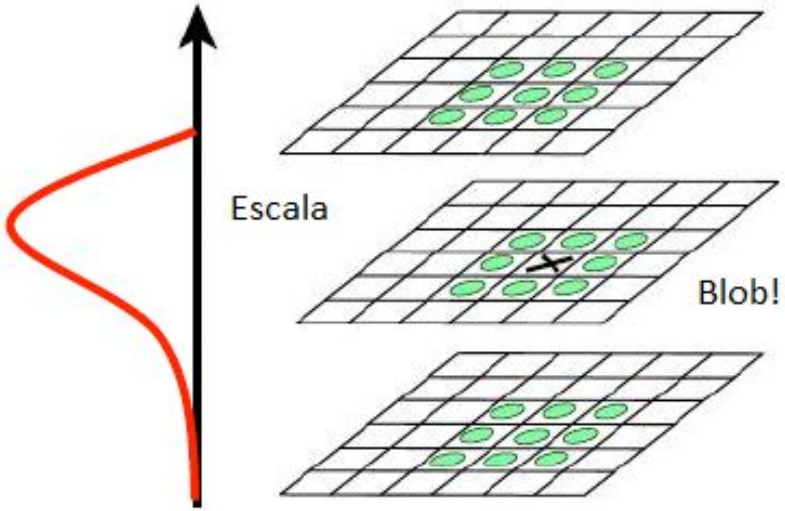
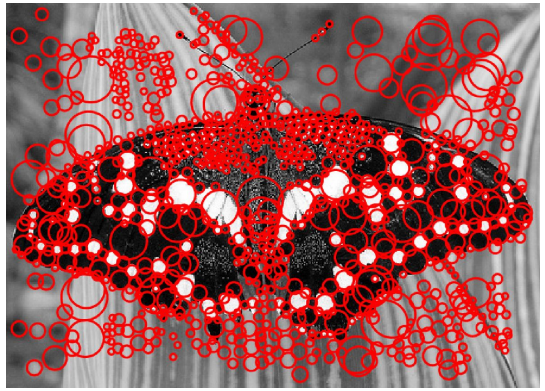


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Blob detection in 2D → **Scale-space blob detector**

sigma = 11.9912



After covolving with various Gaussian kernels of various sizez □

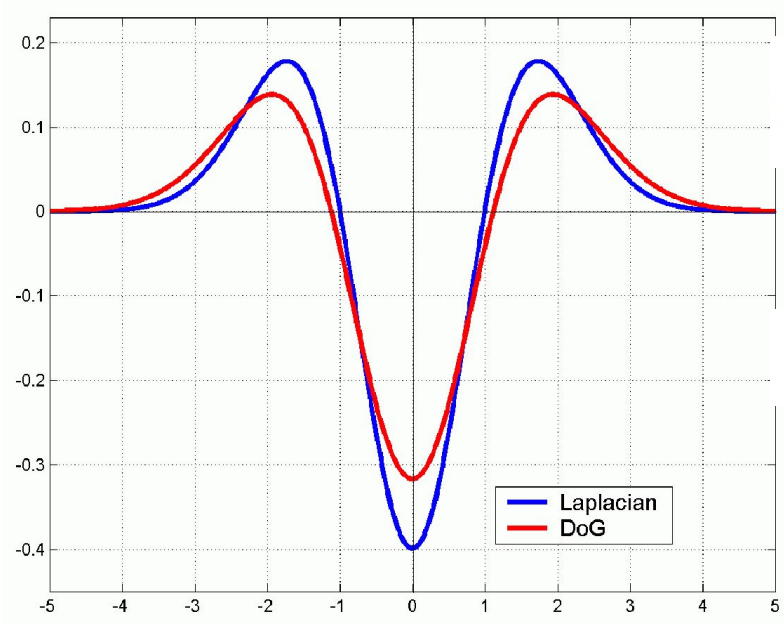
A threshold can be applied to see how many blobs were detected

Image in the rigy shows blobs detected at different scales

Difference of Gaussians (DoG)

David G. **Lowe**. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

Basic Idea: Approximating the Laplacian wti Difference de Gaussianas



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

Laplacian

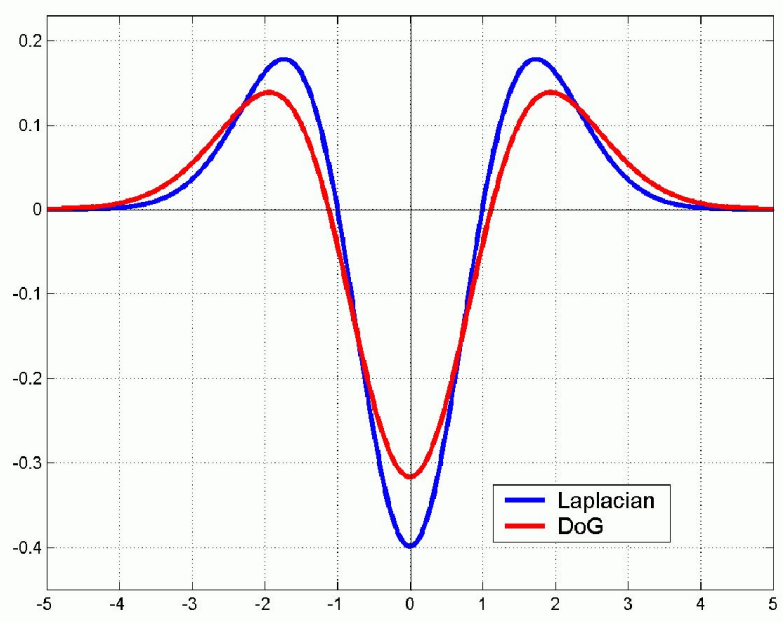
$$DoG = g(x,y,2\sigma) - g(x,y,\sigma)$$

Difference of Gaussians with scales 2σ y σ

Difference of Gaussians (DoG)

David G. **Lowe**. **"Distinctive image features from scale-invariant keypoints."**
IJCV 60 (2), 04

Basic Idea: Approximating the Laplacian wti Difference de Gaussianas



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

$$\approx (k-1)\sigma^2 \mathbf{L}$$

Affine invariant detectors

K. Mikolajczyk and C. Schmid,
Scale and Affine invariant interest point detectors, IJCV 60(1):63-86, 2004.

Similarly to characteristic scale, we can define the **characteristic shape** of a blob

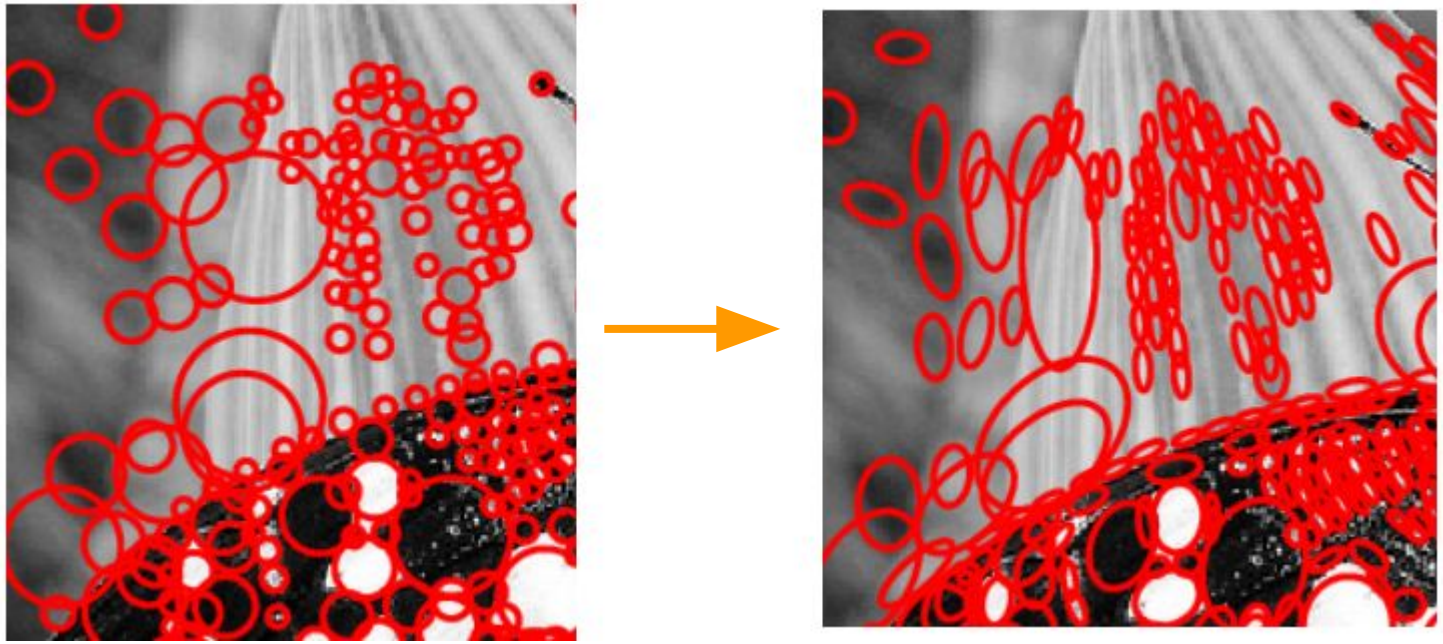


Image Analysis for Computer vision

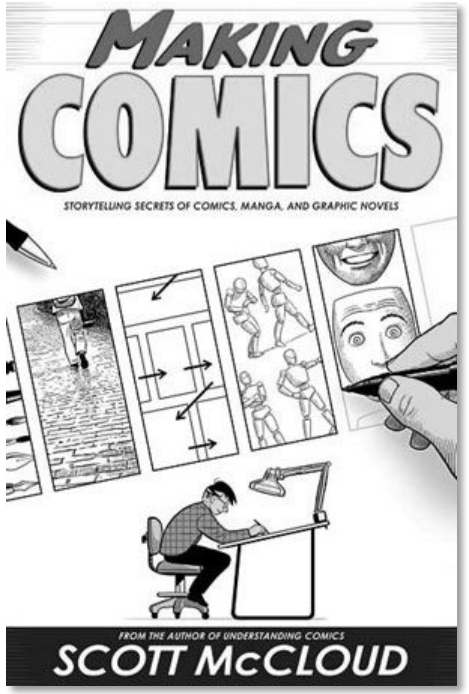
Keypoint Detection and Feature Extraction

Detector	Illumination	Rotation	Scale	View point
Lowe '99 (DoG)	Yes	Yes	Yes	No
Harris corner	Yes	Yes	No	No
Mikolajczyk & Schmid '01, '02	Yes	Yes	Yes	Yes
Tuytelaars, '00	Yes	Yes	No (Yes '04)	Yes
Kadir & Brady, 01	Yes	Yes	Yes	no
Matas, '02	Yes	Yes	Yes	no

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

What is the **geometric relation** between this two images?



¿?

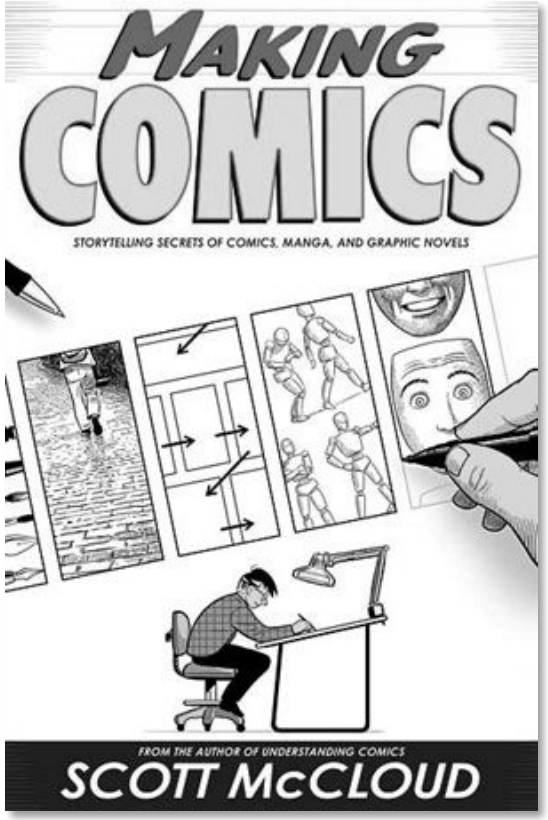
Asnwer: **Similarity transformation** (translation, rotation, uniform scale)

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

What is the **geometric relation** between this two images?

How do we find information about this correspondence?



¡Usando SIFT!

+

Calibración de cámaras

(next class)

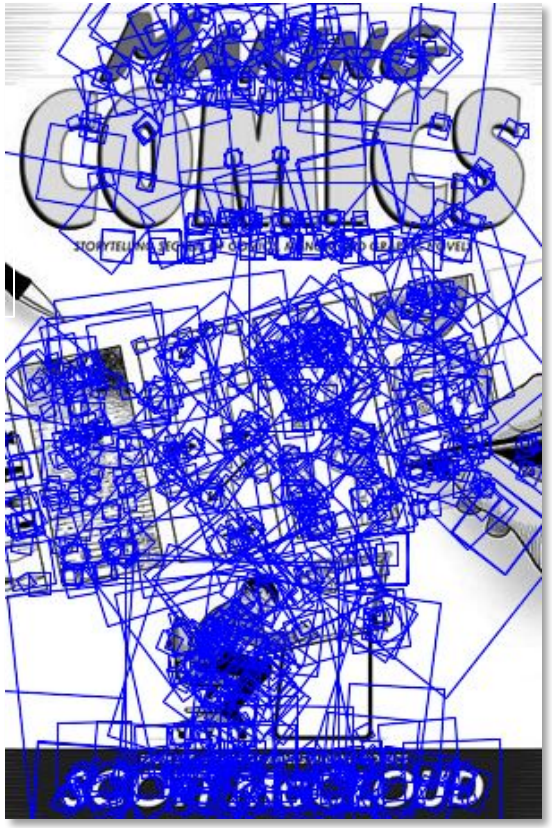
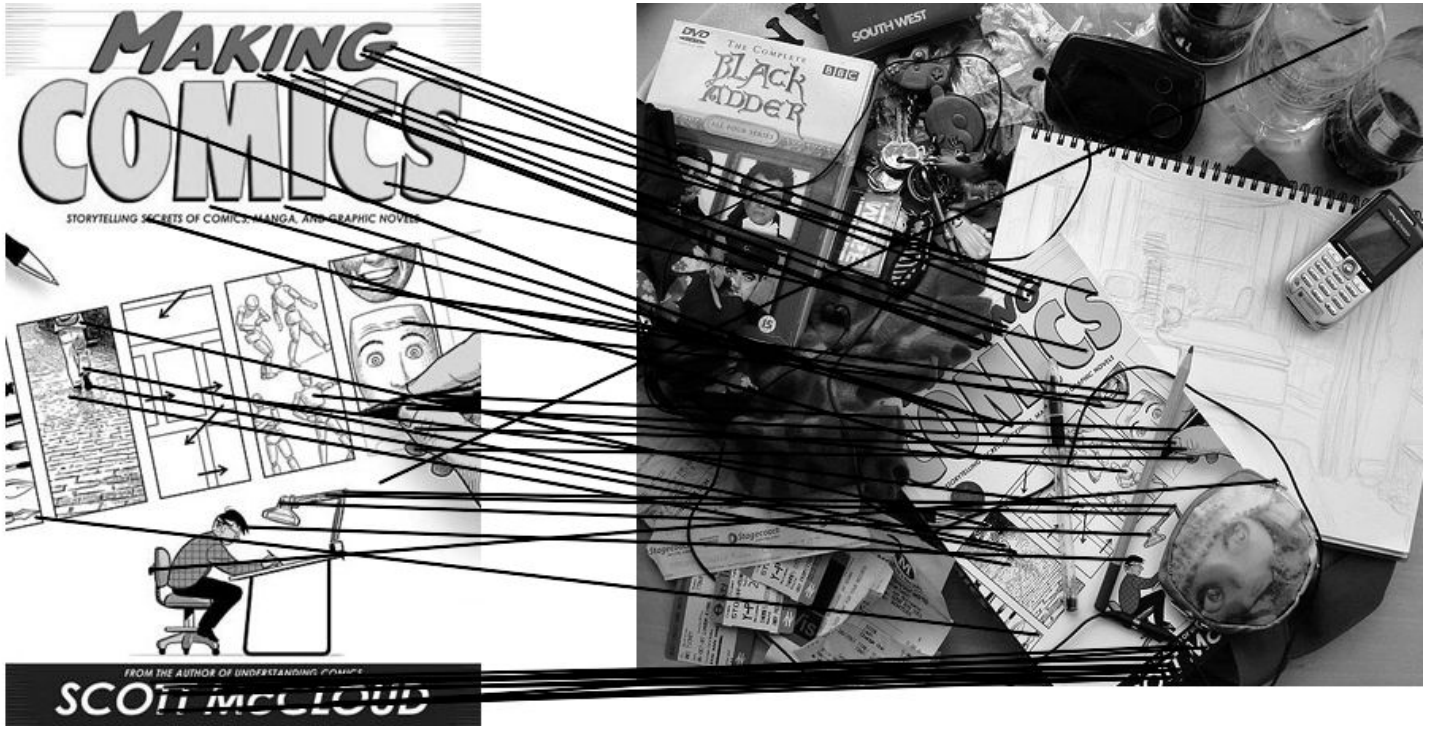


Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

What is the **geometric relation** between this two images?

How do we find information about this correspondence?



We have to do what is known as **feature matching**

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

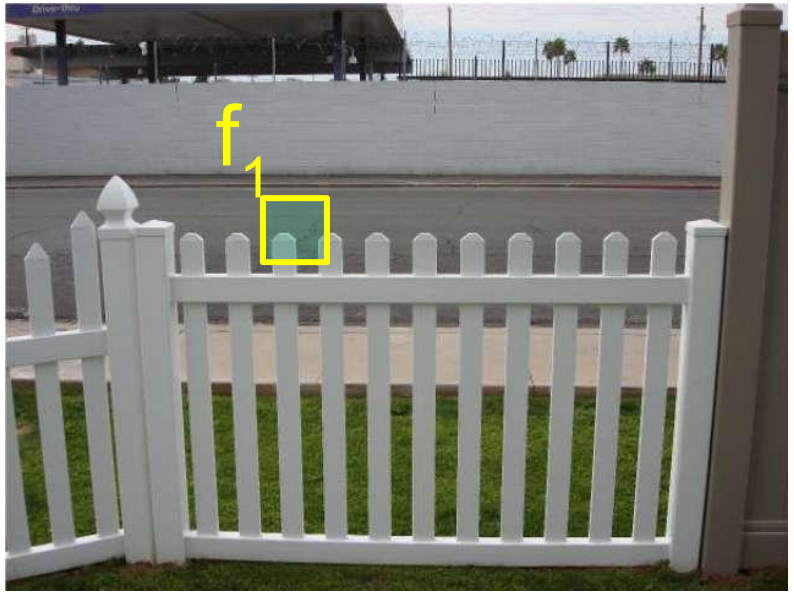
Tenemos que hacer los que se conoce como **feature matching**

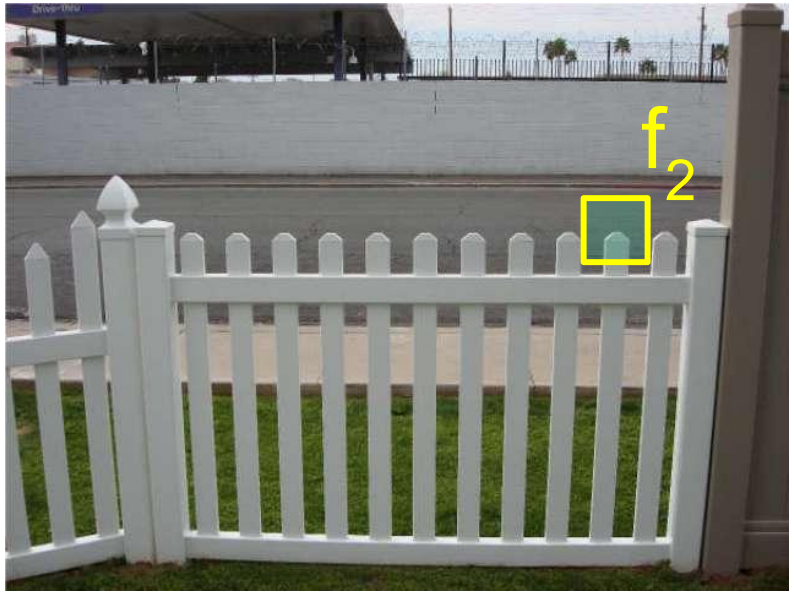
¿Como definimos la **distancia** entre dos **features** f_1, f_2? □

Acercamiento sencillo: Distancia L_2, $||f_1 - f_2||$

Puede dar buenos "scores" a matches ambiguos □ incorrectos

I_1





I_2

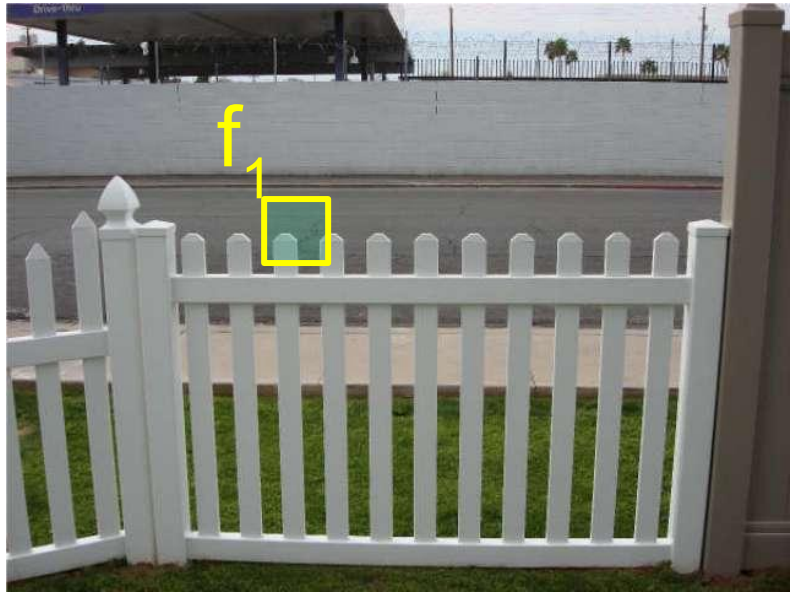
Image Analysis for Computer vision

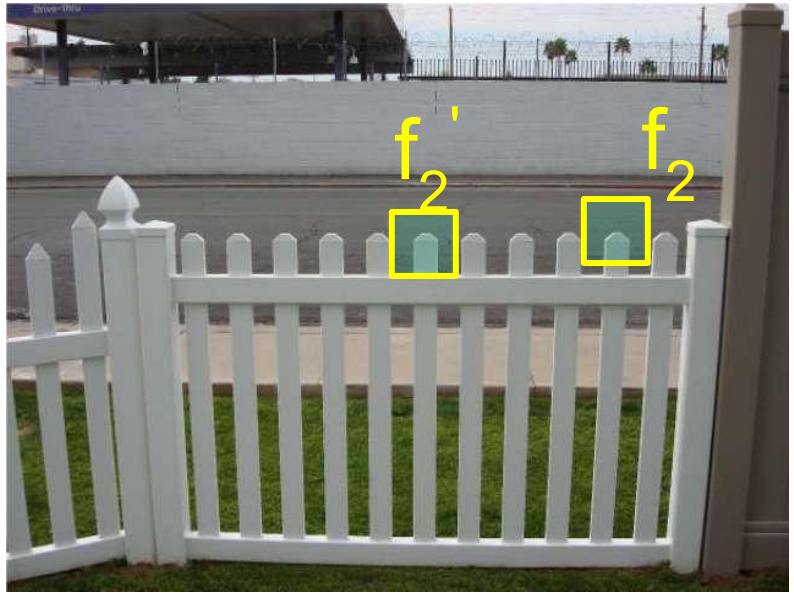
Keypoint Detection and Feature Extraction

Tenemos que hacer los que se conoce como **feature matching**

- Mejor acercamiento: distancia de los radios= $||f_1 - f_2|| / ||f_1 - f_2'||$
- **Sum of Squared Differences (SSD)**
  - $\mathbf{f_2}$ es el 2do SSD match para $\mathbf{f_1}$ en I_2 y $\mathbf{f_2'}$ es el 2do
  - Da valores altos para mathes ambibuos

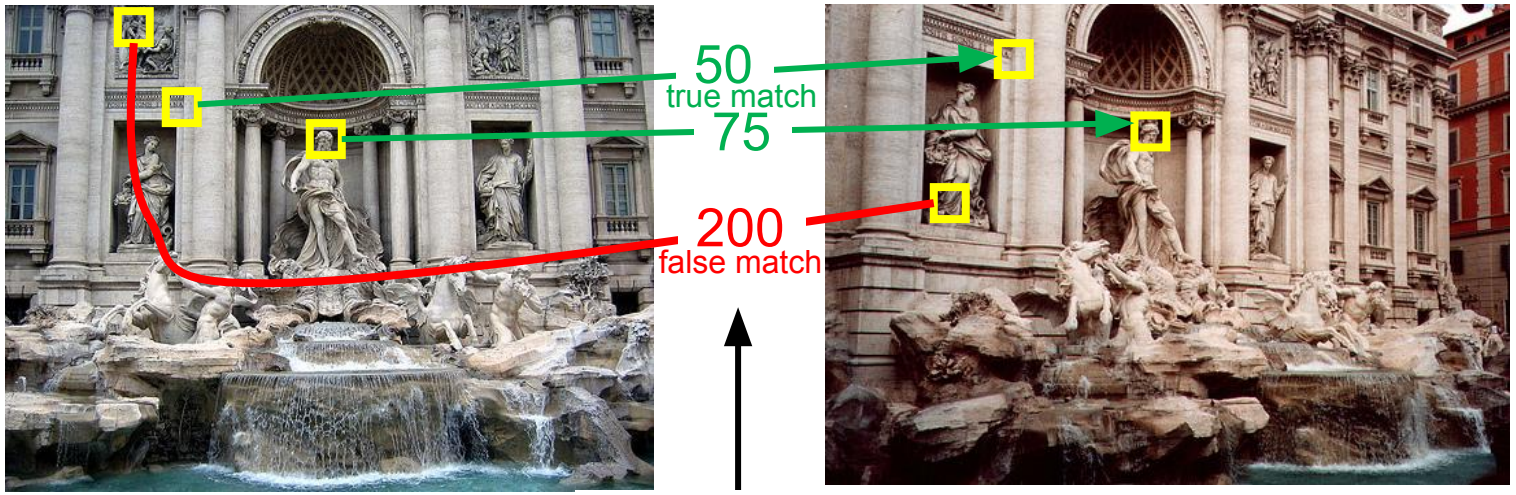
I_1





I_2

¿Como podemos medir el **performance** de un feature matcher?



feature distance

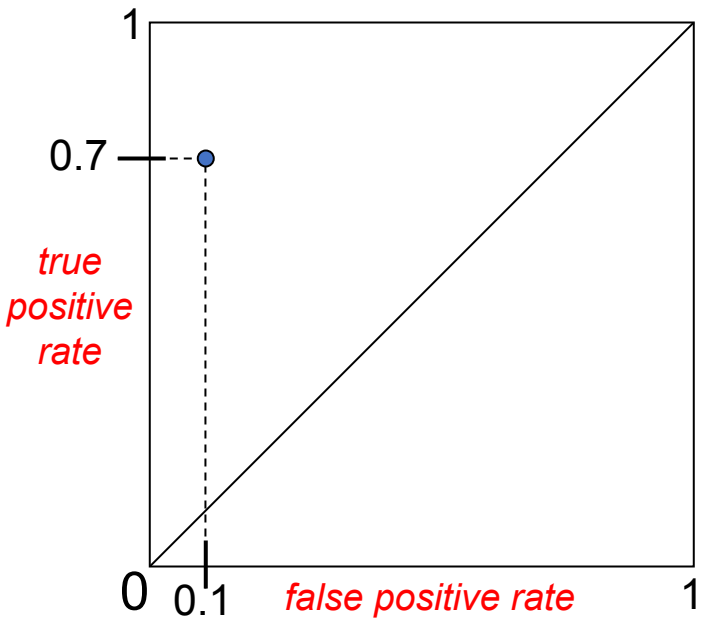
Aplicando un **threshold** afecta el performance

- **True positives** = # matched detectados que son correctos
- **False positives** = # maches detectados que son incorrectos

¿Como podemos medir el **performance** de un feature matcher?

$$\frac{\text{\# true positives}}{\text{\# correctly matched features (positives)}}$$

"recall"



$$\frac{\text{\# false positives}}{\text{\# incorrectly matched features (negatives)}}$$

1 - "precision"

Sum of Squared Differences

Otras opciones existen

Normalized Cross-Correlation

Ver ejemplo en libro
Computer Vision with Python (pag 32)

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

¿Como podemos medir el **performance** de un feature matcher?

$$\frac{\text{\# true positives}}{\text{\# correctly matched features (positives)}}$$

"recall"

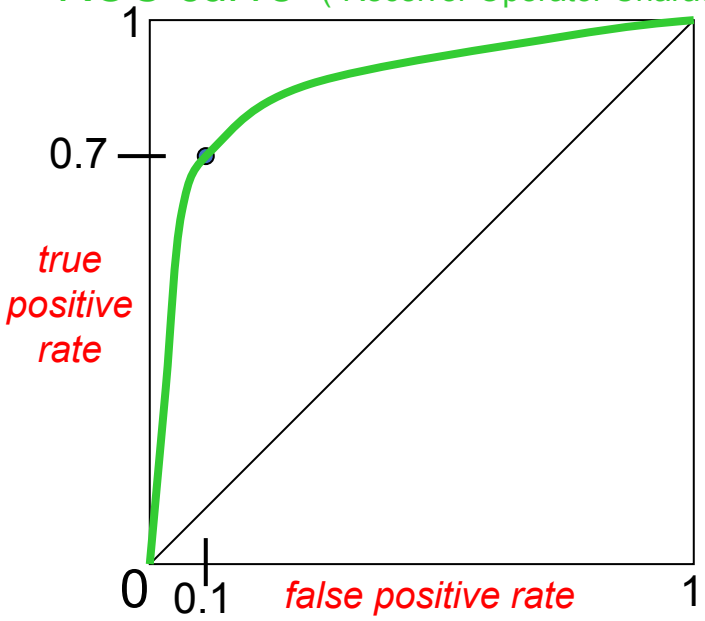
Sum of Squared Differences

Otras opciones existen

Normalized Cross-Correlation

Ver ejemplo en libro
Computer Vision with Python (pag 32)

ROC curve ("Receiver Operator Characteristic")



$$\frac{\text{\# false positives}}{\text{\# incorrectly matched features (negatives)}}$$

1 - "precision"

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

Detector	Illumination	Rotation	Scale	View point
Lowe '99 (DoG)	Yes	Yes	Yes	No
Harris corner	Yes	Yes	No	No
Mikolajczyk & Schmid '01, '02	Yes	Yes	Yes	Yes
Tuytelaars, '00	Yes	Yes	No (Yes '04)	Yes
Kadir & Brady, 01	Yes	Yes	Yes	no
Matas, '02	Yes	Yes	Yes	no

Los detectores son importantes para encontrar zonas de interes pero no son informativos se necesitan descriptores locales (proxima clase

Image Analysis for Computer vision

Keypoint Detection and Feature Extraction

SUSAN: Smallest Univalue Segment Assimilating Nucleus, Smith and Brady, 1997

SURF: Speeded Up Robust Features, Tuytelars & Van Gool, 2005

GLOH: Gradient Location and Orientation Histogram, K. Mikolajczyk, 2005

FAST (corner detector): A machine learning approach to corner detection, 2006

ORB: An efficient alternative to SIFT or SURF, Rublee, 2011

FREAK: Fast Retina Keypoint, Alahi, 2012

BRISK: Binary Robust Invariant Keypoint, 2013

¡Presentacion 5ta clase!

LIFT: Learned Invariant Feature Transform, Fua 2016

SuperPoint: Self Supervised Interest Point Detection, 2018

Copyright 2020. Tecnologico de Monterrey.
The total or partial reproduction of this work is
prohibited without explicit authorization of Tecnologico
de Monterrey