DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Procesamiento y Análisis de Imágenes

Violeta Chang

violeta.chang@usach.cl

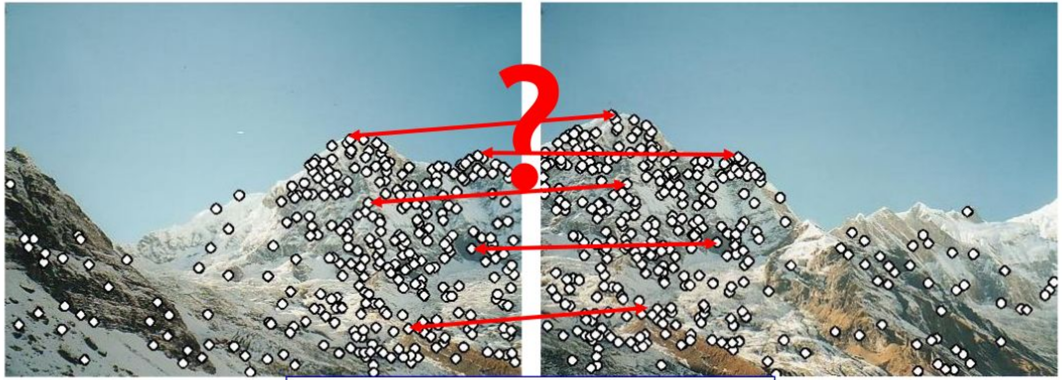
Créditos por slides: José Saavedra, Juan Carlos Niebles, Domingo Mery, David Lowe

Descriptores de características locales

Local Descriptors

- We know how to detect points
- Next question:

How to *describe* them for matching?

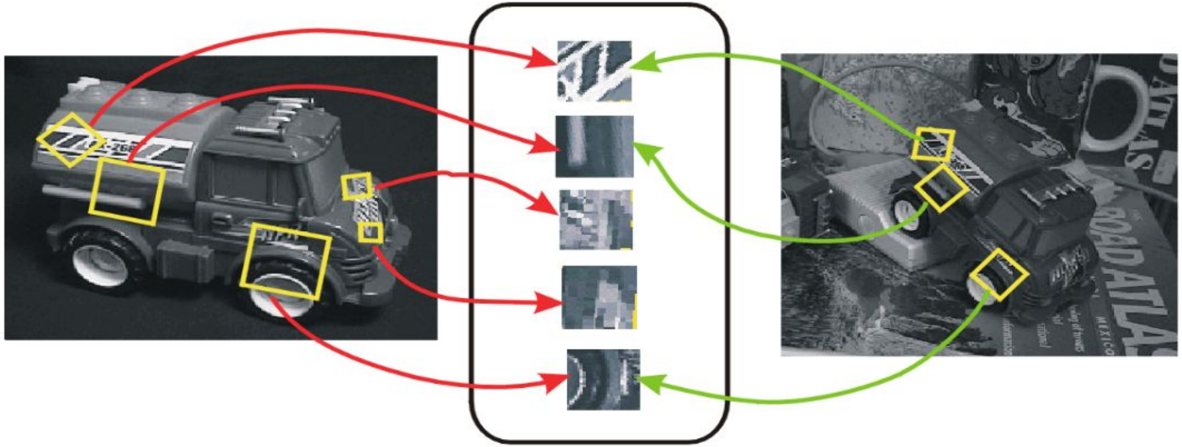


Point descriptor should be:
1. Invariant
2. Distinctive

Descriptores de características locales

Invariant Local Features

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



Descriptores de características locales

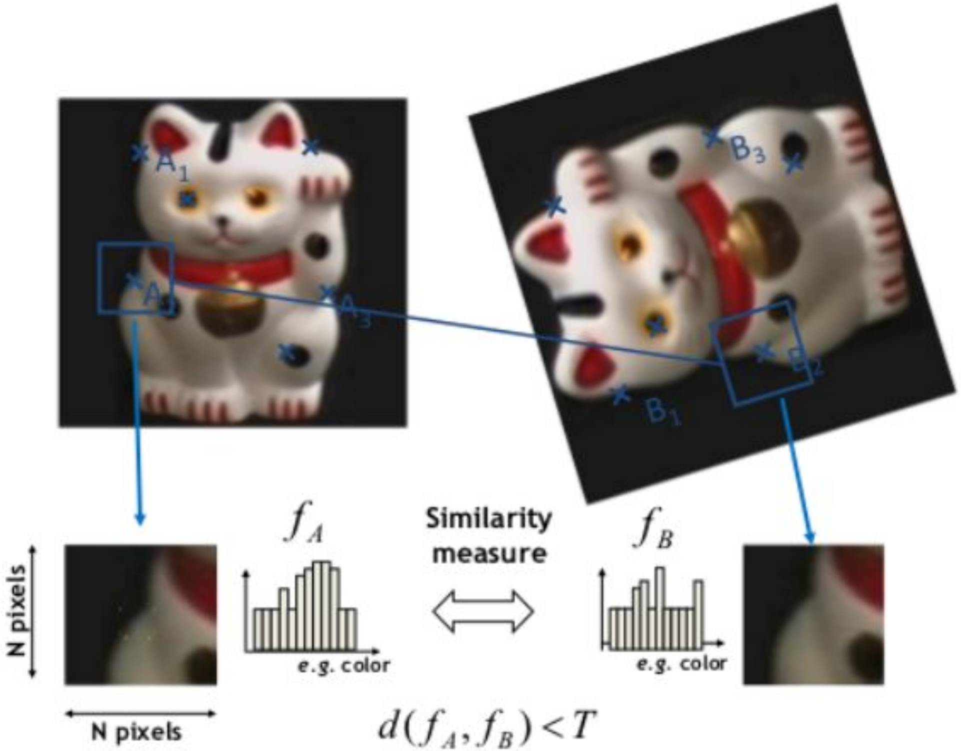
1. Detectar *keypoints*

2. Definir una región local (invariante a escala)

3. Normalizar el contenido de la región

4. Describir la región (Descriptores locales)

5. *Matching* de descriptores

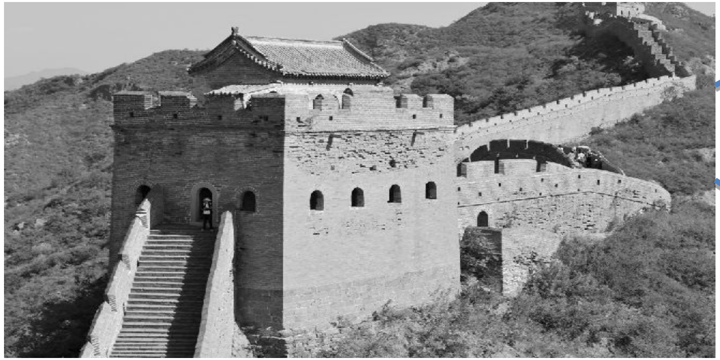


Descriptores de características locales

Advantages of invariant local features

- Locality: features are local, so robust to occlusion and clutter (no prior segmentation)
- Distinctiveness: individual features can be matched to a large database of objects
- Quantity: many features can be generated for even small objects
- Efficiency: close to real-time performance
- Extensibility: can easily be extended to wide range of differing feature types, with each adding robustness

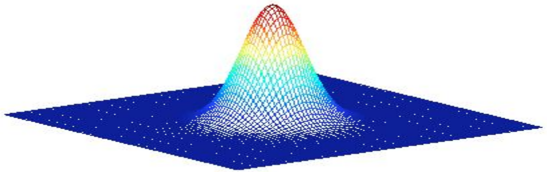
Histogram of gradients



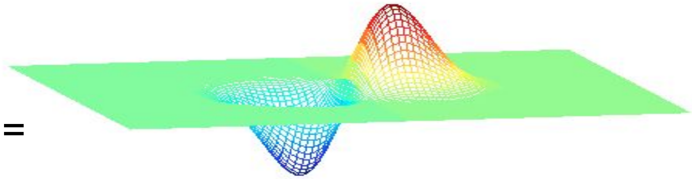
Gradient in x direction → G_x

Gradient in y direction → G_y

σ'



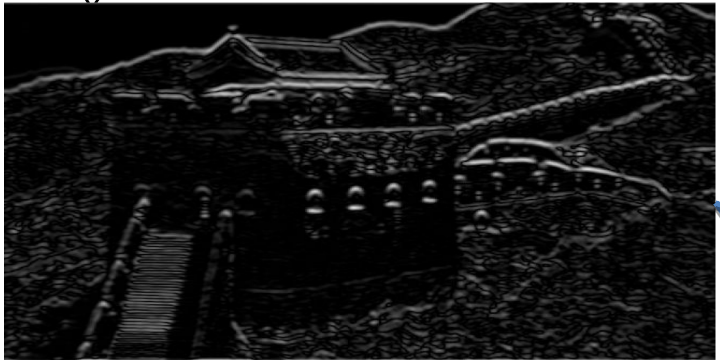
★ [-1 0 1] =



★ : Convolution

Histogram of gradients

G_x: Gradient in x



G_y: Gradient in y

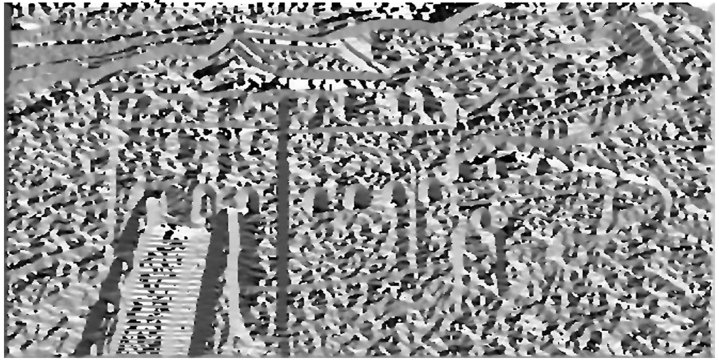
Magnitude → $R=\mathrm{sqrt}(G_x^2+G_y^2)$

Angle → $A=\arctan(G_y/G_x)$

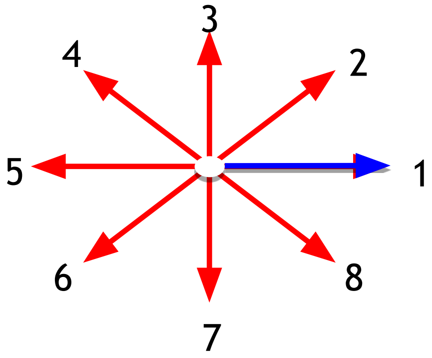
Histogram of gradients

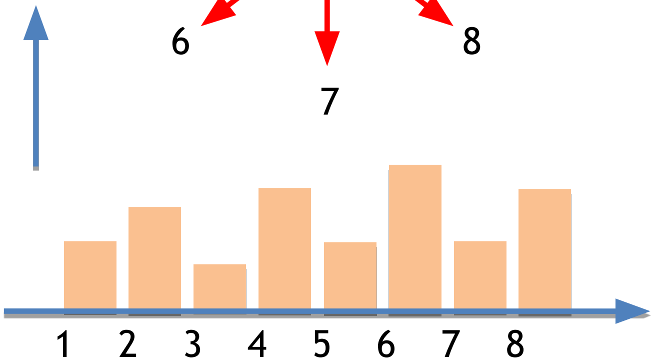
R: Magnitude

A: Angle



Histogram of 8 directions

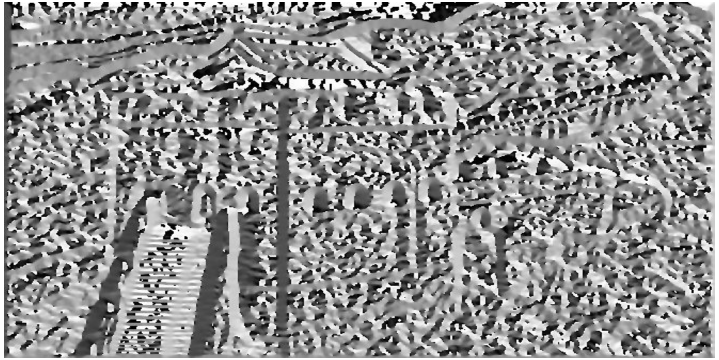




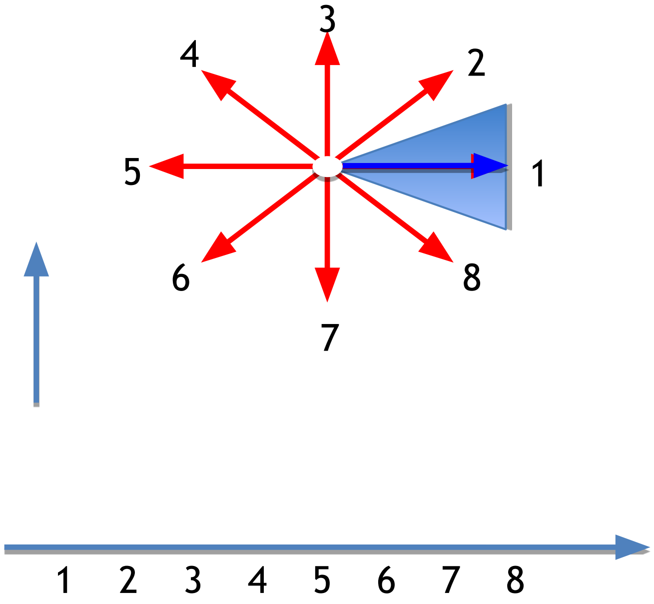
Histogram of gradients

R: Magnitude

A: Angle



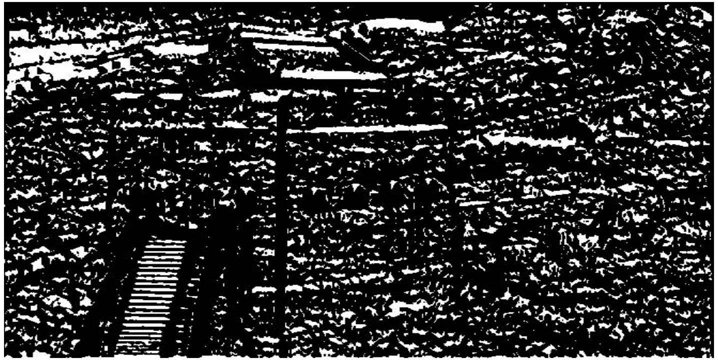
Histogram of 8 directions
(computation of first bin)



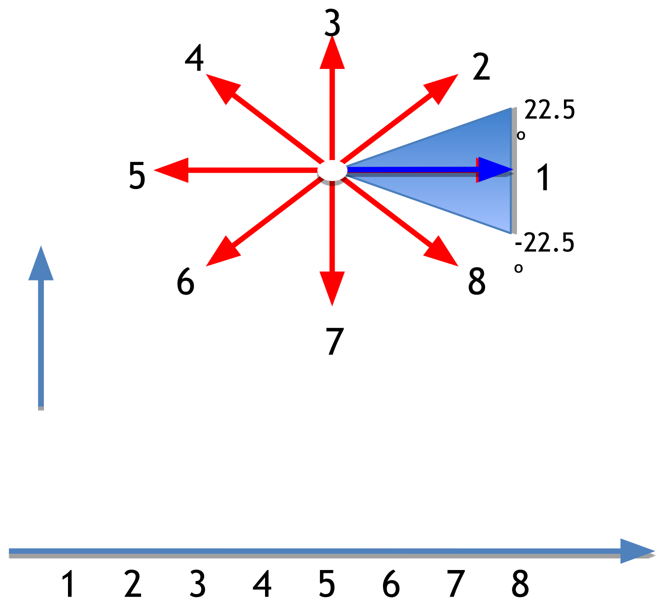
Histogram of gradients

R: Magnitude

A: Angle between -22.5° and 22.5°

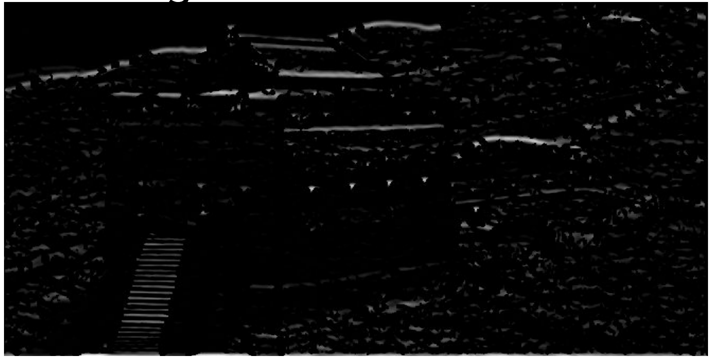


Histogram of 8 directions
(computation of first bin)

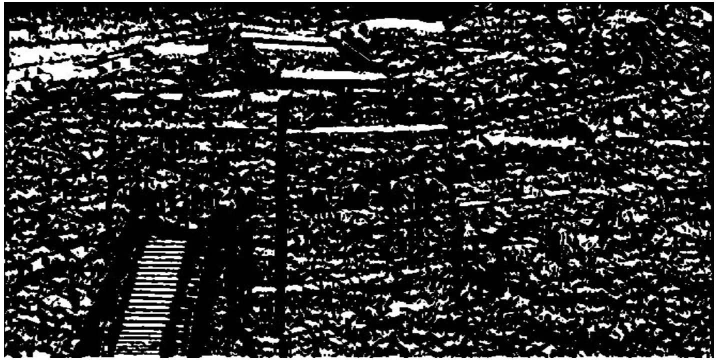


Histogram of gradients

R: Magnitude in this direction

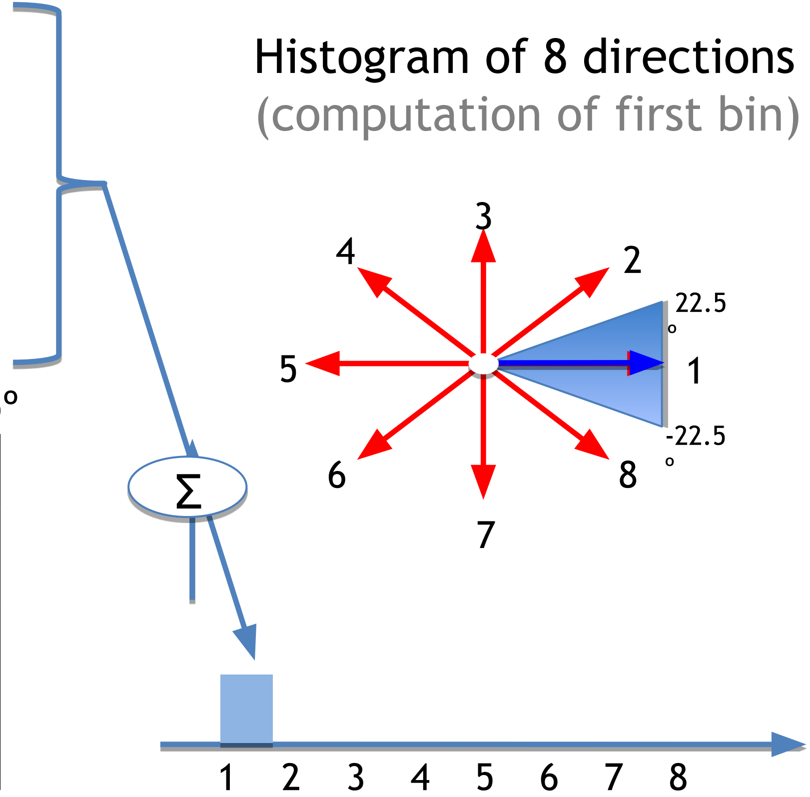


A: Angle between -22.5º and 22.5º



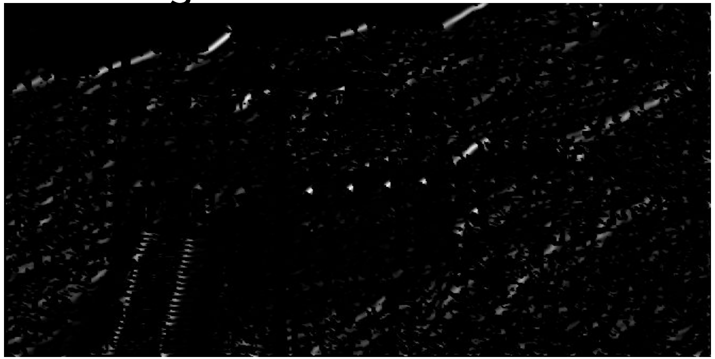
Histogram of 8 directions

(computation of first bin)

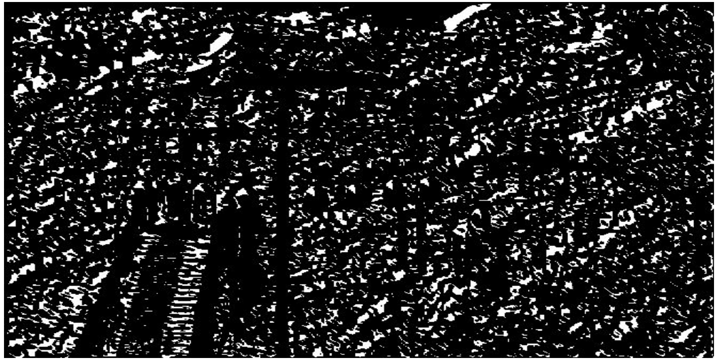


Histogram of gradients

R: Magnitude in this direction

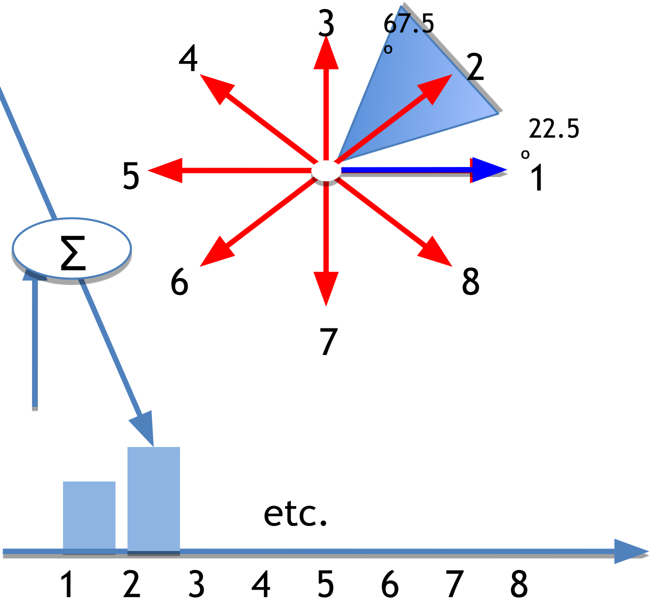


A: Angle between 22.5° and 67.5°



Histogram of 8 directions

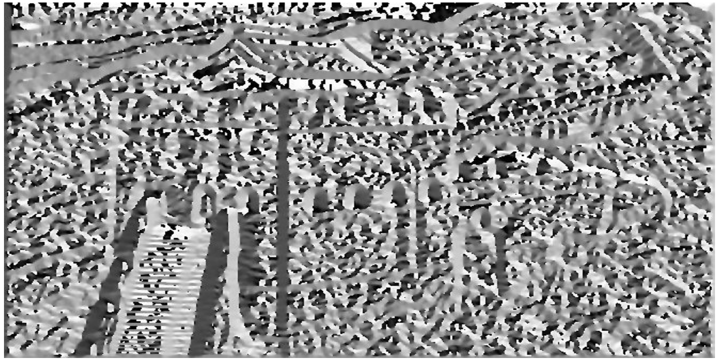
(computation of second bin)



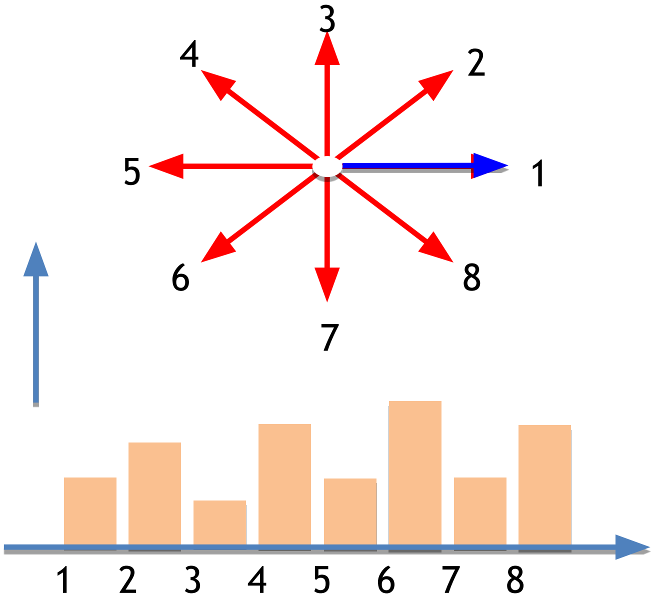
Histogram of gradients

R: Magnitude

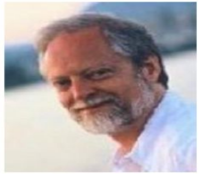
A: Angle



Histogram of 8 directions



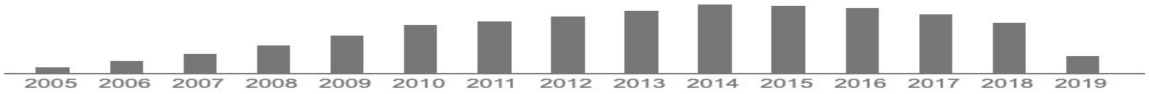
SIFT: Scale Invariant Feature Transform



David Lowe

Distinctive image features from scale-invariant keypoints

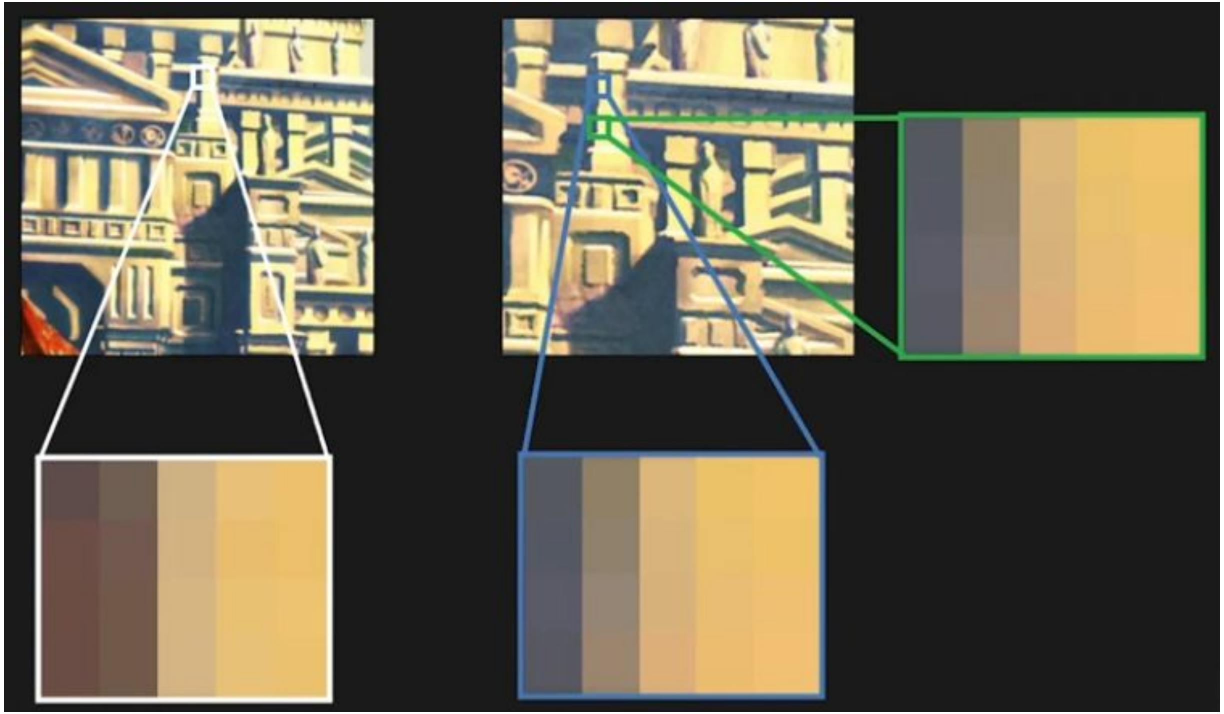
Authors	David G Lowe
Publication date	2004/11/1
Journal	International journal of computer vision
Volume	60
Issue	2
Pages	91-110
Publisher	Springer Netherlands
Description	This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through ...
Total citations	Cited by 50930



D. G. Lowe. [Distinctive image features from scale-invariant features](). Intl. Journal of Computer Vision, 60(2):91-110, 2004.

SIFT: Scale Invariant Feature Transform

¿Por qué no usar simplemente los bordes?

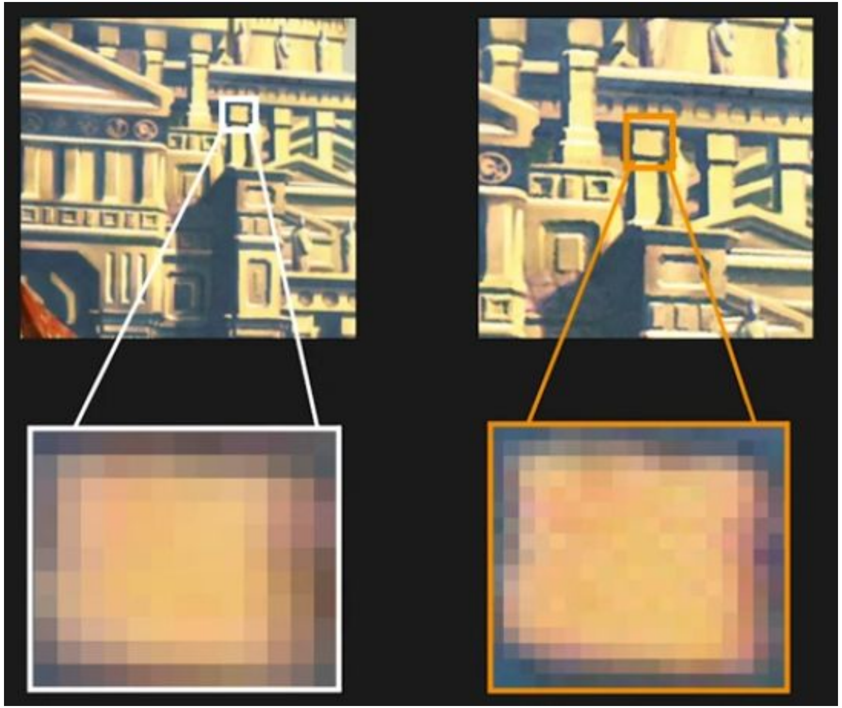


No nos permite localizar un objeto complejo en el espacio.

SIFT: Scale Invariant Feature Transform

Estaremos interesados en la detección de **keypoints** (o blobs):

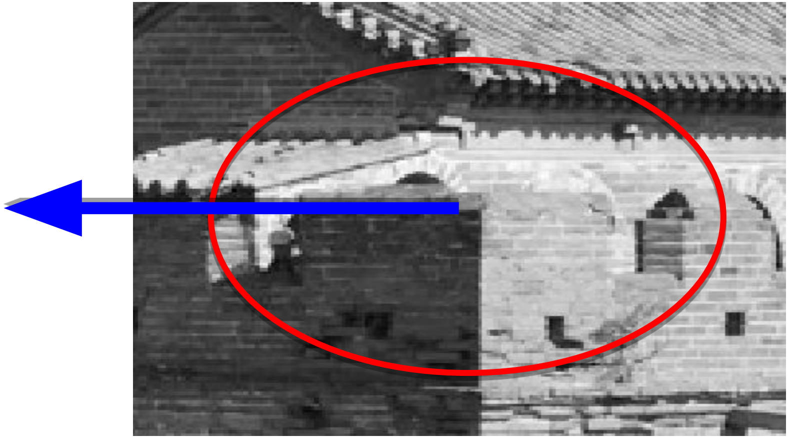
- Tienen una **posición** y **tamaño** bien definidos en la imagen.



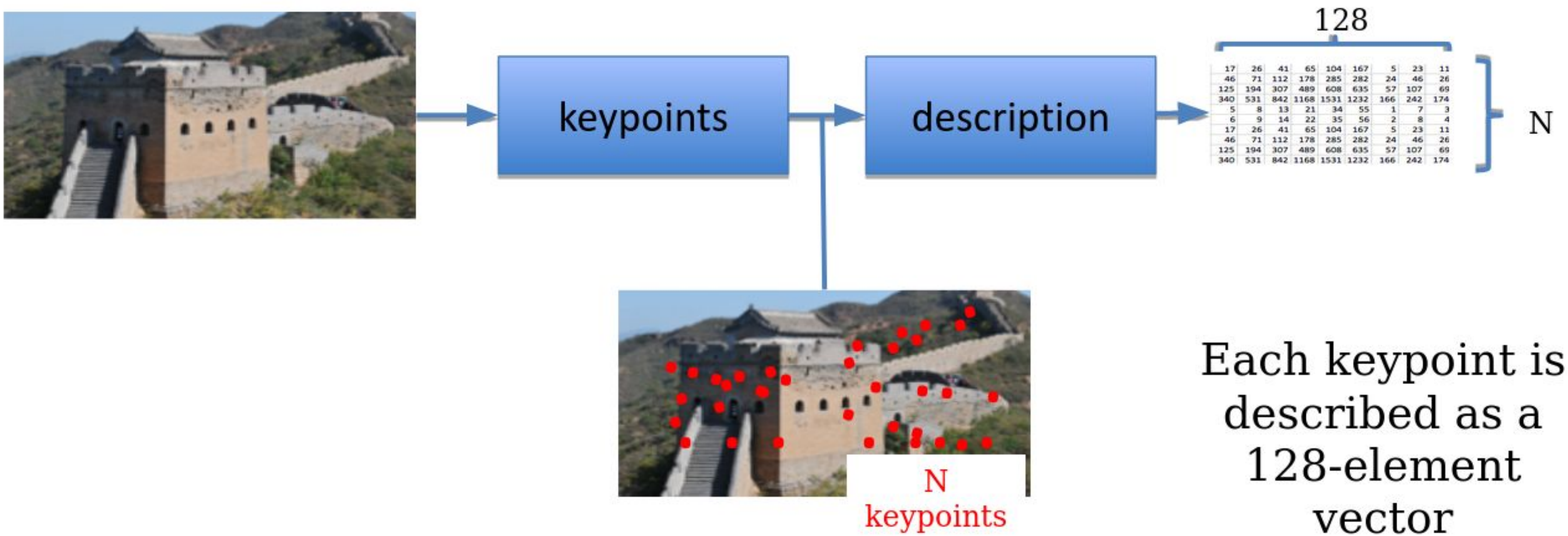
SIFT: Scale Invariant Feature Transform

Para que un keypoint sea de utilidad, debemos ser capaces de :

- Localizar su posición en la imagen.
- Determinar su tamaño.
- Determinar su orientación.
- Formular una descripción o “firma” que sea independiente de su tamaño y orientación.

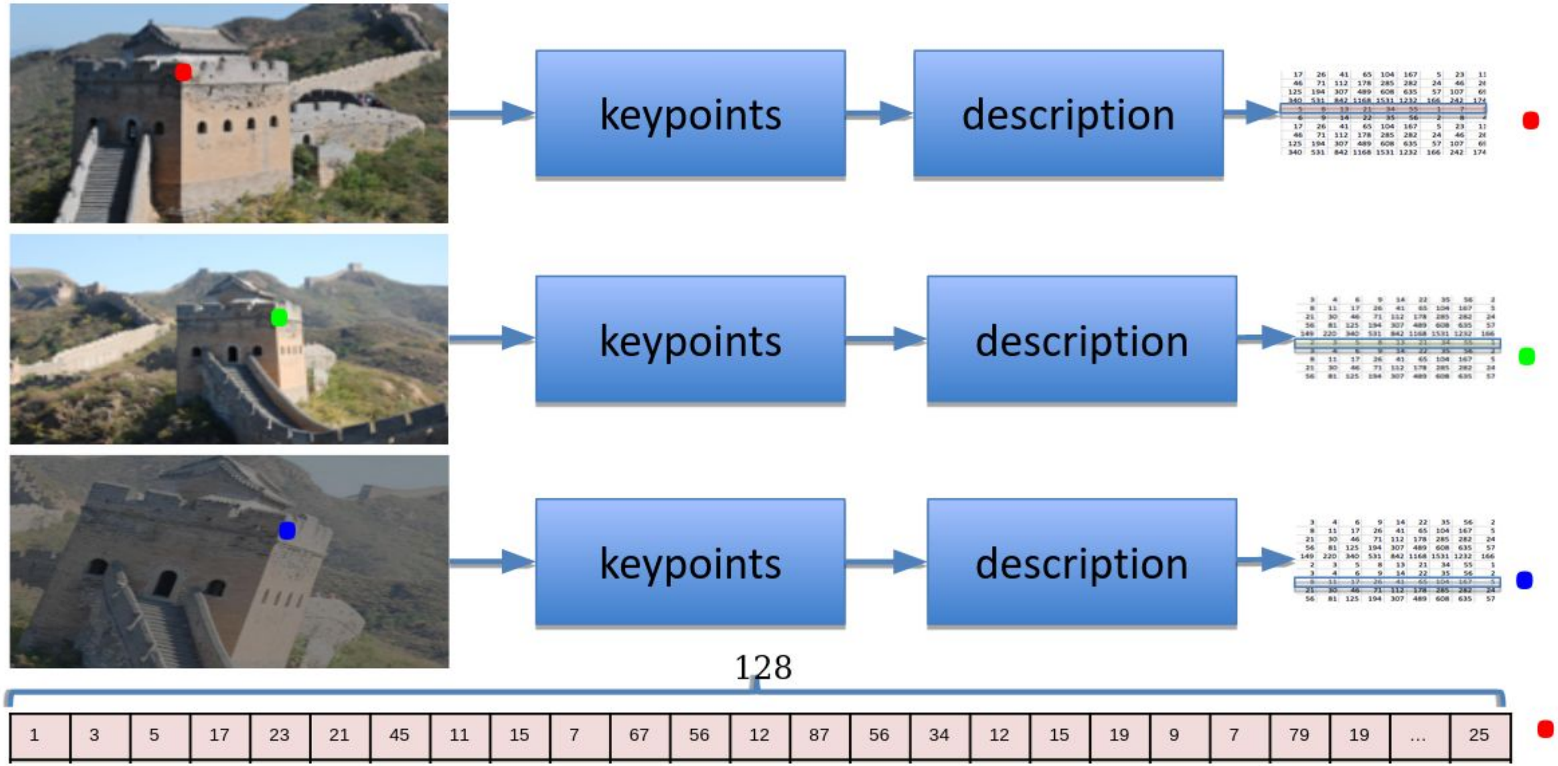


SIFT: Scale Invariant Feature Transform

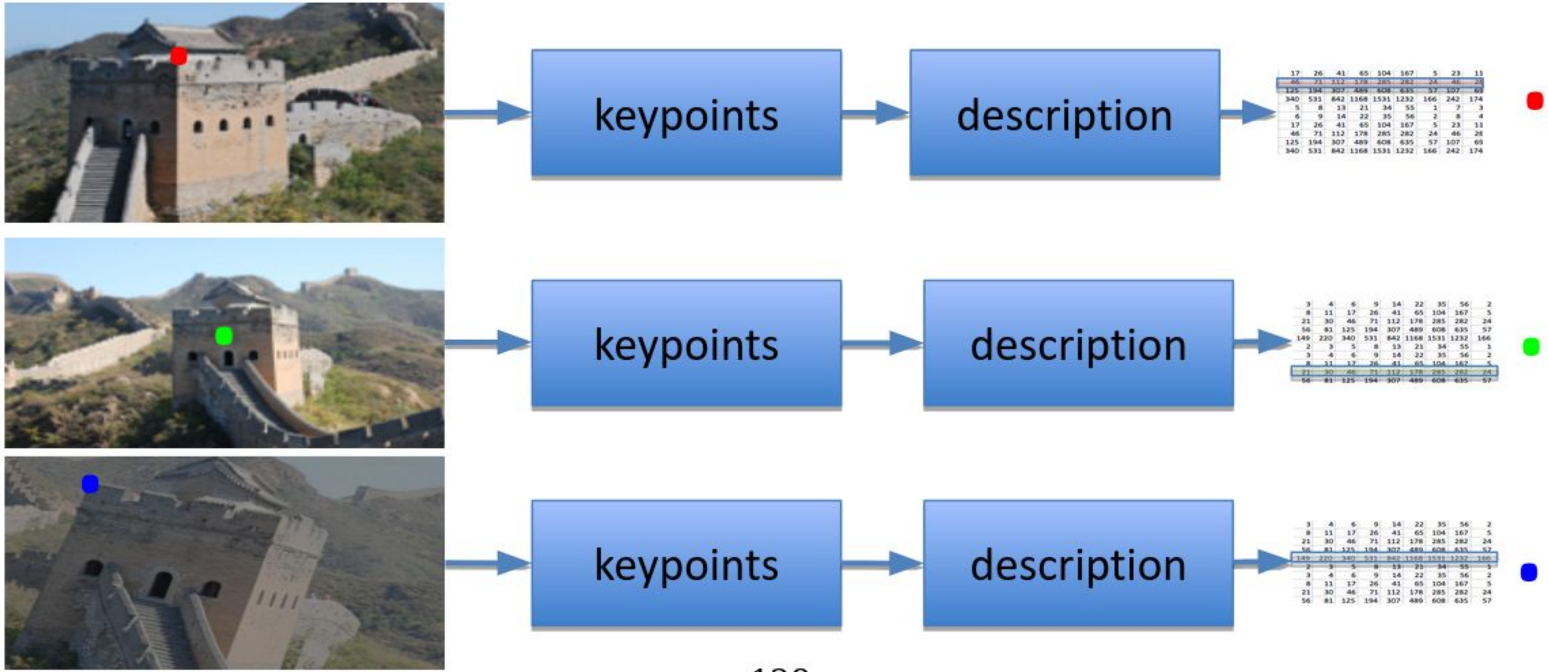


Each keypoint is described as a 128-element vector

SIFT: Scale Invariant Feature Transform



SIFT: Scale Invariant Feature Transform



128

1	3	5	17	23	21	45	11	15	7	67	56	12	87	56	34	12	15	19	9	7	79	19	…	25
23	65	32	90	76	56	34	98	6	8	56	8	9	23	8	45	2	43	67	19	62	78	55	…	7
99	76	34	12	98	76	90	55	43	87	65	43	32	65	7	9	1	3	45	66	39	18	39	…	78

SIFT: Scale Invariant Feature Transform

- It is used to detect local features using keypoints
- Each keypoint is described using a 128-element vector called ‘SIFT-descriptor’
- SIFT-descriptor is:
  - Scale invariant
  - Rotation invariant
  - Illumination invariant
  - Viewpoint invariant
- SIFT-descriptor is like a ‘signature’:
  - SIFT-descriptors of the same point (in different images) are very similar.
  - SIFT-descriptors of different points are very different.

SIFT: Scale Invariant Feature Transform

SIFT

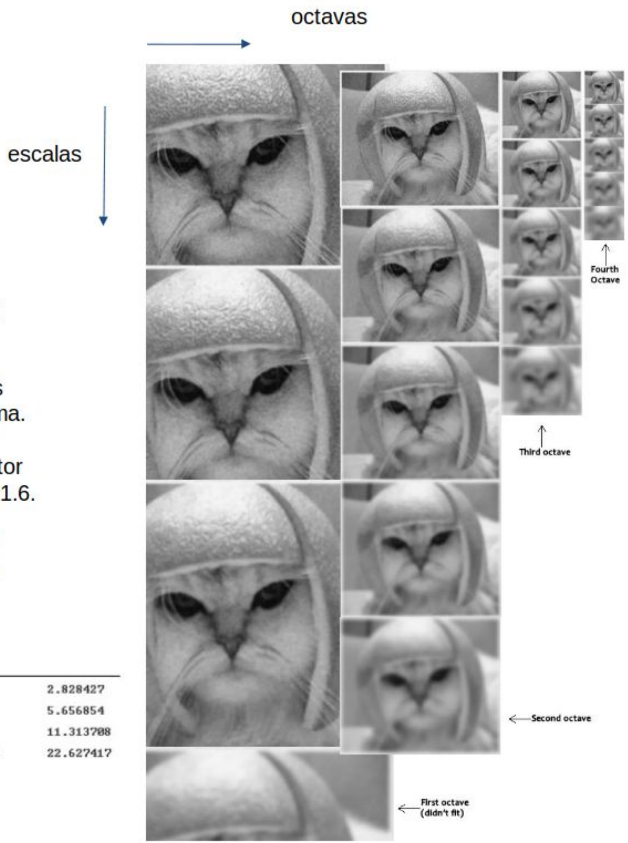
Scale-Space

El espacio de escala está formado por N octavas, cada una con M escalas.

Las escalas se forman con convoluciones Gaussianas, que dependen de cierto sigma. Cada sigma siguiente se genera multiplicando el sigma anterior por un factor k. Comúnmente el sigma base es igual a 1.6.

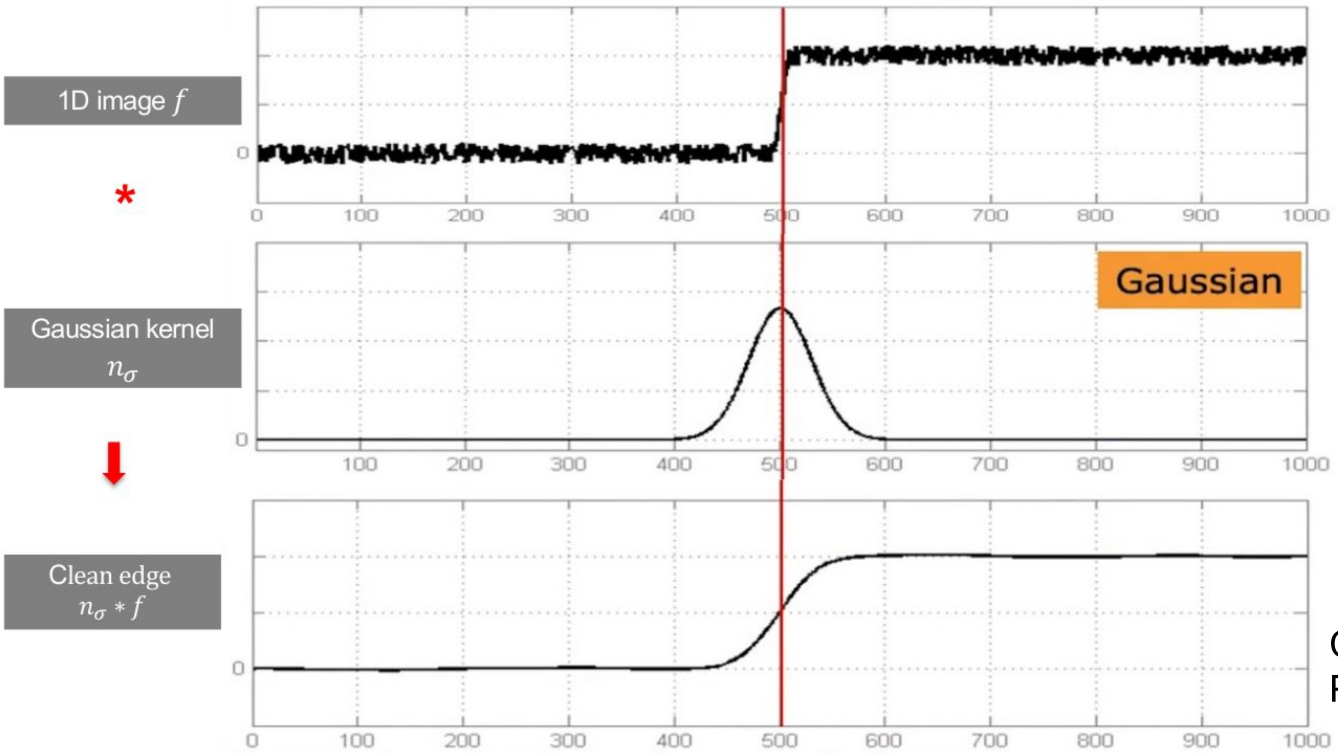
Una octava comienza cuando se llega al doble del sigma con respecto a la octava anterior.

octave \ scale →				
0.707107	1.000000	1.414214	2.000000	2.828427
1.414214	2.000000	2.828427	4.000000	5.656854
2.828427	4.000000	5.656854	8.000000	11.313708
5.656854	8.000000	11.313708	16.000000	22.627417



SIFT – Detection of keypoints

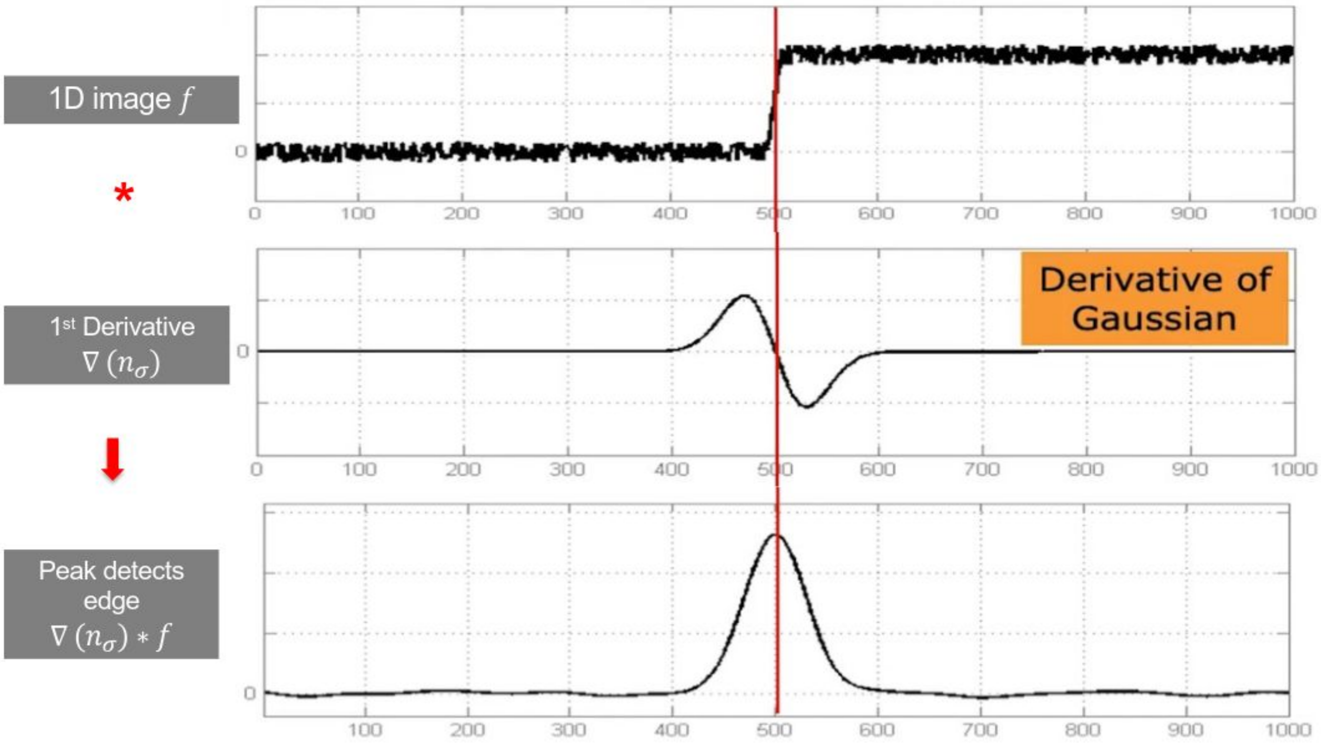
Recordatorio:



Créditos:
Peter YK Cheung

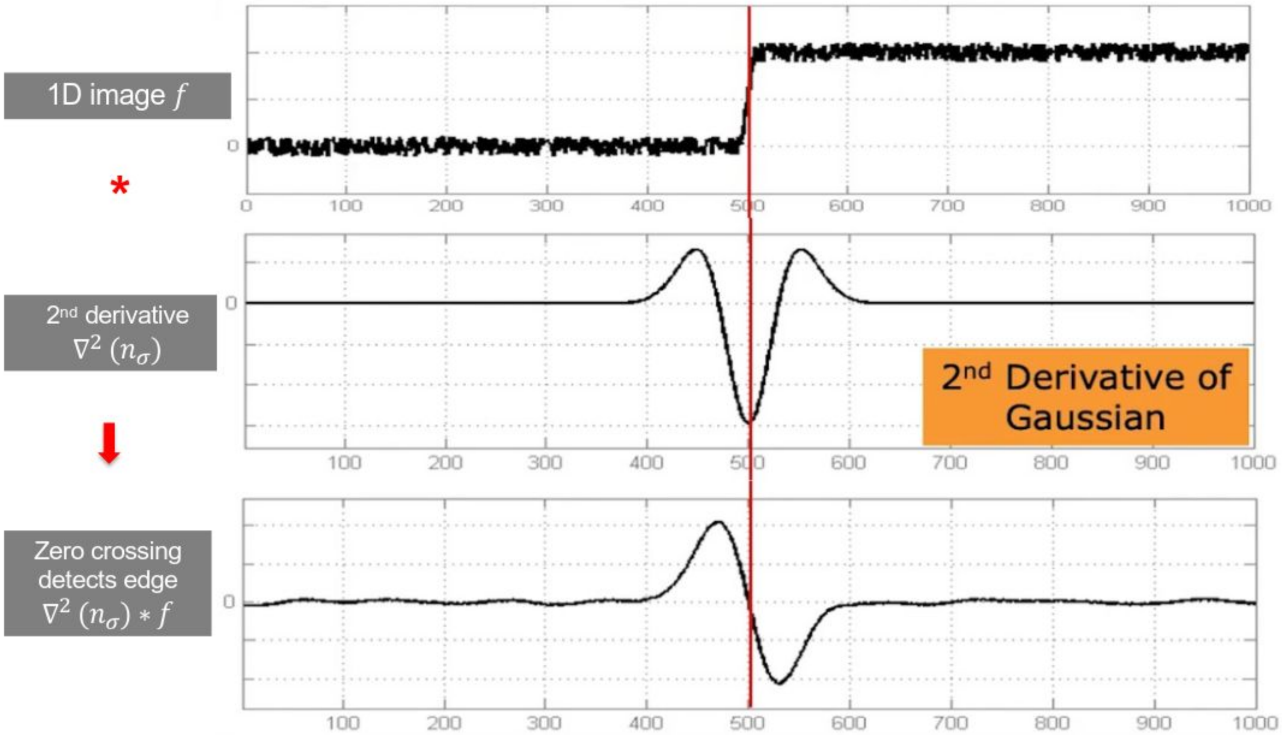
SIFT – Detection of keypoints

Recordatorio:



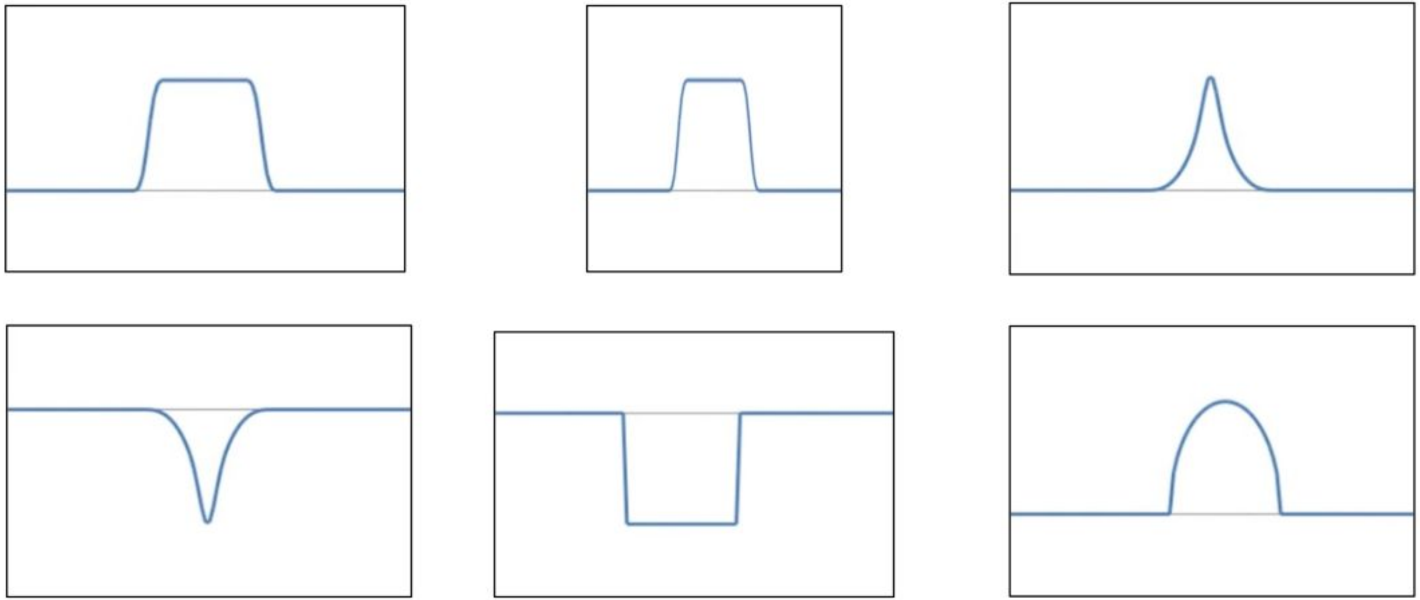
SIFT – Detection of keypoints

Recordatorio:

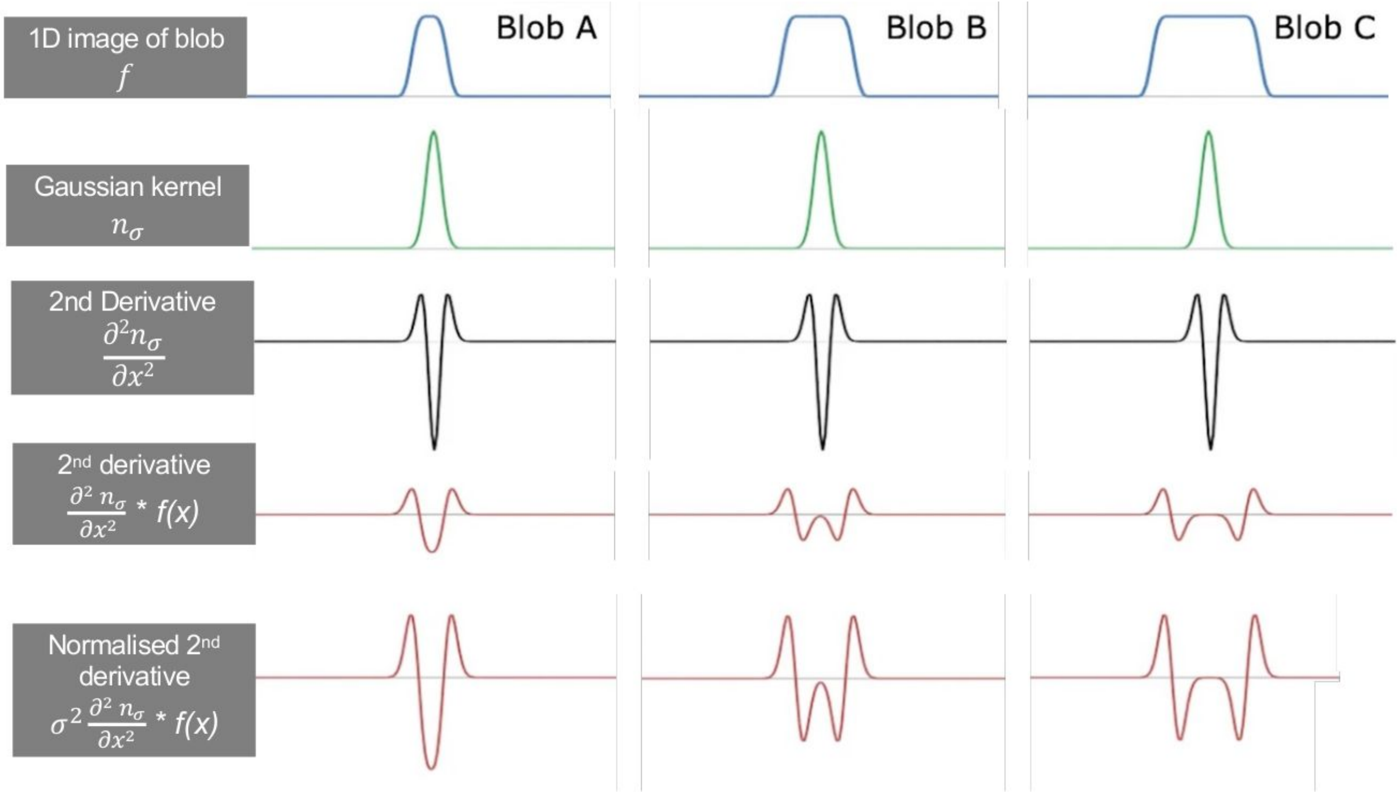


SIFT – Detection of keypoints

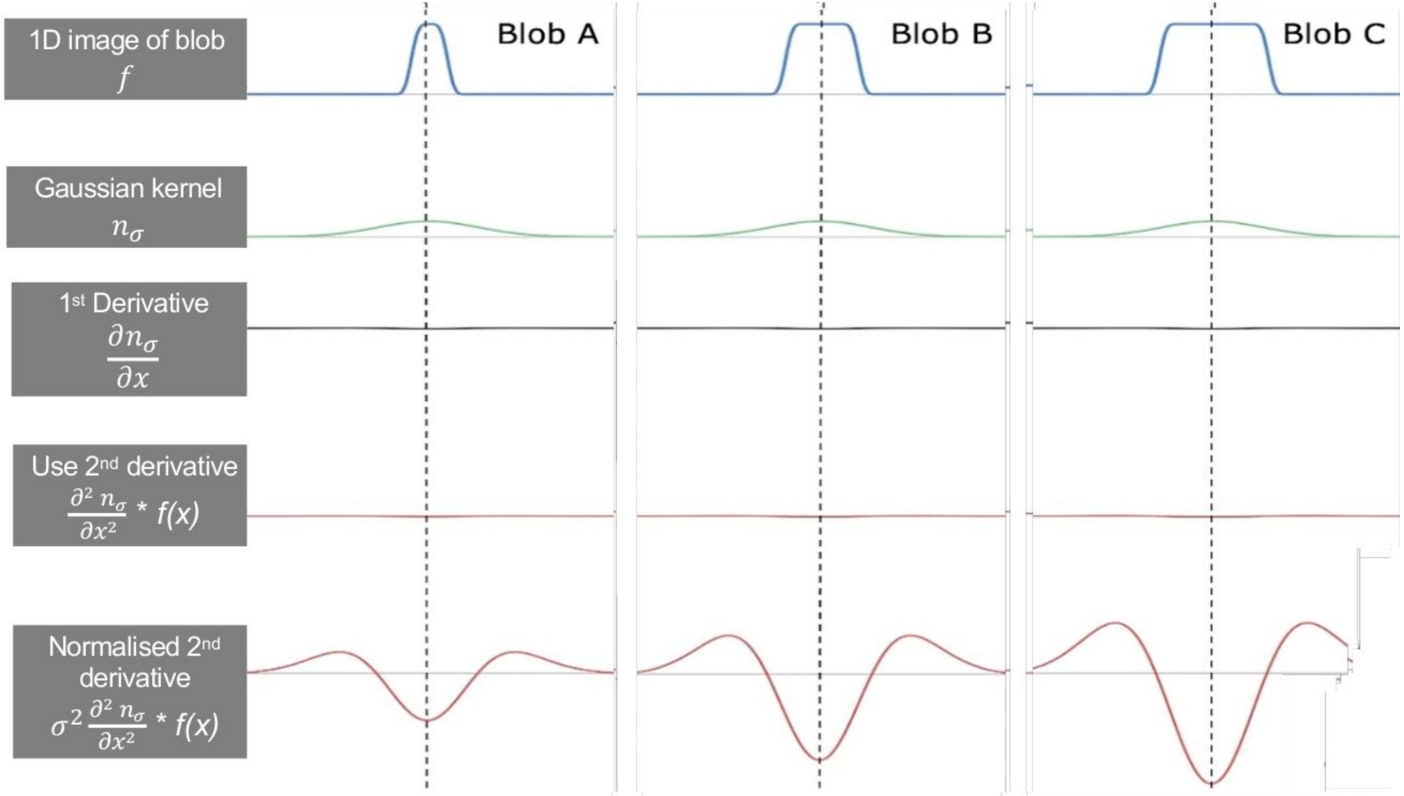
- Queremos ser capaces de detectar blobs en una imagen.
- Estos pueden tener distintas formas y tamaños.
- Consideremos los siguientes ejemplos en 1D:



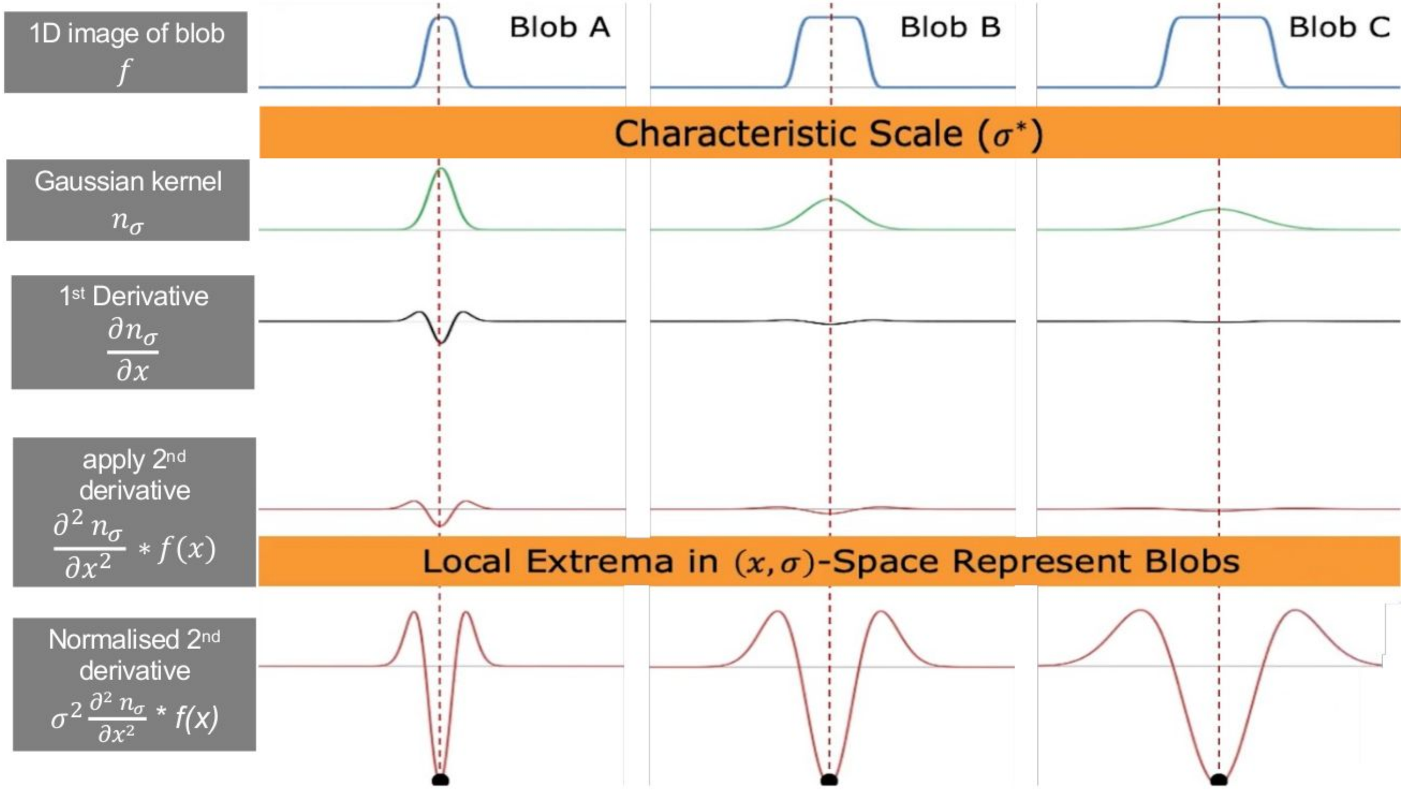
SIFT – Detection of keypoints



SIFT – Detection of keypoints



SIFT – Detection of keypoints



SIFT – Detection of keypoints

La detección de blobs en 1D seguirá los siguientes pasos:

1. Given a 1D signal $f(x)$, convolve it with σ-normalized 2nd derivative function:

   Compute: $\sigma^2 \frac{\partial^2 n_\sigma}{\partial x^2} * f(x)$ at different scales $(\sigma_0, \sigma_1, \ldots, \sigma_k)$.

2. Find $(x^*, \sigma^*) = \max_{(x,\sigma)} \left| \sigma^2 \frac{\partial^2 n_\sigma}{\partial x^2} * f(x) \right|$

3. Blob position = x^*

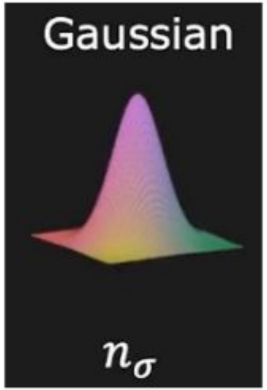
4. Blob size = σ^*

SIFT – Detection of keypoints

For 2D image $I(x, y)$, use **Normalized Laplacian of Gaussian** (NLoG) for blob detection:

Laplacian

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$



SIFT – Detection of keypoints

1. Given an image $I(x, y)$, convolve it with $NLoG$ at many scales of σ .

   Compute: $(\sigma^2\,\nabla^2 n_\sigma) * I(x,y)$ at different scale $(\sigma_0, \sigma_1, \dots, \sigma_k)$.

2. Find $(x^*, y^*, \sigma^*) = \max\limits_{(x,y,\sigma)} \left|(\sigma^2\,\nabla^2 n_\sigma) * I(x,y)\right|$

3. Blob position = (x^*, y^*)

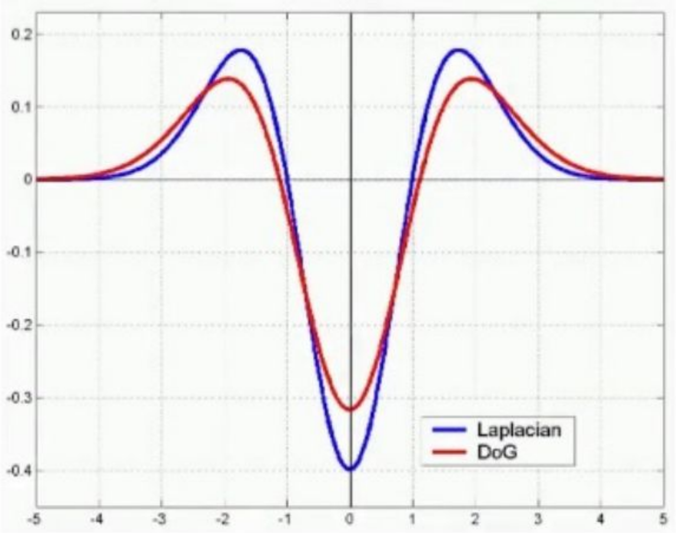
4. Blob size = σ^*

SIFT – Detection of keypoints

- Is there a faster way to compute NLoG?
- Difference of Gaussian (DoG):

$$DoG = (n_{s\sigma} - n_{\sigma}) \approx (s-1)\underbrace{\sigma^2\nabla^2 n\_\sigma}_{\text{NLoG}}$$

- s is different multipliers (octave) of σ.



$$DoG \approx (s-1)NLoG \qquad s > 1$$

SIFT – Detection of keypoints

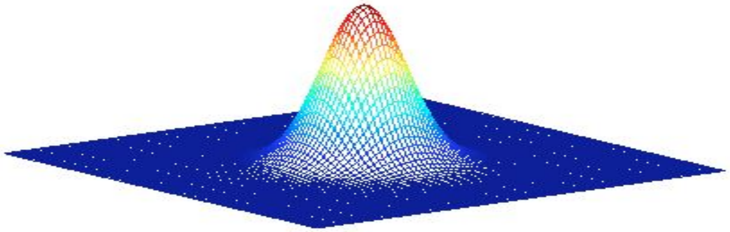
A synthetic image with a spot

Two goals:

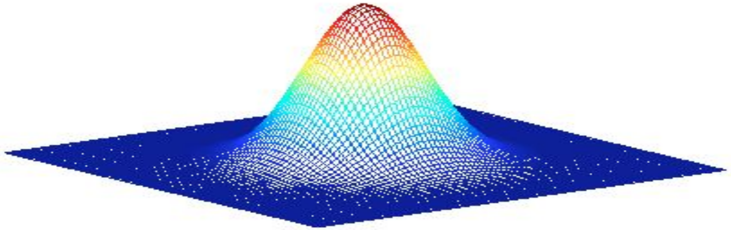
- Where?
- Size?

SIFT – Detection of keypoints

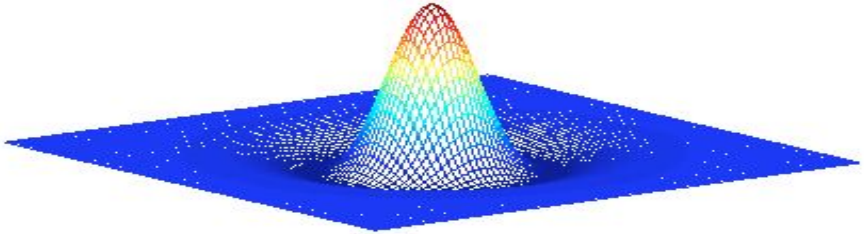
For the detection a DoG – Mask is used:



G1 = Gaussian 1 (with σ)

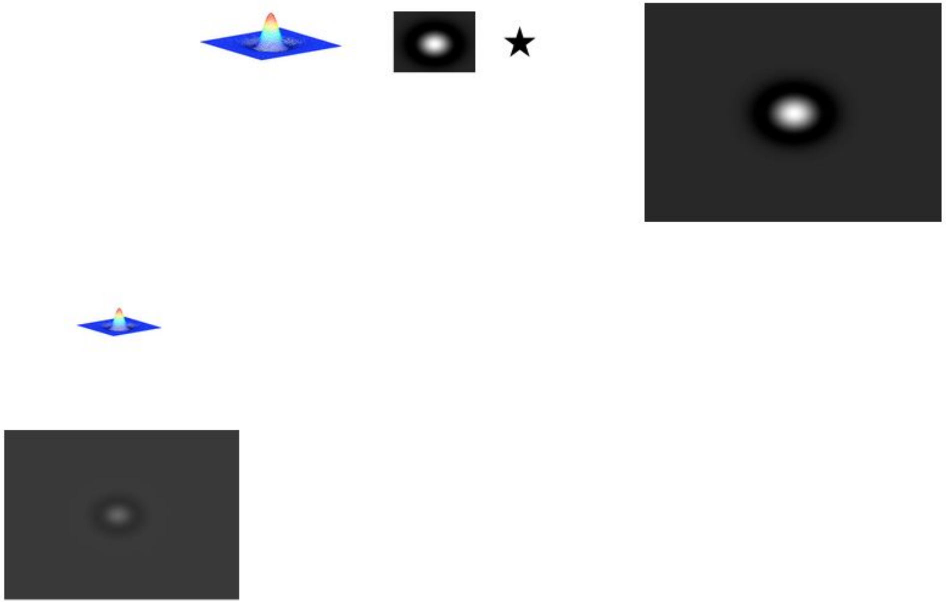


G2 = Gaussian 2 (with *k*σ)



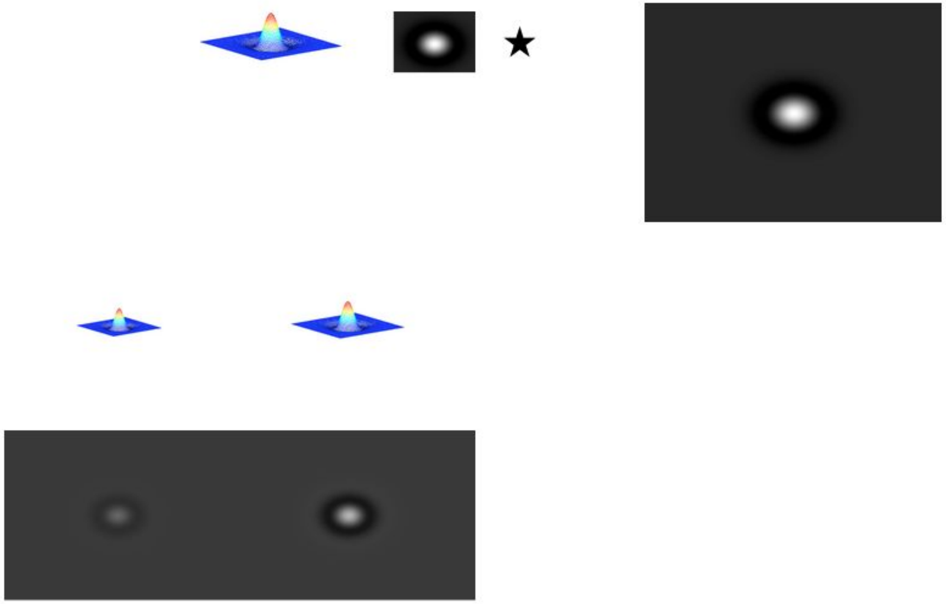
Difference of Gaussians (DoG): G2-G1

SIFT – Detection of keypoints



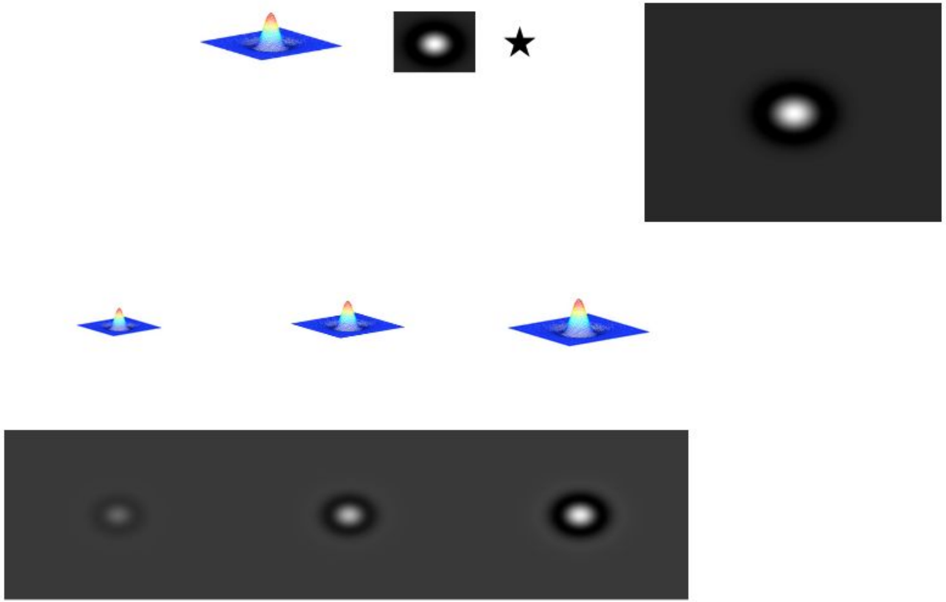
★ : Convolution

SIFT – Detection of keypoints



⋆ : Convolution

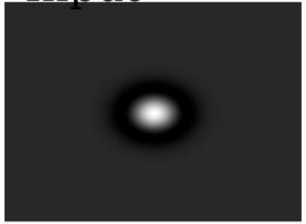
SIFT – Detection of keypoints



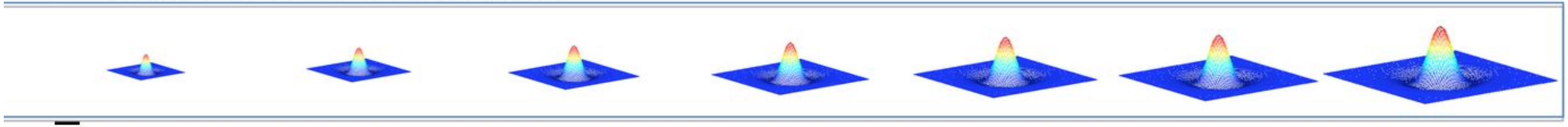
★ : Convolution

SIFT – Detection of keypoints

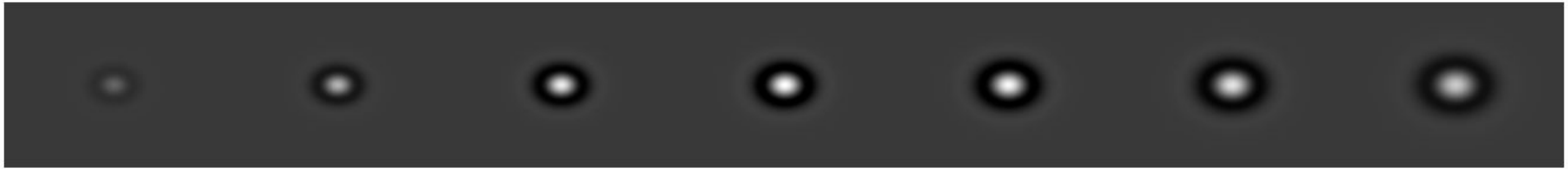
Input



- with DoG of different σ

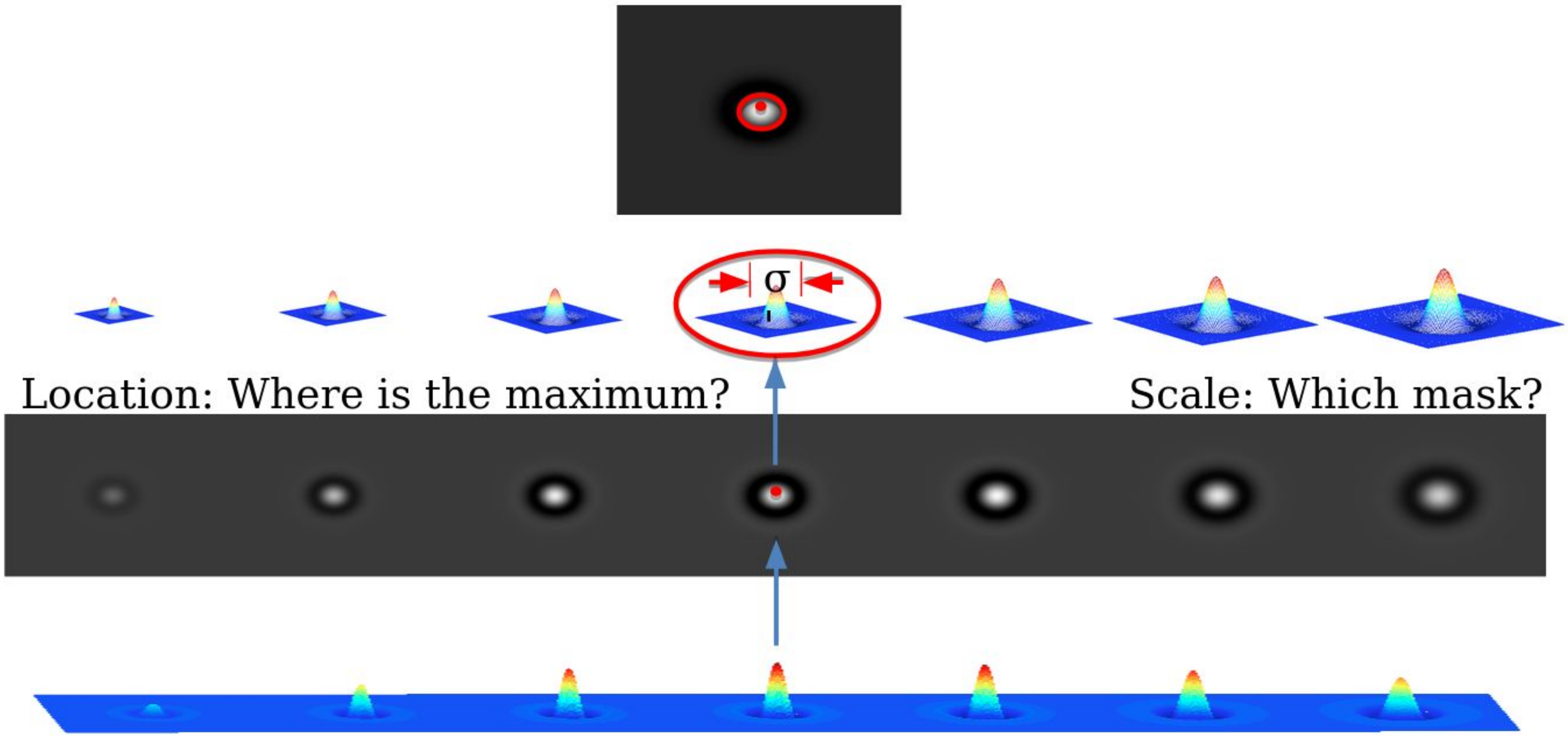


=



Outputs

SIFT – Detection of keypoints



SIFT – Detection of keypoints

$S(x, y, \sigma_0)$ $S(x, y, \sigma_1)$ $S(x, y, \sigma_2)$ $S(x, y, \sigma_3)$

Increasing σ, Higher Scale, Lower Resolution

Scale Space: Stack created by filtering an image with Gaussians of different sigma (σ)

$$\boxed{S(x, y, \sigma) = n(x, y, \sigma) * I(x, y)}$$

SIFT – Detection of keypoints

$S(x, y, \sigma_0)$ $S(x, y, \sigma_1)$ $S(x, y, \sigma_2)$ $S(x, y, \sigma_3)$ …

Increasing σ, Higher Scale, Lower Resolution

Selecting sigmas to generate the scale-space:

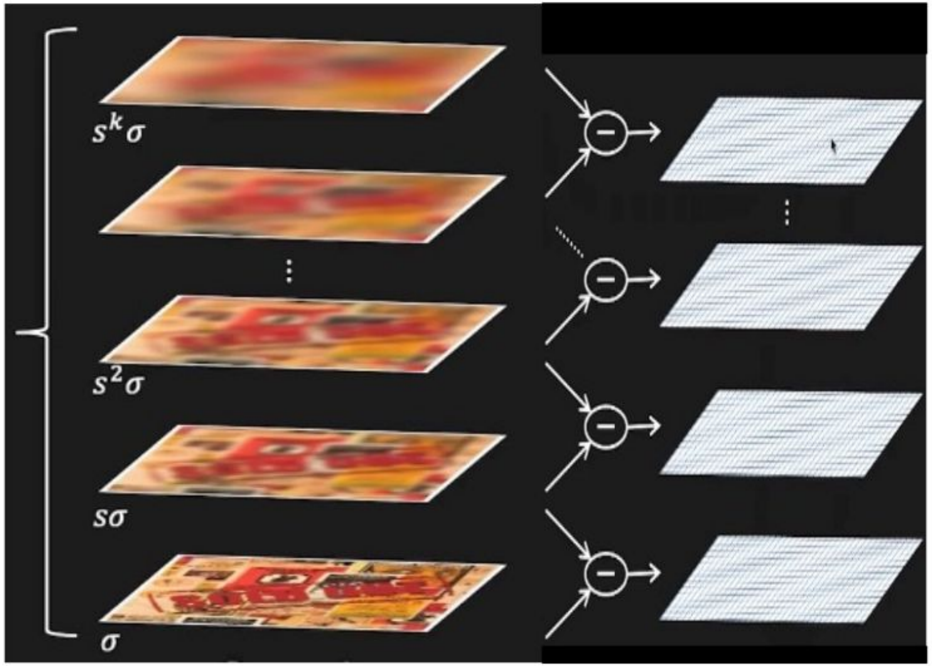
$$\boxed{\sigma_k = \sigma_0 s^k} \quad k = 0,1,2,3, \ldots$$

s: Constant multiplier

σ_0: Initial Scale

SIFT – Detection of keypoints

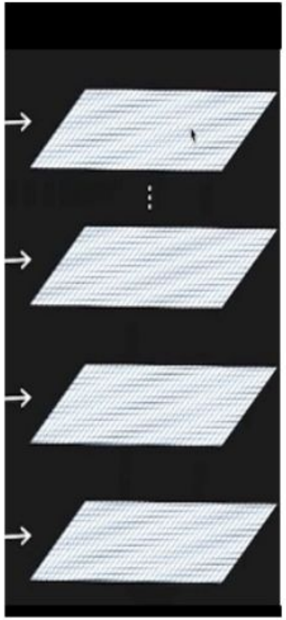
Image
$I(x,y)$



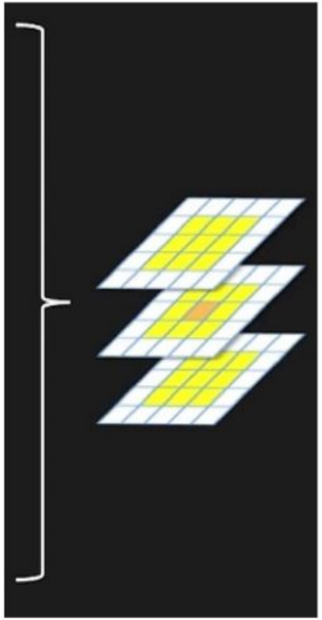
Gaussian Scale Space
$S(x,y,\sigma)$

Difference of Gaussians (DoG)
$\approx (s-1)\sigma^2\nabla^2 S(x,y,\sigma)$

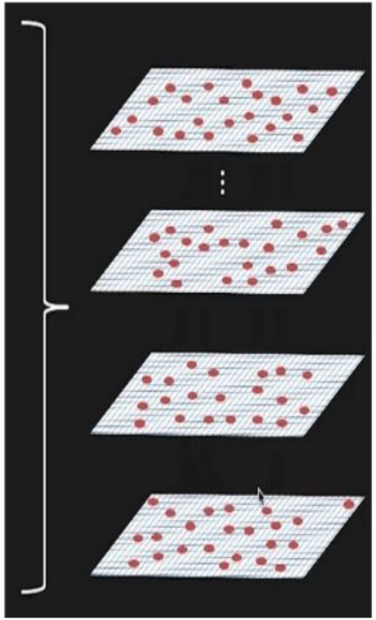
SIFT – Detection of keypoints



Difference of Gaussians (DoG)
$\approx (s-1)\sigma^2\nabla^2 S(x,y,\sigma)$

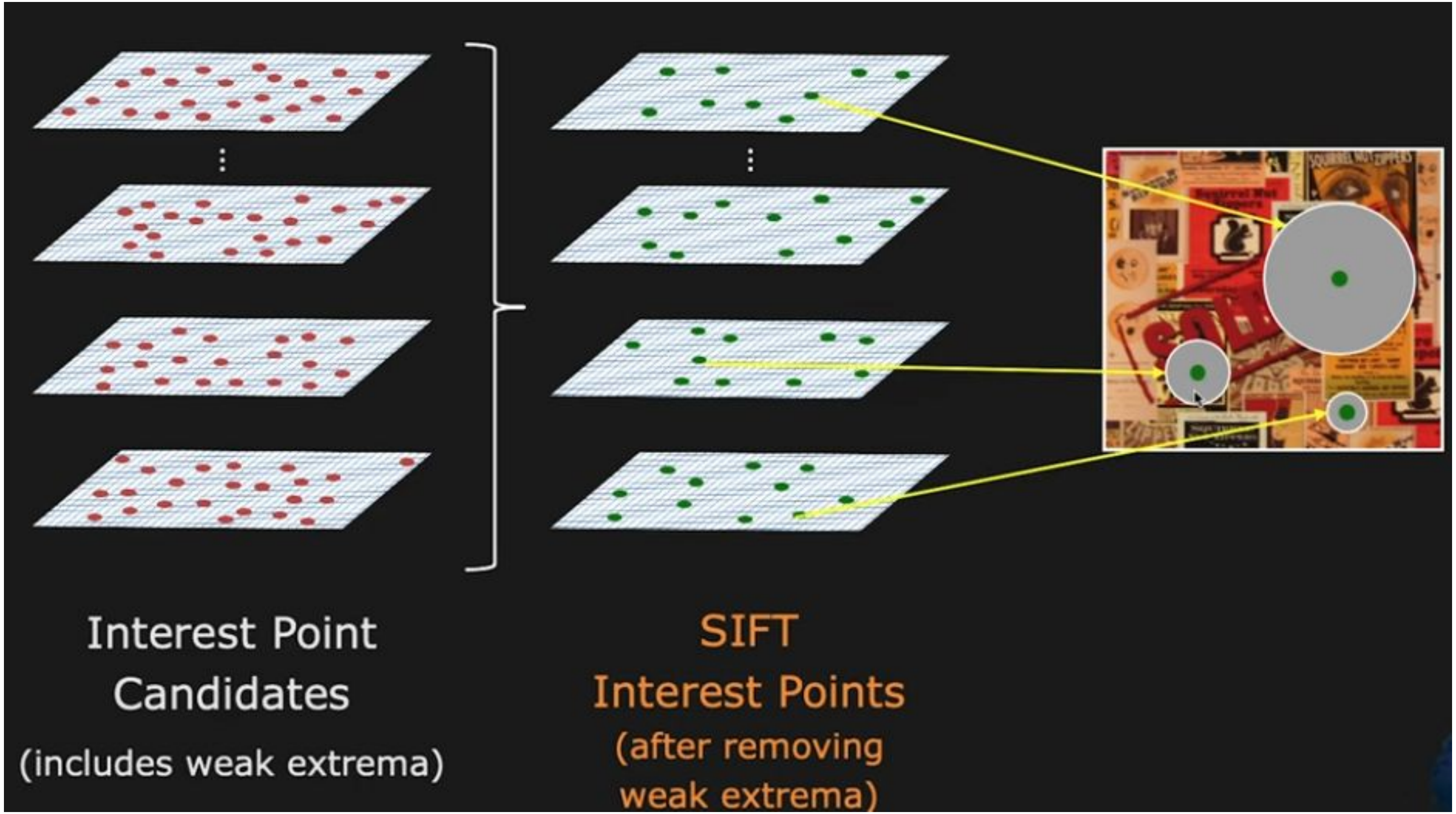


Find peaks (extrema) in every 3x3 grid

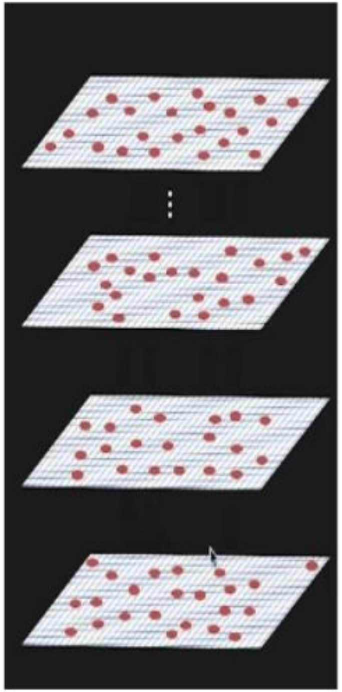


Candidates for Interest Point

SIFT – Detection of keypoints



SIFT – Detection of keypoints



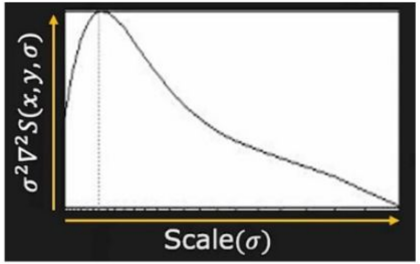
Candidates for Interest Point

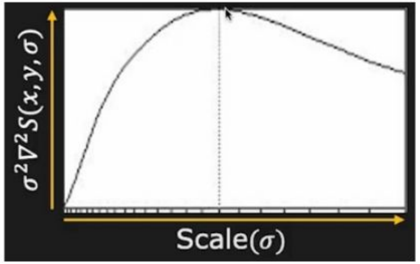
SIFT Interest Points after removing weak peaks

Annotated SIFT Features

SIFT – Detection of keypoints

Invarianza de escala:





Ratio of Blob Sizes

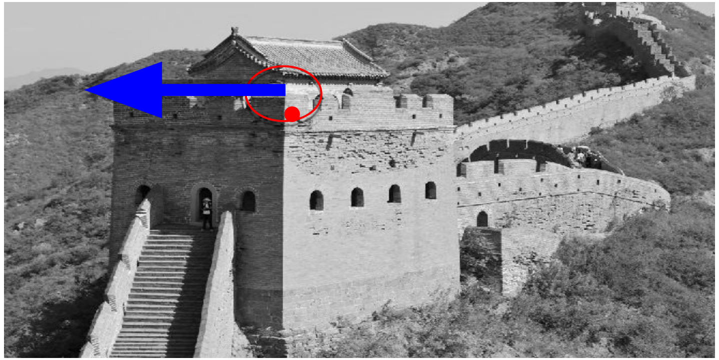
$$= \frac{\sigma_1^*}{\sigma_2^*}$$

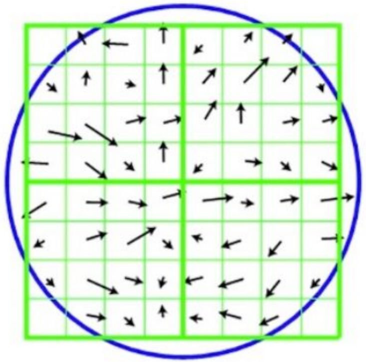
SIFT – Descriptor of keypoint

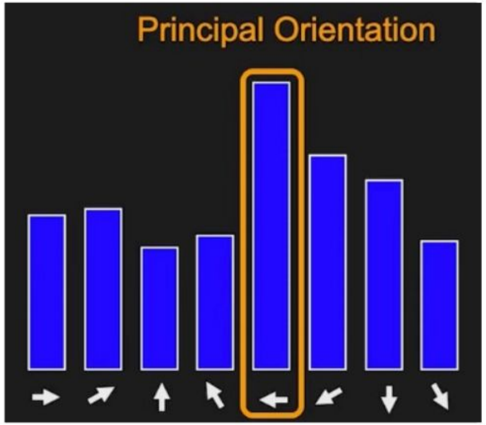
1. Find a keypoint (x,y,σ')

SIFT – Descriptor of keypoint

1. Find a keypoint (x,y,σ’)
2. Find the orientation using A(x,y) matrix







SIFT – Descriptor of keypoint

El paso anterior permite que el método sea invariante a rotaciones:

Imagen A

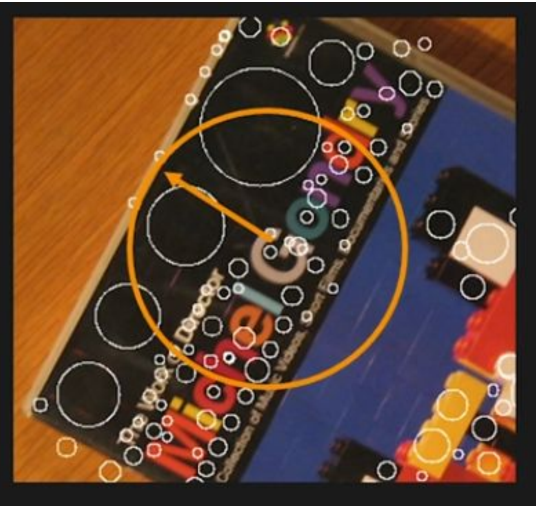


Imagen B

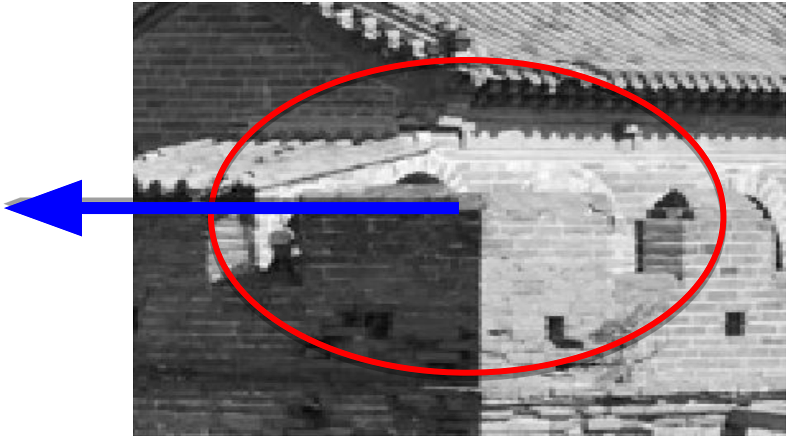
Imagen B luego de compensar según la dirección principal

SIFT – Descriptor of keypoint

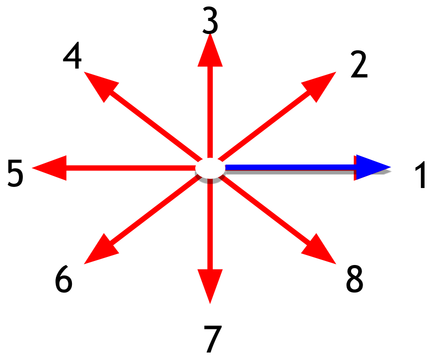
1. Find a keypoint (x,y,σ')
2. Find the orientation using $A(x,y)$ matrix
3. Take a window centered in the keypoint of size 1.5 σ'

SIFT – Descriptor of keypoint

1. Find a keypoint (x,y,σ’)
2. Find the orientation using A(x,y) matrix
3. Take a window centered in the keypoint of size 1.5 σ’
4. Align the window to direction ‘1’.

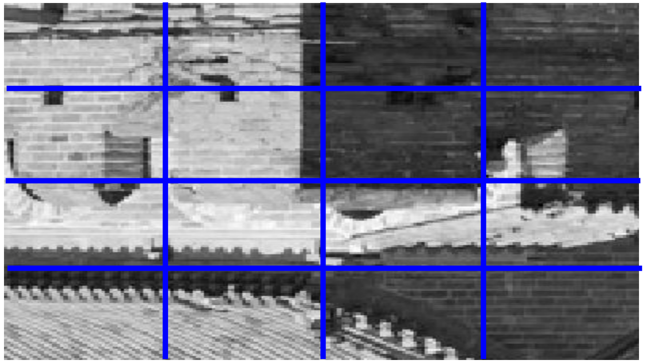


8 directions

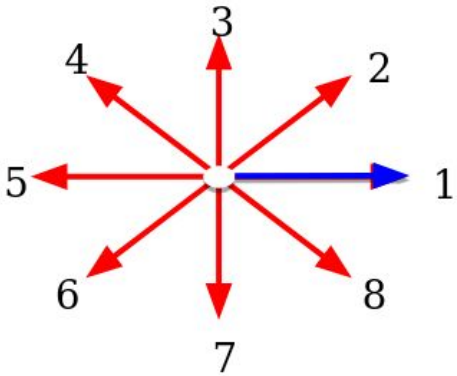


SIFT – Descriptor of keypoint

1. Find a keypoint (x,y,σ’)
2. Find the orientation using A(x,y) matrix
3. Take a window centered in the keypoint of size 1.5 σ’
4. Align the window to direction ‘1’
5. Define 16 partitions

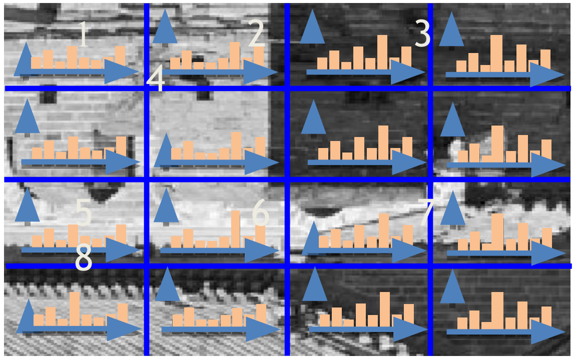


8 directions

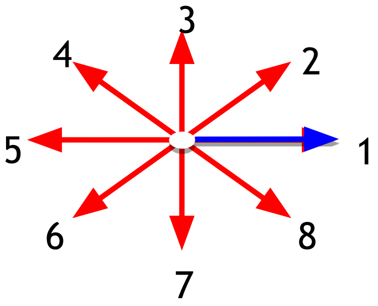


SIFT – Descriptor of keypoint

1. Find a keypoint (x,y,σ’)
2. Find the orientation using A(x,y) matrix
3. Take a window centered in the keypoint of size 1.5 σ’
4. Align the window to direction ‘1’
5. Define 16 partitions
6. Compute 8 bin histograms in each partition

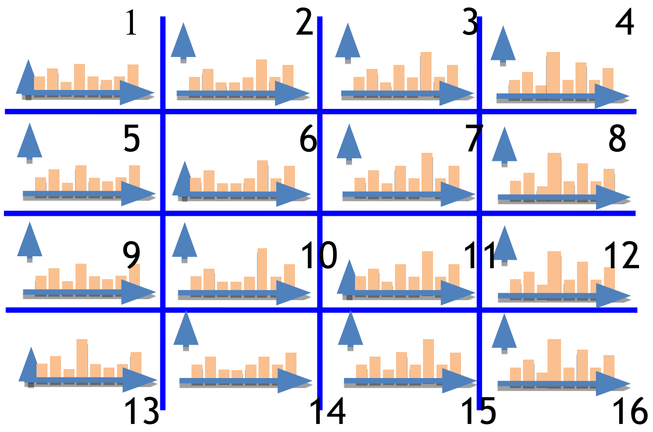


8 directions



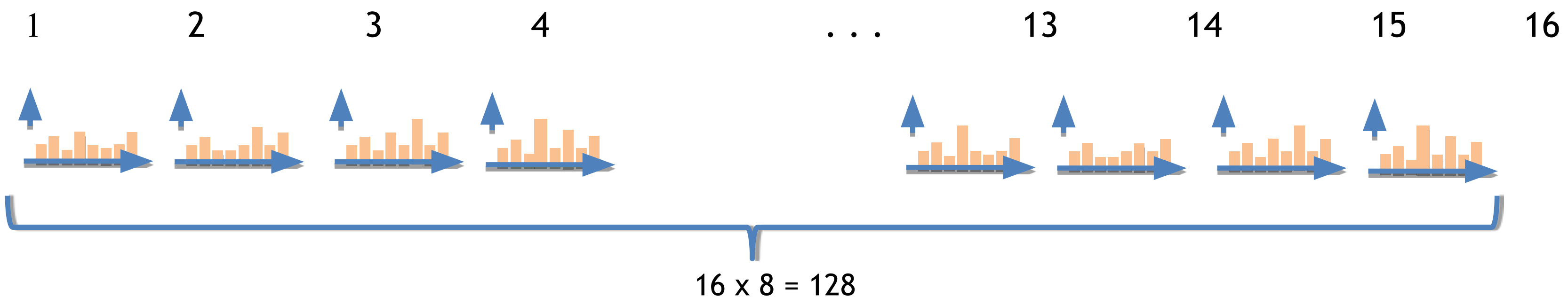
SIFT – Descriptor of keypoint

1. Find a keypoint (x,y,σ')
2. Find the orientation using A(x,y) matrix
3. Take a window centered in the keypoint of size 1.5 σ'
4. Align the window to direction '1'
5. Define 16 partitions
6. Compute 8 bin histograms in each partition



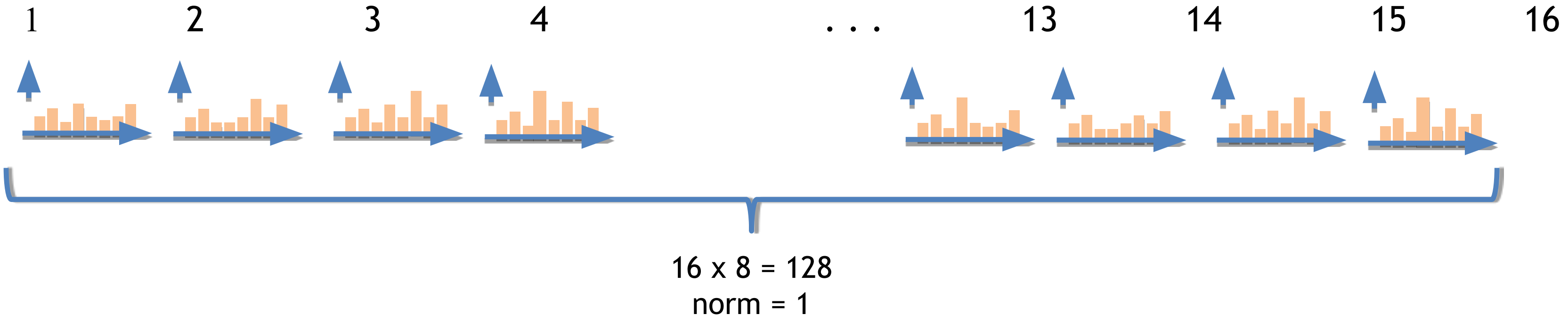
SIFT – Descriptor of keypoint

7. Concatenate all histograms



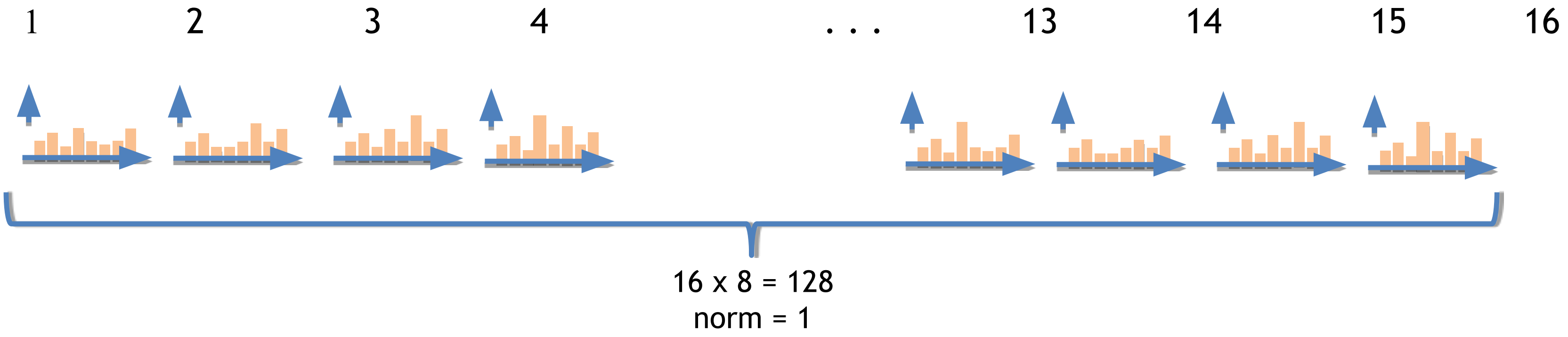
SIFT – Descriptor of keypoint

7. Concatenate all histograms
8. Normalize

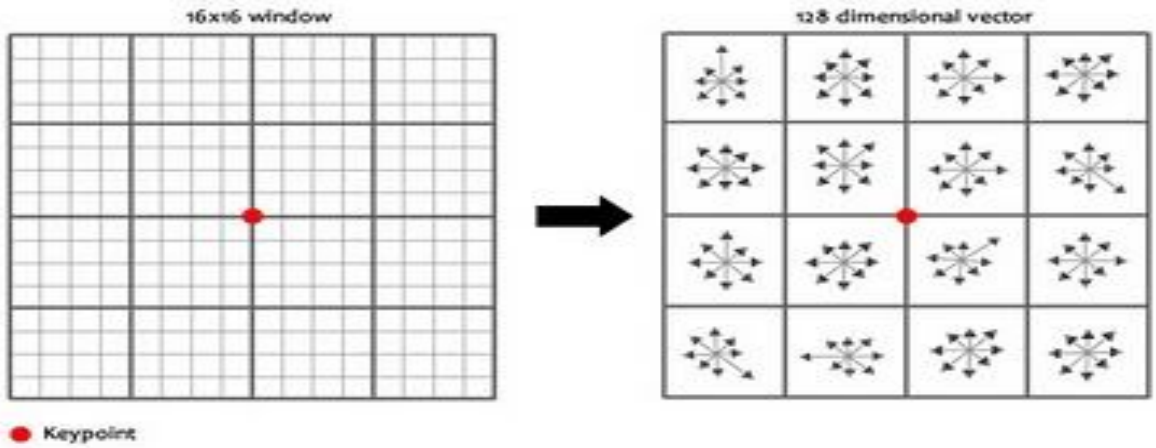


SIFT – Descriptor of keypoint

7. Concatenate all histograms
8. Normalize
9. **The end**



SIFT: Scale Invariant Feature Transform



Each keypoint is described as a
128-element vector

A. Saharkhiz: Details behind SIFT
http://areshmatlab.blogspot.com/2010/07/details-behind-sift-feature-detection.html

SIFT: Scale Invariant Feature Transform

Finalmente, para encontrar correspondencias entre un keypoint k1 de una imagen A (descrito por un vector H1)y un keypoint k2 de una imagen B (descrito por un vector H2), se puede utilizar las siguientes métricas:

1. L2 Distance:

$$d(H_1, H_2) = \sqrt{\sum_k \left(H_1(k) - H_2(k)\right)^2}.$$ (Smaller d = better match.)

2. Normalized Correlation:

$$d(H_1, H_2) = \frac{\sum_k [(H_1(k) - \overline{H_1})(H_2(k) - \overline{H_2})]}{\sqrt{\sum_k (H_1(k) - H_1)^2} \sqrt{\sum_k (H_2(k) - H_2)^2}},$$

where $\overline{H_i} = \frac{1}{N} \sum_{k-1}^{N} H_i(k)$. (Larger d = better match.)

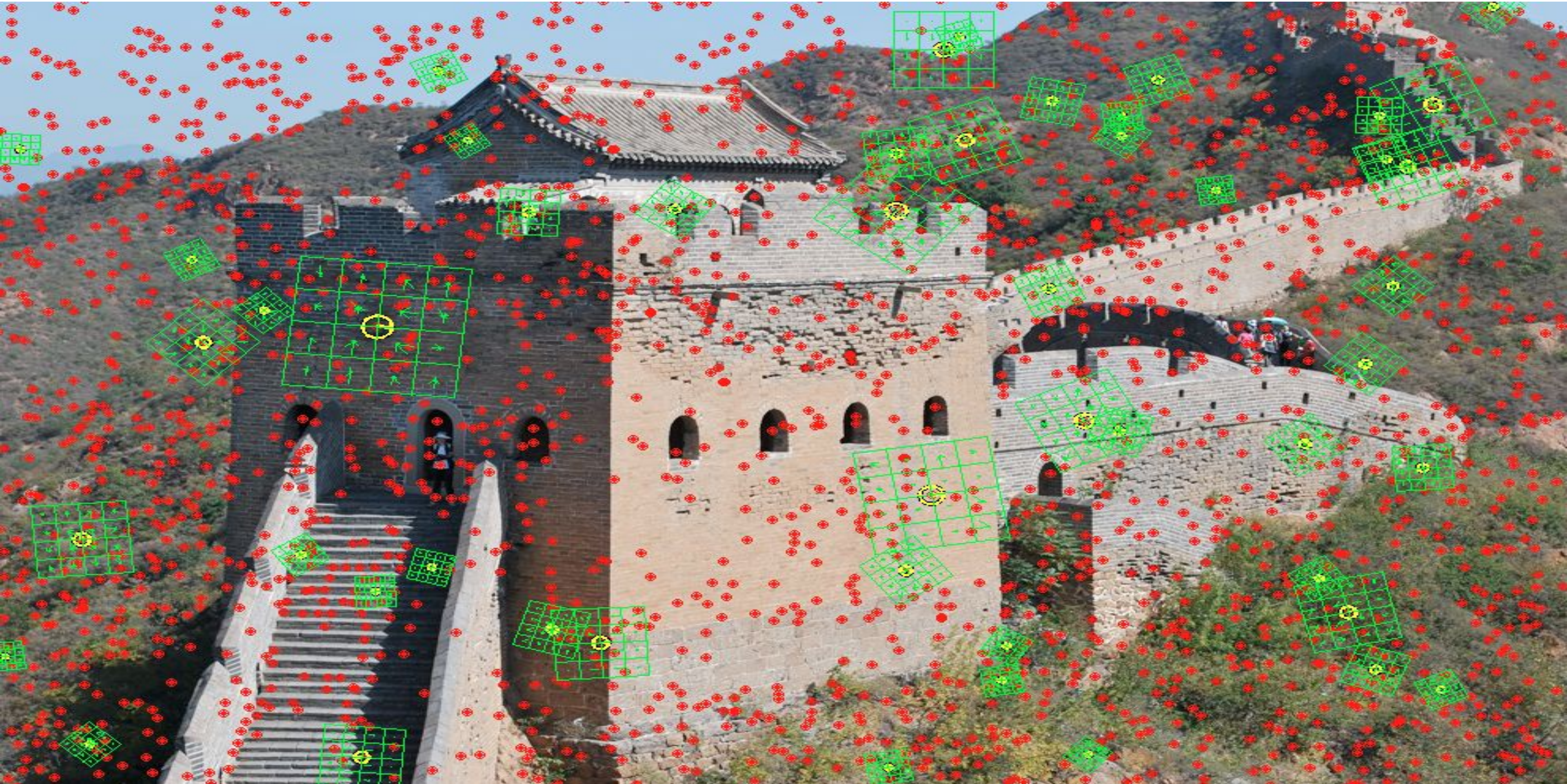
3. **Intersection:**

$$d(H_1, H_2) = \sum_k \min(H_1(k), H_2(k))$$

(Larger d = better match.)

SIFT keypoints

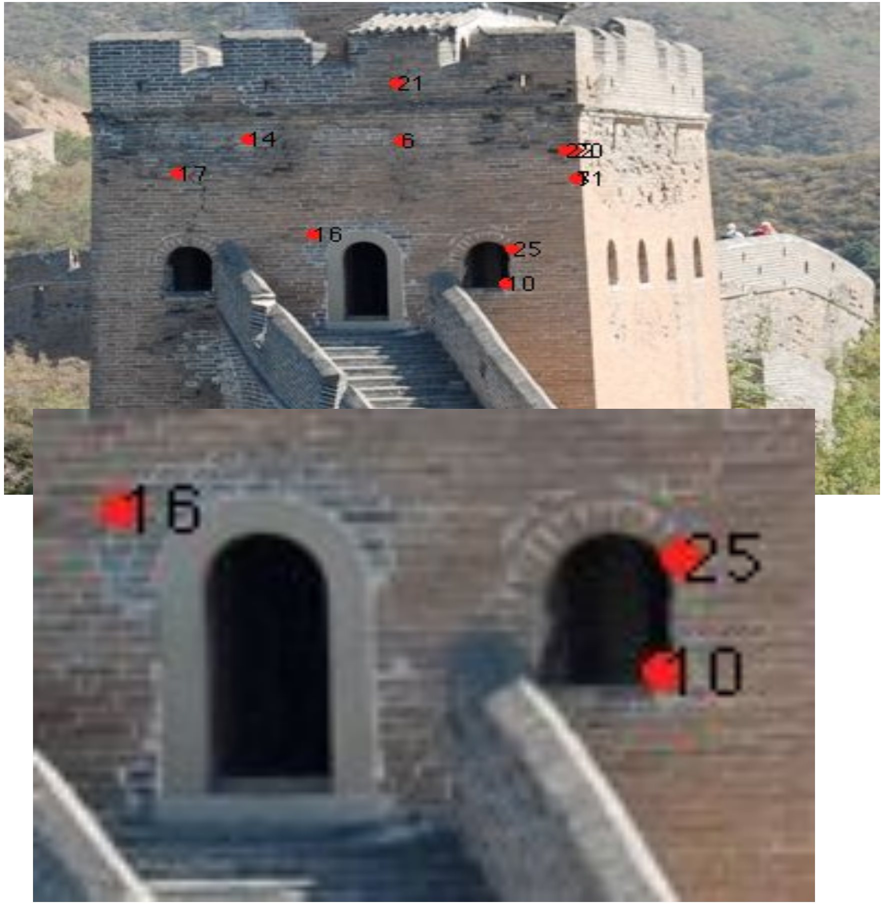
SIFT keypoints and descriptors



SIFT matching points in image 1

SIFT matching points in image 2

SIFT matching points in images 1 and 2



© Lowe

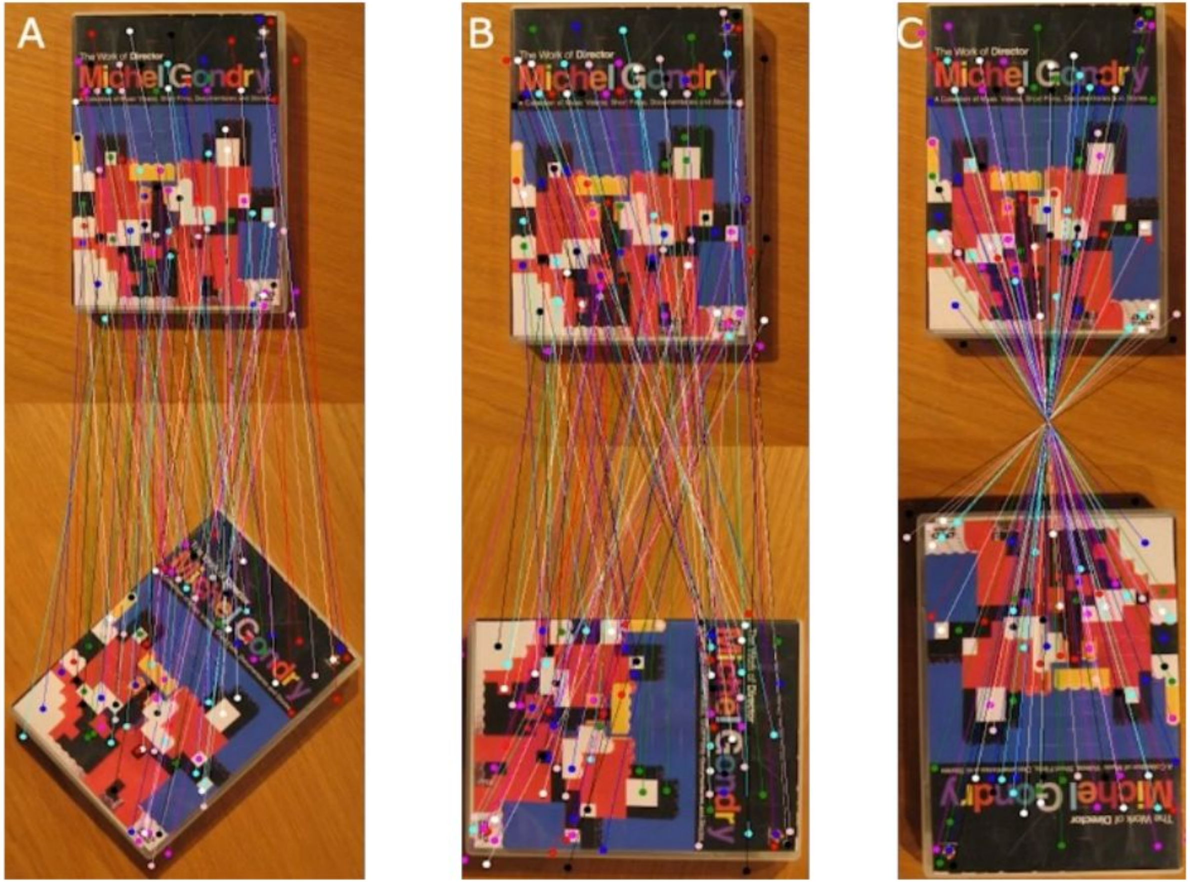
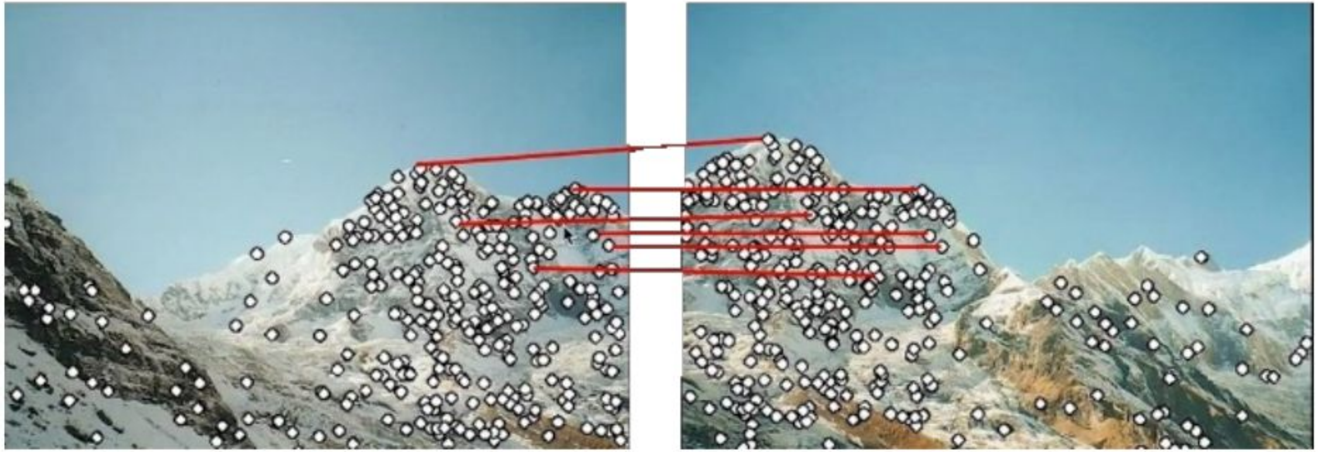
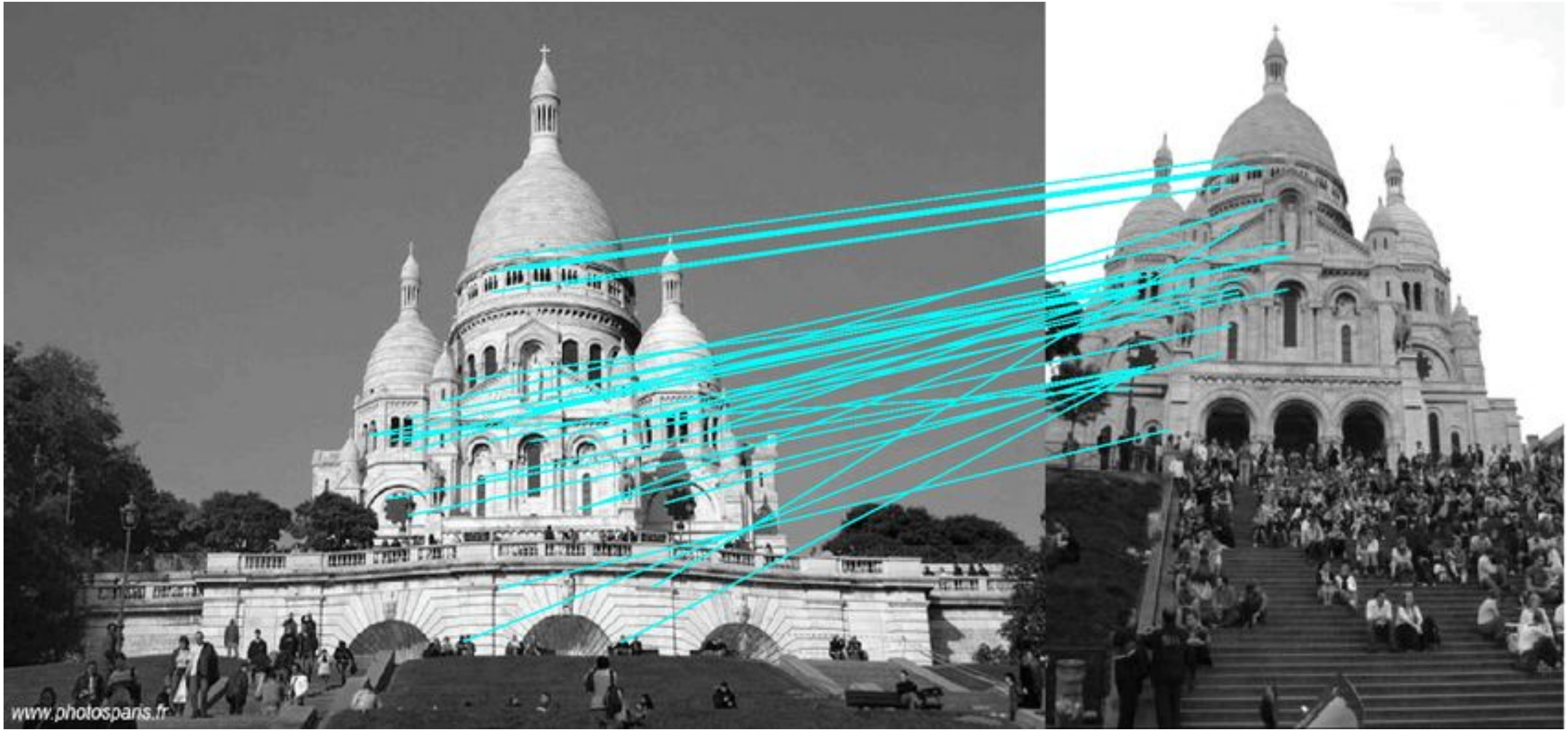

A
B
C
The Work of Director
Michel Gondry

Image 1

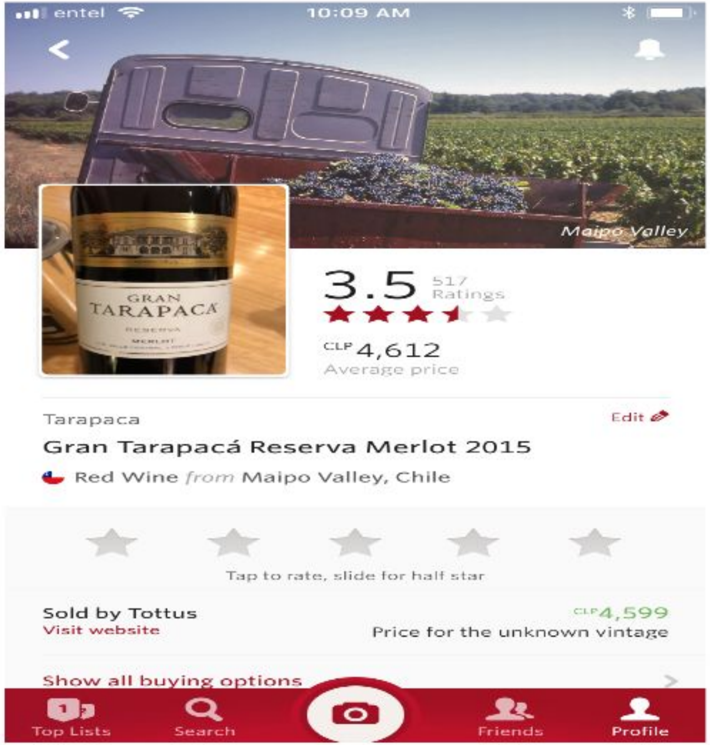
Image 2



Matched SIFT Interest Points


www.photosparis.fr

SIFT Applications - VIVINO



SIFT Applications - Amazon