



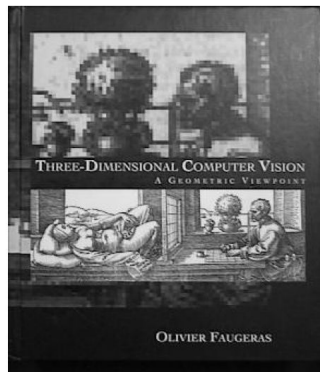
Introducción a la Visión Computacional

Daniela Opitz
Universidad del Desarrollo
Telefónica I+D



Motivación

Detección de Características para Reconocimiento



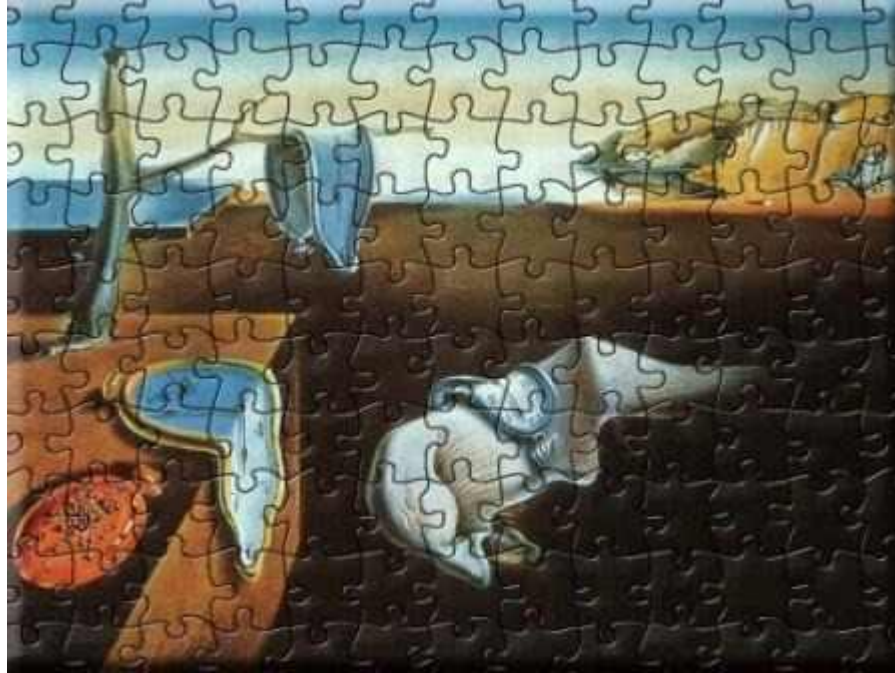
Dónde está libro?

Motivación

Construcción de Panoramas



Detección y Descripción de Características

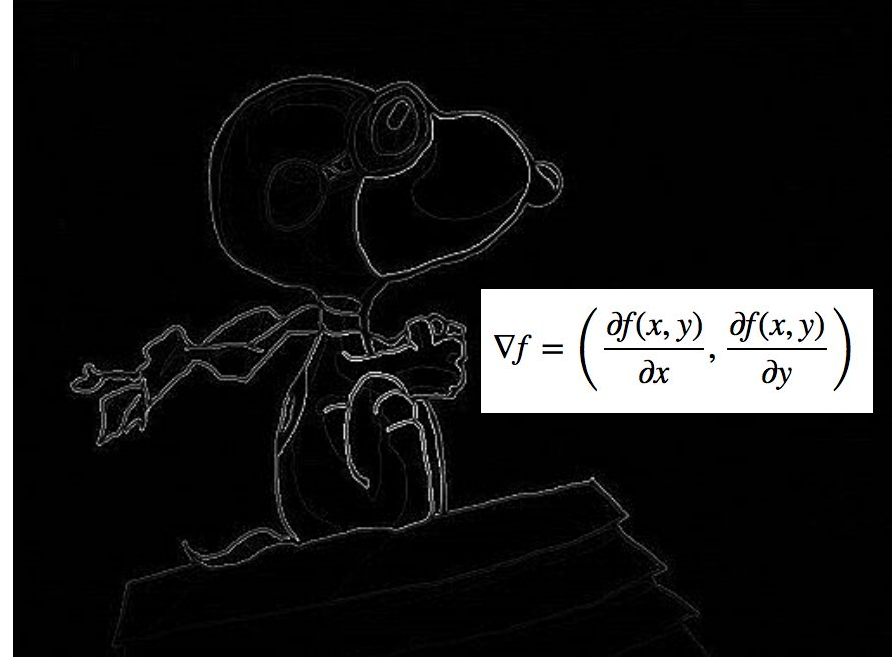


Bordes y Esquinas



Bordes





- La detección de bordes es una herramienta común e importante en **CV**, particularmente en las áreas de detección y extracción de características. Se pueden definir como **el cambio de intensidad entre dos regiones de la imagen**.
- La mayoría de las técnicas para detectar bordes utilizan aproximaciones de la primera y la segunda derivada de la función intensidad de una imagen.



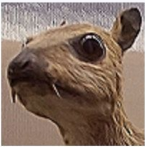
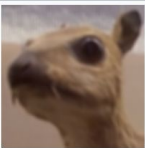
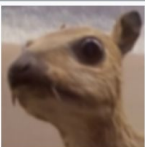
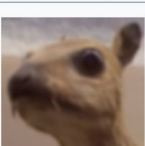
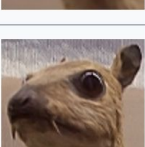
$$\nabla f = \left(\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right)$$

Kernels

- En CV son operadores que sirven para calcular aproximaciones de operaciones específicas.
- También se les conoce como matriz de convoluciones o máscaras.

Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	

Kernels

Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur 3 × 3 (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
Gaussian blur 5 × 5 (approximation)	$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	
Unsharp masking 5 × 5 Based on Gaussian blur with amount as 1 and threshold as 0 (with no image mask)	$\frac{-1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	

¿Cómo Usar Kernels?

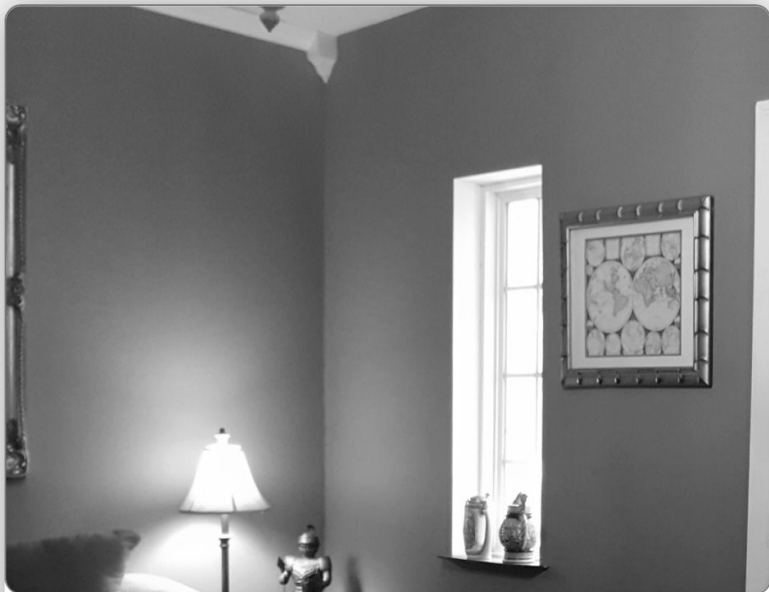


Figure 1: The original grayscale image

105	102	100	97	96
103	99	103	101	102
101	98	104	102	100
99	101	106	104	99
104	104	104	100	98



Figure 2: The first 5 columns and rows of the image in Figure 1

Fuente:

http://machinelearningguru.com/computer_vision/basics/convolution/image_convolution_1.html

¿Cómo Usar Kernels?

1. Volte el kernel horizontal y verticalmente. Como el kernel seleccionado es **simétrico**, el kernel volteado es igual al original.
2. Coloque el elemento central del kernel volteado en cada píxel de la imagen.
3. Multiplique cada elemento del kernel con el píxel correspondiente de la imagen (el que está superpuesto con él) y sume.

$$\text{Kernel} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

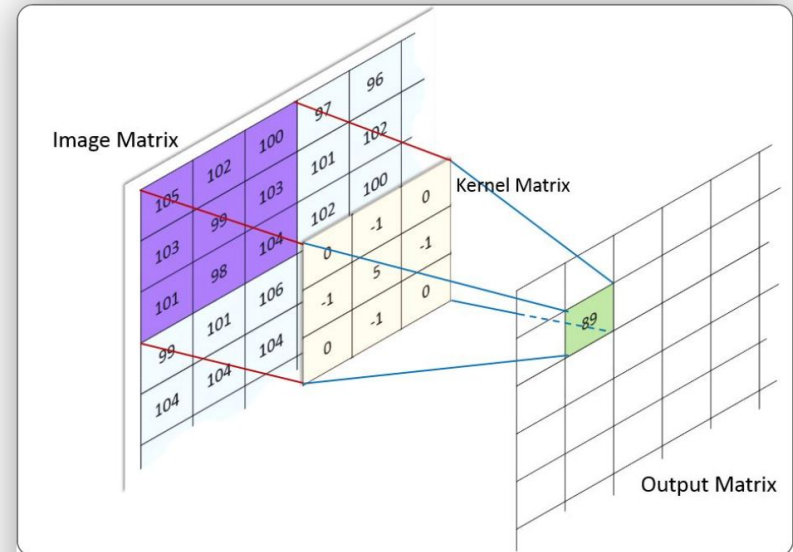


Figure 3: To calculate the value of convolution output at pixel (2,2), center the kernel at the same pixel position on the image matrix

Kernels

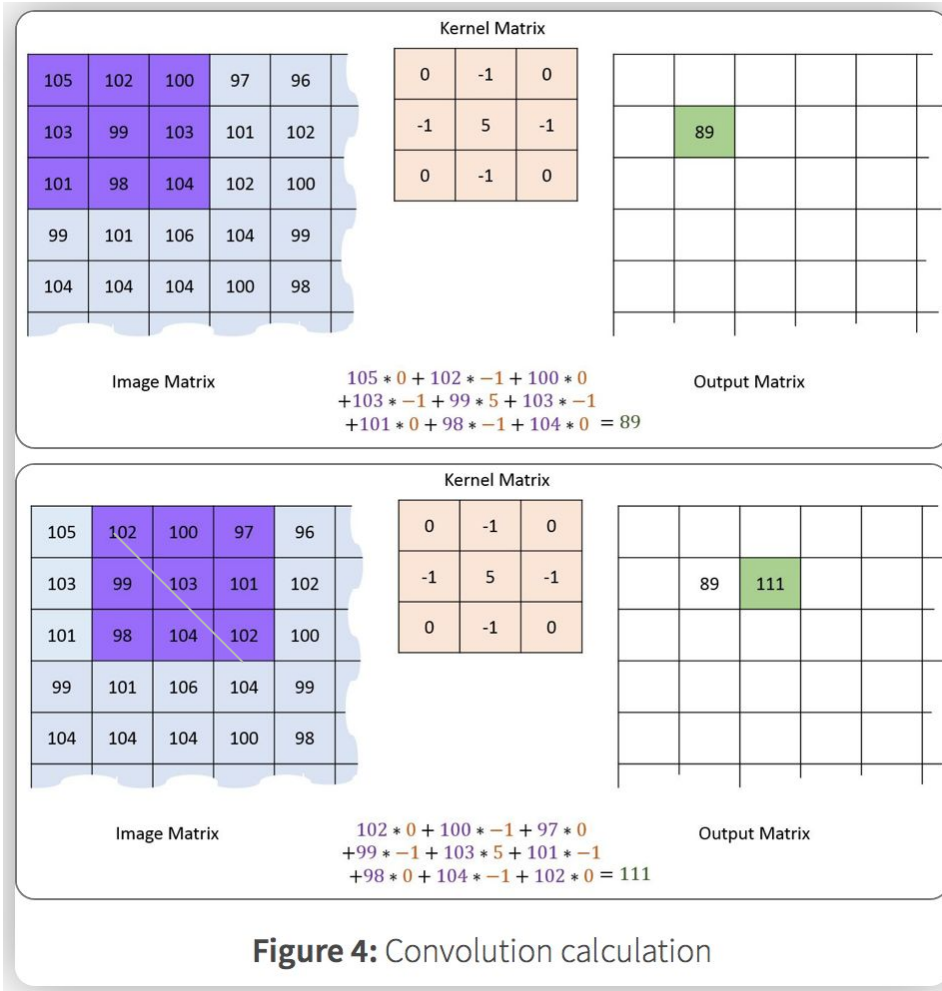
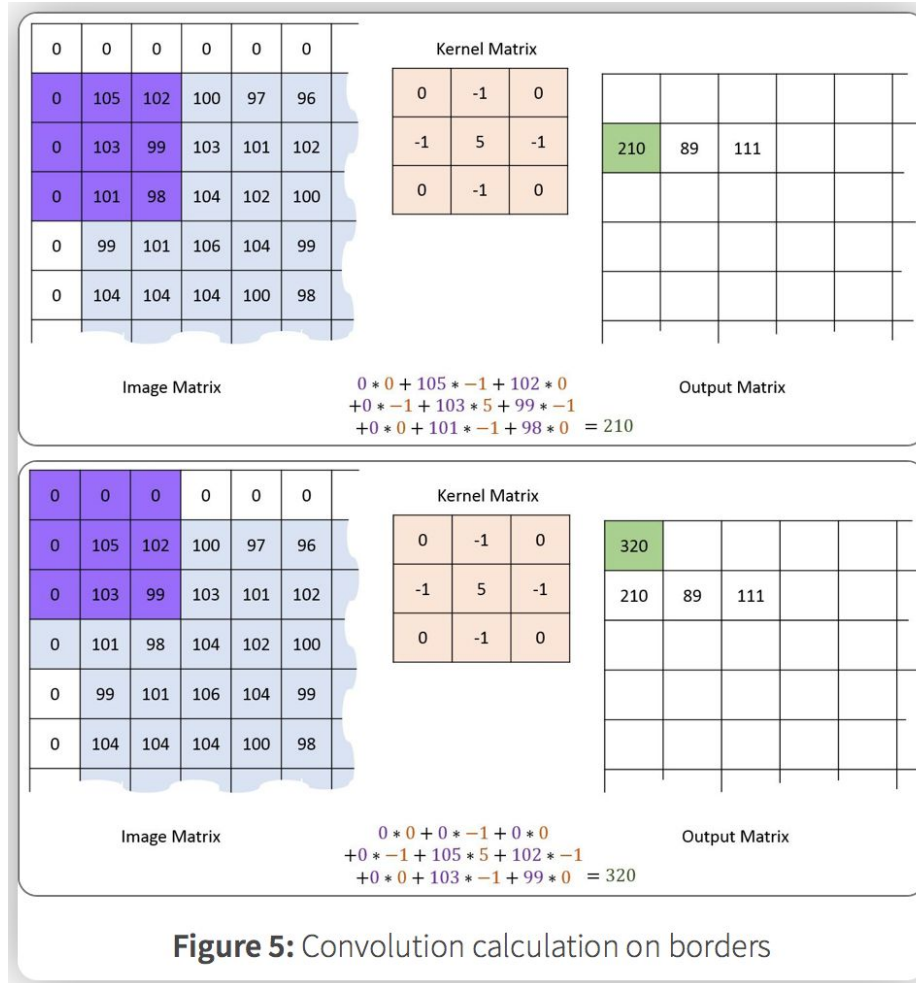


Figure 4: Convolution calculation

Kernels



Kernels



Figure 1: The original grayscale image

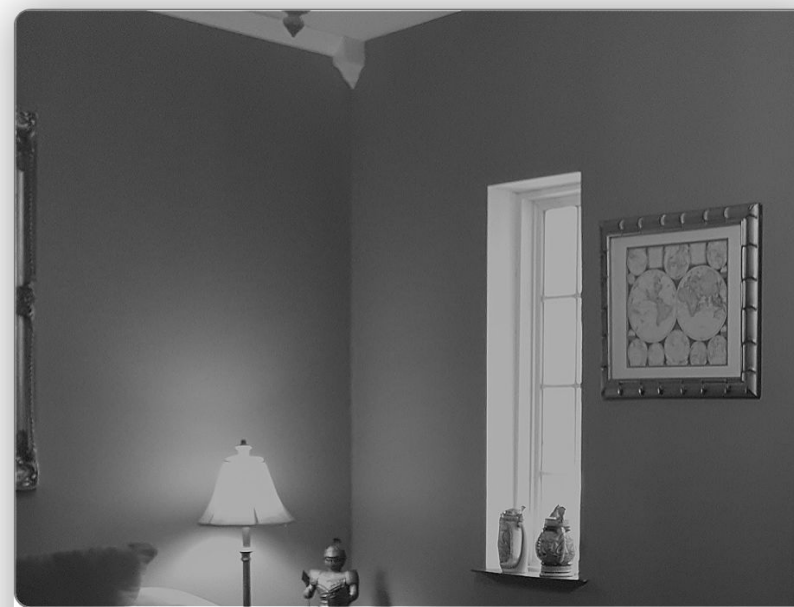


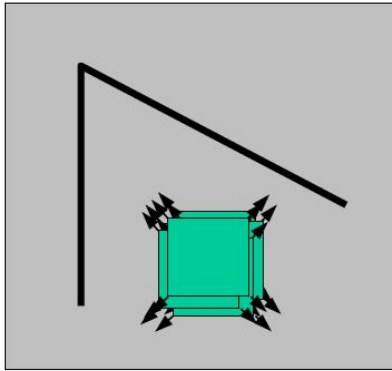
Figure 6: Sharpened image



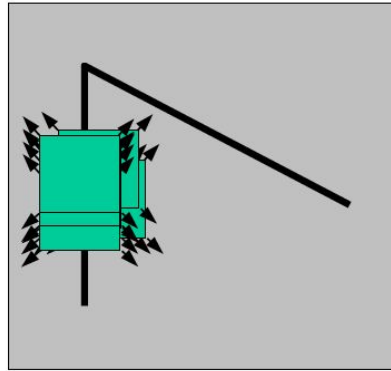
Jupyter

Detector de Esquinas Harris-Detector

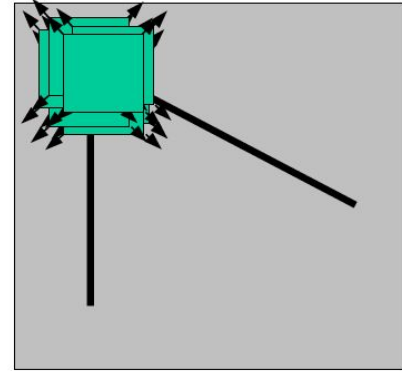
C.Harris, M.Stephens. "A Combined Corner and Edge Detector". 1988



Region Plana
No hay cambios en
ninguna dirección.



Borde
Hay cambios en
algunas direcciones



Esquina
Hay cambios en todas
las direcciones.

Detector de Esquinas Harris-Detector

$$E(u, v) = \sum_{x,y} \underbrace{w(x, y)}_{\text{windows-intensity}} \underbrace{[I(x + u, y + v) - I(x, y)]^2}_{\substack{\text{shifted-intensity} \\ \text{intensity}}}$$

El término $I(x + u, y + v)$ puede ser aproximado utilizando una expansión de Taylor tal que

$$I(x + u, y + v) \approx I(x, y) + I_x(x, y)u + I_y(x, y)v$$

donde I_x I_y son las derivadas o gradientes en las direcciones x e y .

Reemplazando el término anterior en la expresión para $E(u, v)$ se obtiene:

$$E(u, v) \approx \sum_{x,y} w(x, y) [I_x u + I_y v]^2$$

Lo que puede ser reescrito como:

$$E(u, v) \approx (u, v) M \begin{pmatrix} u \\ v \end{pmatrix}$$

donde

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



Jupyter

SIFT - Scale-Invariant Feature Transform

- SIFT es un algoritmo utilizado para detectar y describir características locales en imágenes. Localiza ciertos puntos clave y luego proporciona información cuantitativa (descriptores) que pueden usarse, por ejemplo, para el reconocimiento de objetos. **Los descriptores son invariables frente a rotación y escalamiento.**
- Dos muy buenos links que explican SIFT muy bien (en inglés) son:

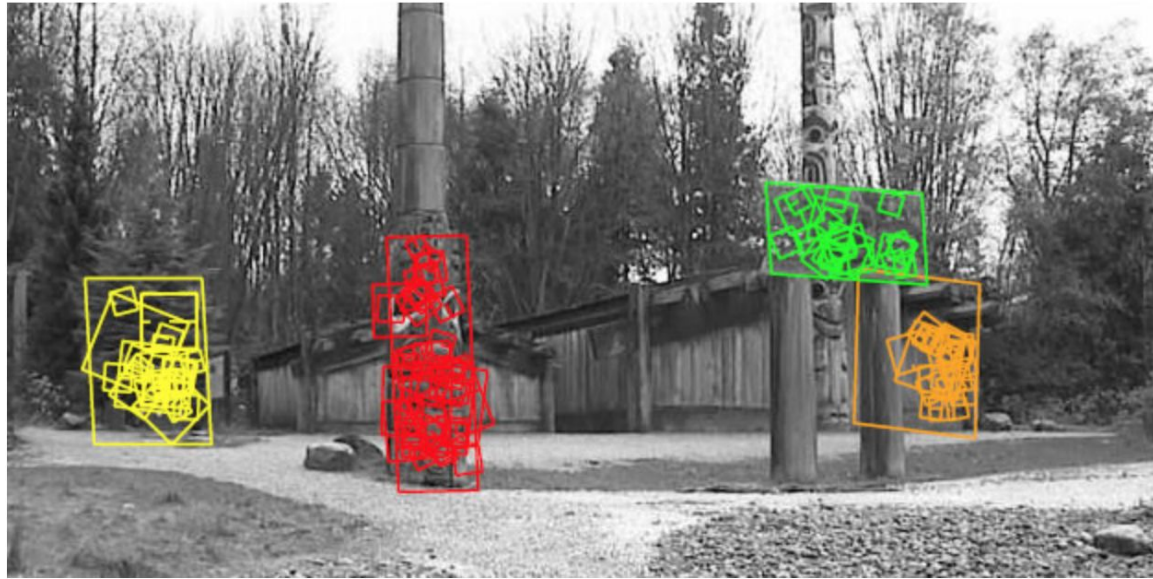
<https://ianlondon.github.io/blog/how-to-sift-opencv/>

<http://aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/>

¿Para Qué Sirve SIFT?



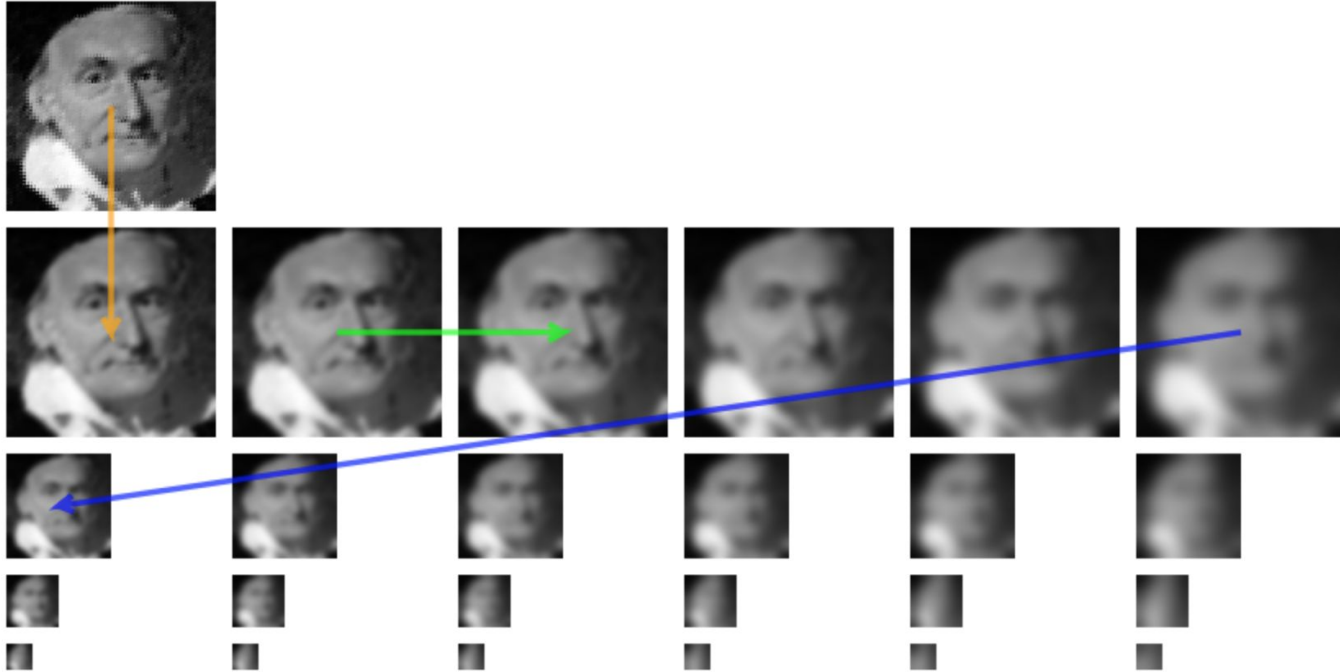
¿Para Qué Sirve SIFT?



¿Cómo Funciona SIFT?

1. Se construye espacio de la escala
2. Se calculan las DoGs (Diferencias de Gaussianas)
3. Se encuentran puntos clave y se remueven los puntos clave de bajo contraste y los bordes.
4. Define orientaciones de los puntos claves.
5. Generación de características

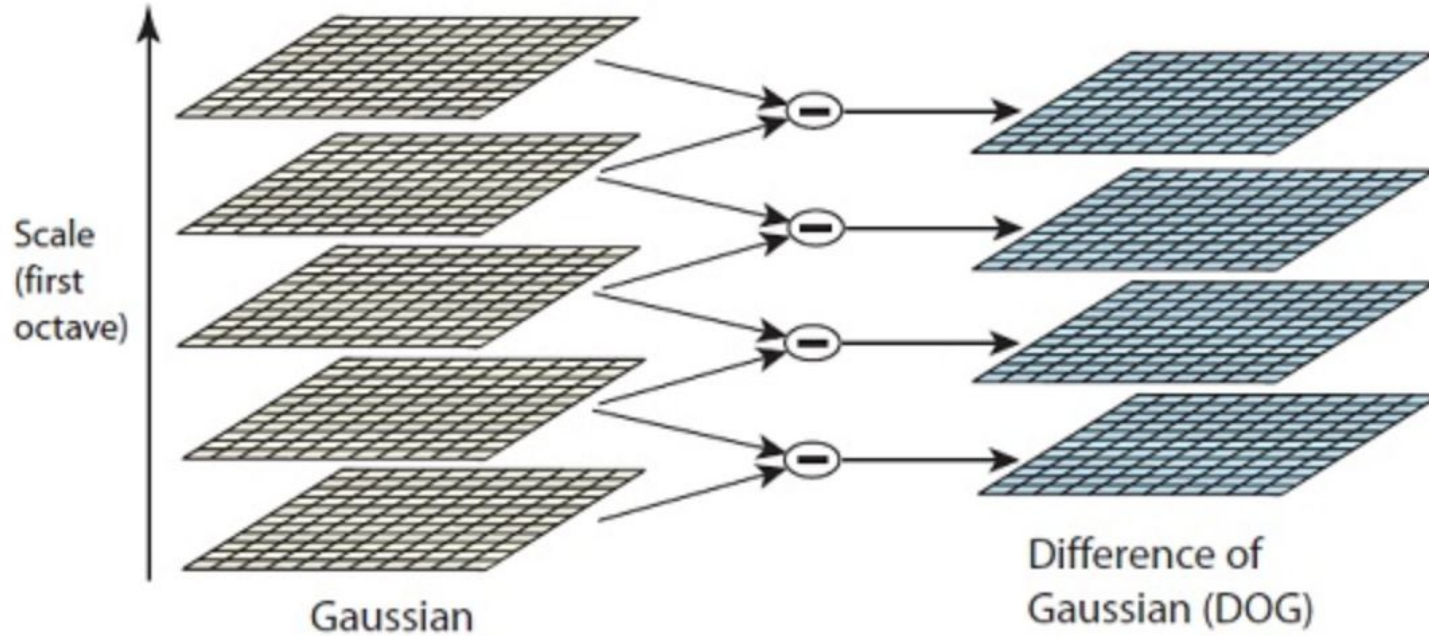
1- ¿Cómo Funciona SIFT? Scale-Space



1- Scale Space

- Se busca simular diferentes escalas de observación (a medida que avanzas más abajo en la tabla) y suprimir las estructuras de escala fina (a medida que te mueves hacia la derecha).
- El algoritmo primero duplica el ancho y la altura de la imagen entrada.
- Esta imagen se difumina posteriormente utilizando una convolución gaussiana (flecha naranja).
- La difuminación es secuencial con desviación estándar creciente (flecha verde).
- La imagen al final de cada fila se encoge (flecha azul) y se difumina otra vez.
- Se repite este proceso hasta que las imágenes son demasiado pequeñas para procesarlas.
- Cada fila se denomina octava.

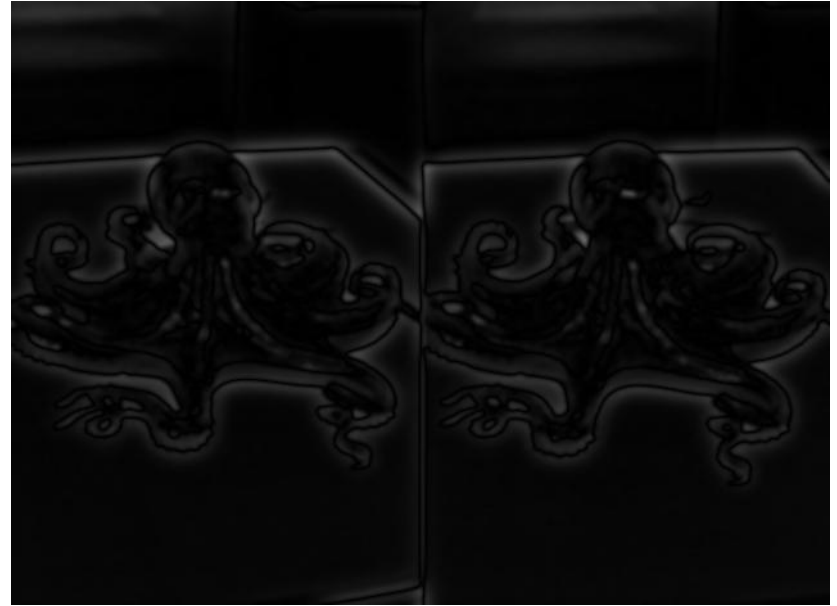
2- DoG (Diferencias de Gaussianas)



2-Diferencias de Gaussianas



sigma 10,5



sigma 25,5

2-Diferencias de Gaussianas



sigma 25,10



sigma 100,25

3-Detección de Puntos Clave

1-Se remueven los bordes calculando gradientes perpendiculares (los bordes no son tan buenos para caracterizar).

2-Para cada punto de interés se guarda su octava.

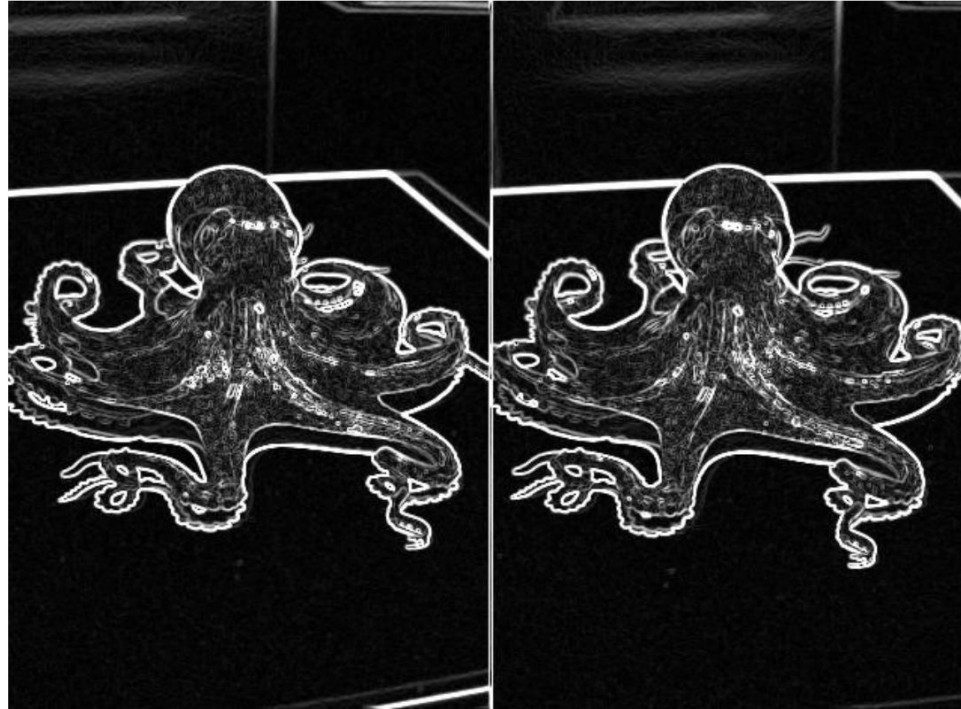
2-Se refinan los puntos de interés removiendo puntos de interés con bajo contraste (intensidad).

-**Región plana:** Dos gradientes iguales.

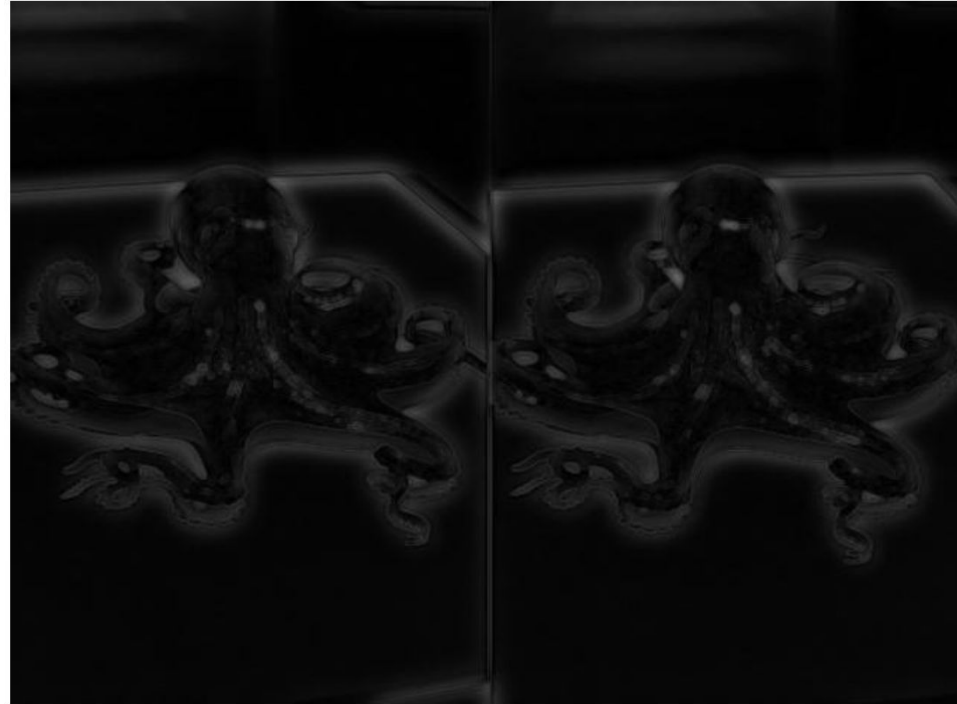
-**Borde:** Un gradiente grande (perpendicular al borde) y otro pequeño (a lo largo del borde).

-**Esquina:** Ambos gradientes son grandes

3-Detección de Puntos Clave



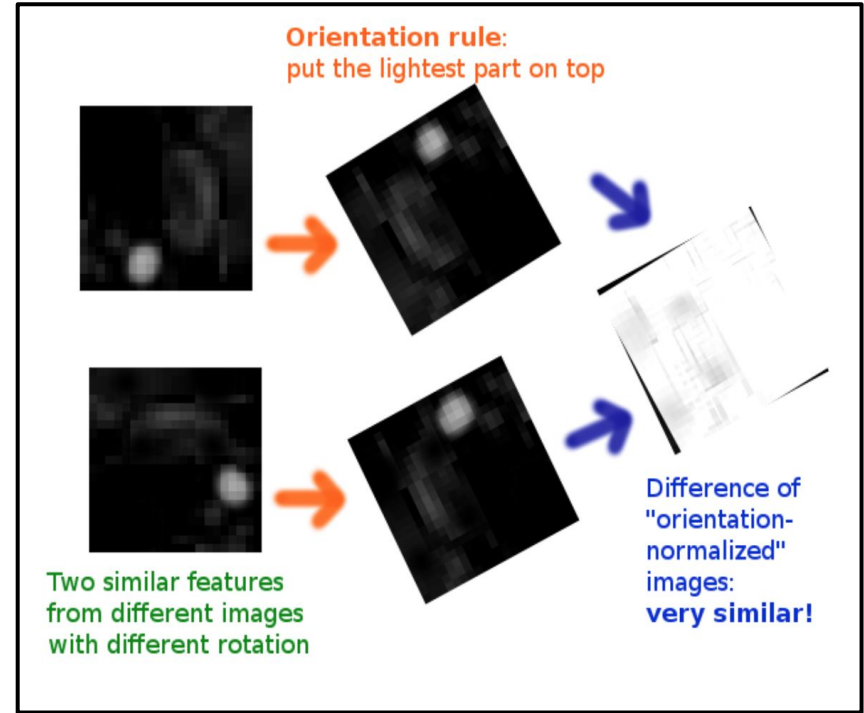
3-Detección de Puntos Clave

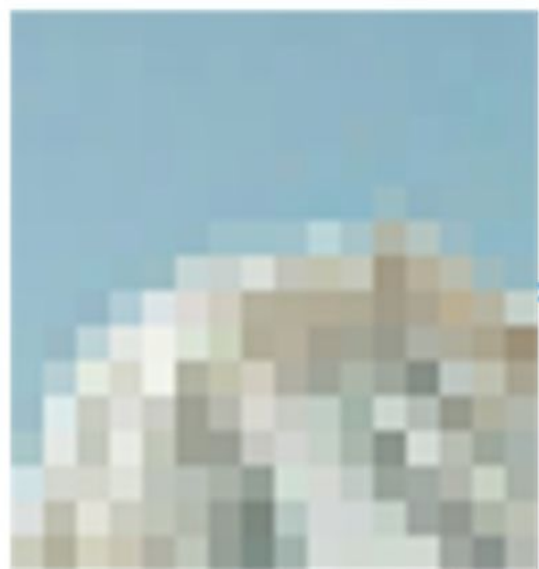


4-Definición de Orientaciones

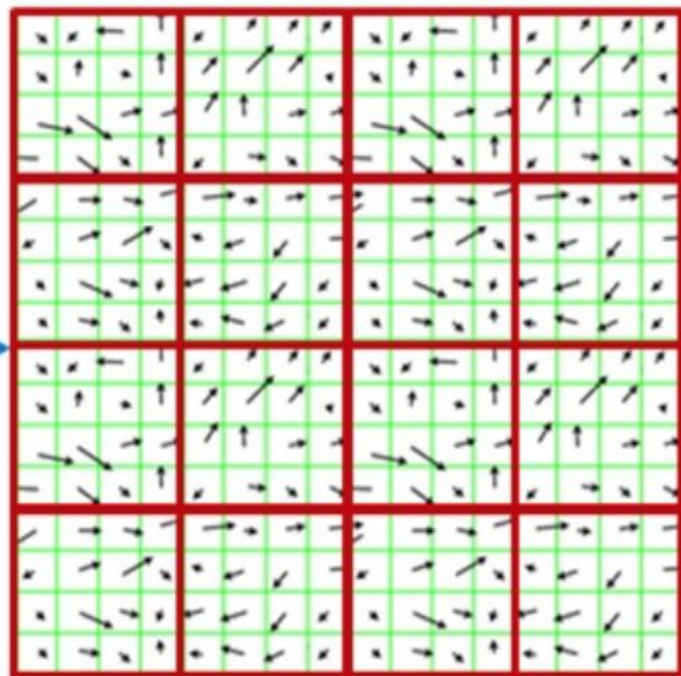
Se asigna una orientación en base a la intensidad.

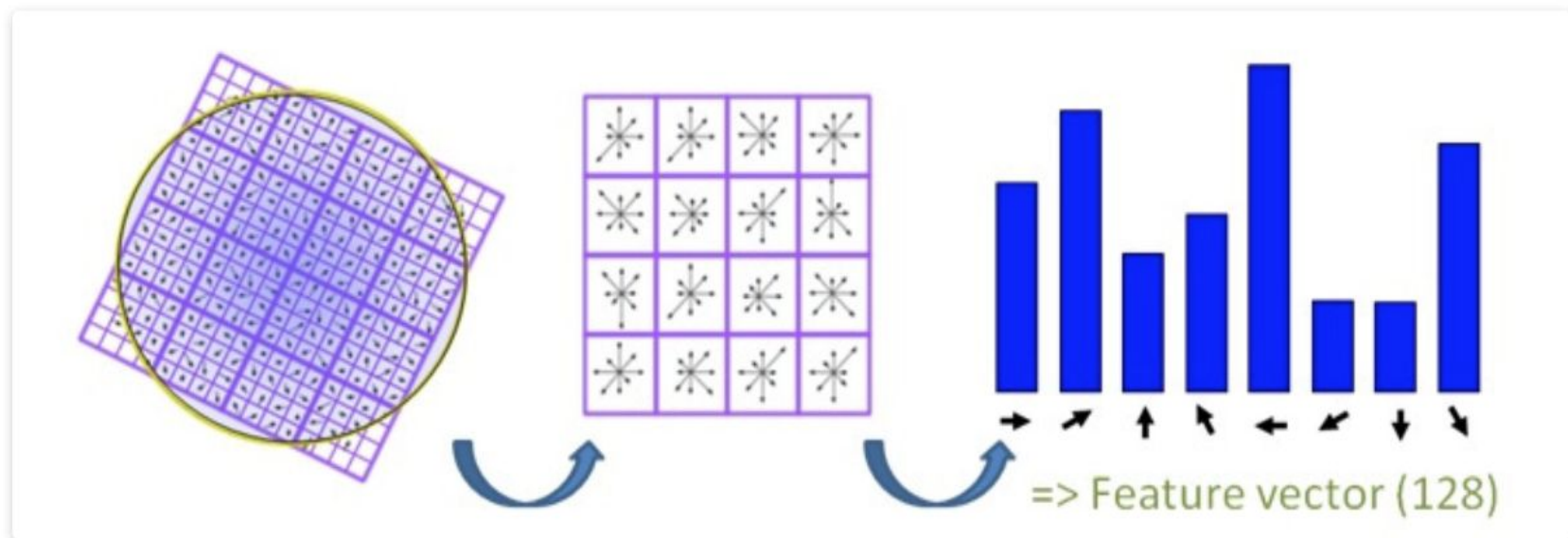
1. Se calculan los gradientes entorno al punto clave.
2. Se calculan las orientaciones y la magnitudes.
3. Se crea un histograma de orientaciones donde cada bin representa un rango en grados de 0-360.
4. Se define el peak del histograma como la orientación de característica.





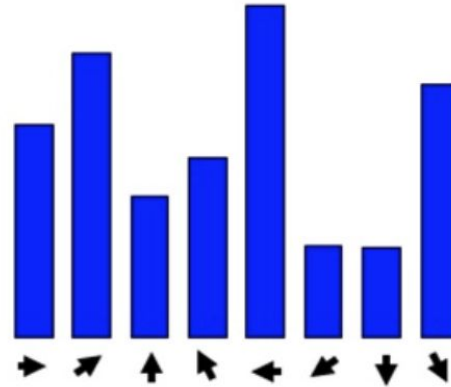
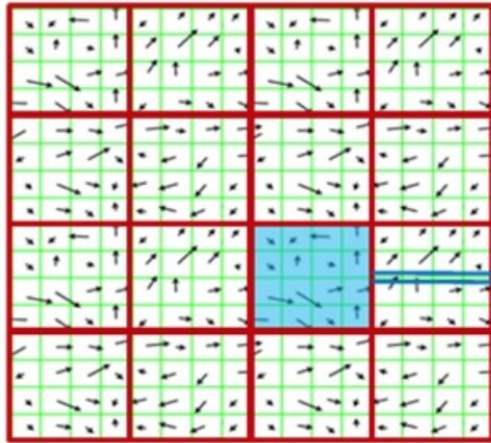
16x 16





SIFT descriptors illustration

4-Definición de Orientaciones





Jupyter