

# Descriptores de imágenes I

Visión por Computador, curso 2023-2024

---

Silvia Martin Suazo, silvia.martin@u-tad.com

11 de octubre de 2023

U-tad | Centro Universitario de Tecnología y Arte Digital

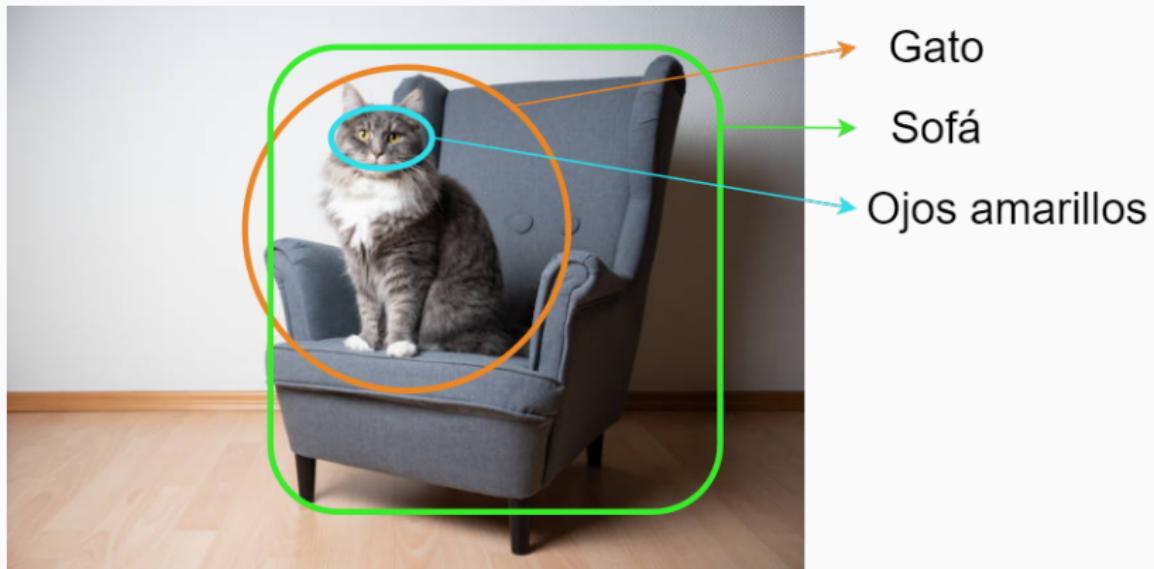


# Introducción

---

# Descripción de imágenes y reconocimiento de objetos

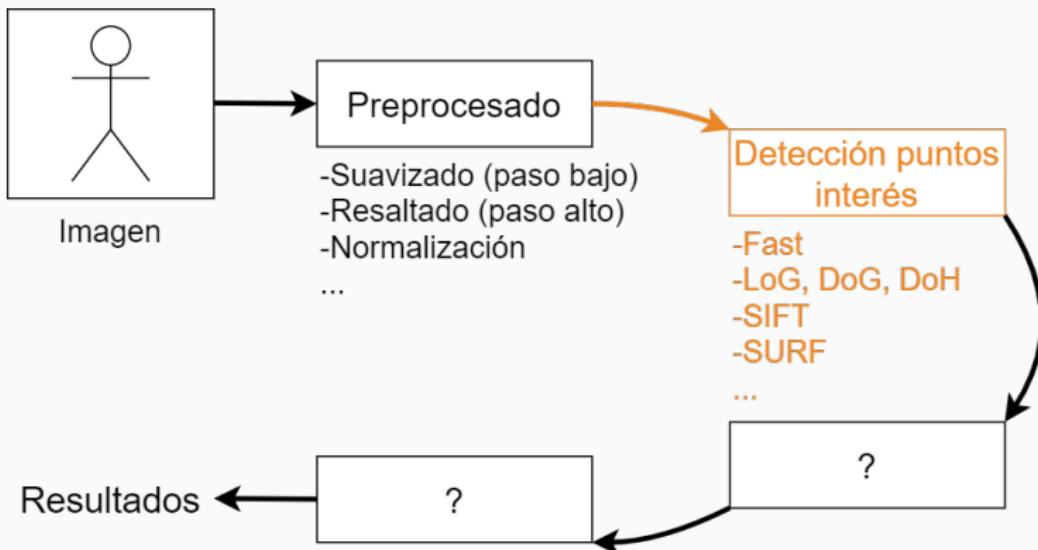
La **descripción de imágenes** es un proceso por el cual se consiguen extraer las **características más importantes** que componen una escena, como por ejemplo qué elementos la componen, en qué posición, etc.



# Descripción de imágenes en el procesado de imágenes

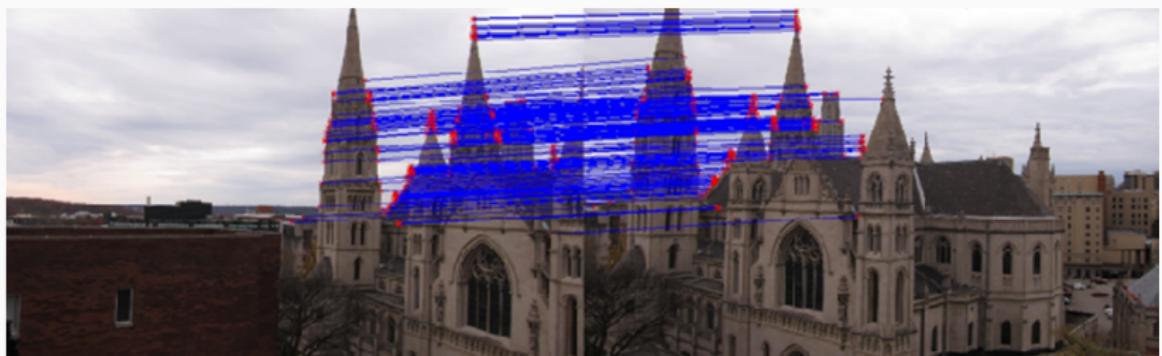
La descripción de imágenes forma parte de uno de los **primeros pasos** a la hora de procesar imágenes.

Dentro de la metodología para procesar imágenes, a través de las descripciones se consiguen identificar **puntos de interés** que identifican la **composición** de una imagen.



# Ejemplos

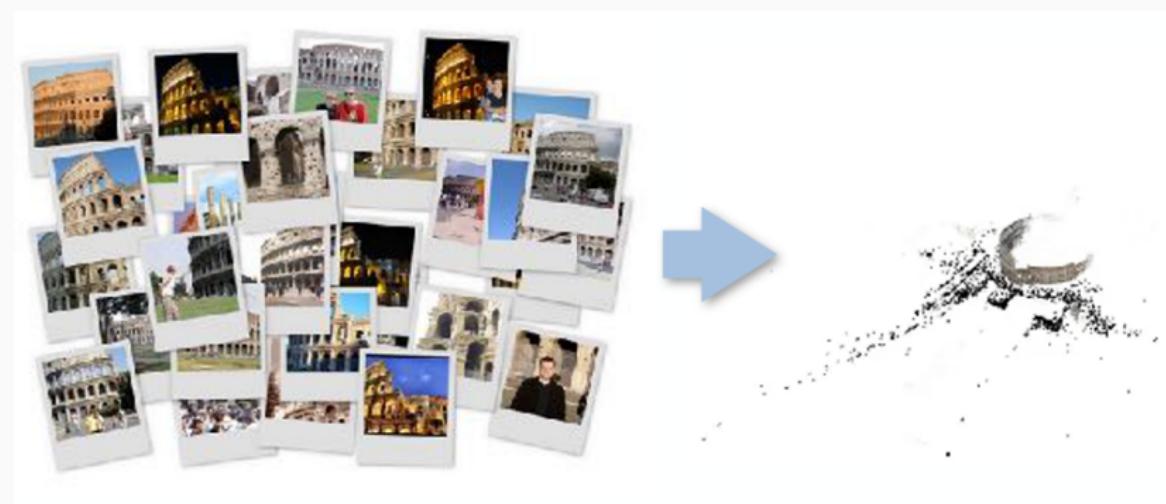
- Correspondencia entre dos imágenes



[1]

# Ejemplos

- Creación de escenarios 3D



[1]

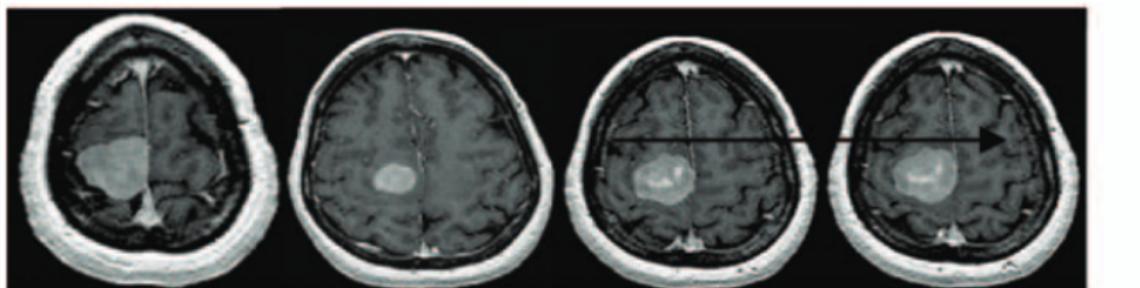
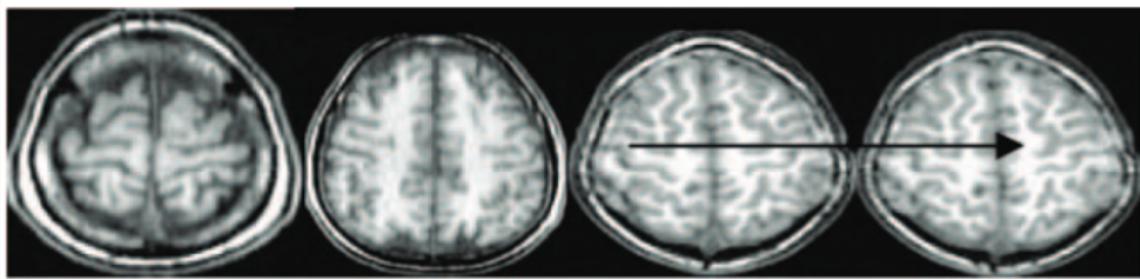
Correspondencia entre imágenes + Estimación de geometría

# Ejemplos

- Reconocimiento de objetos



- Diagnóstico comparativo



[2]

# Algoritmos de detección de puntos de interés

---

# Notebook de ejemplos de descriptores

Los algoritmos que se explicarán a continuación pueden ser observados de manera práctica en el siguiente notebook.



- 02.05-Descriptores.ipynb

Existen dos aproximaciones a la hora de reconocer qué puntos definen a una imagen:

- Detección de puntos o *keypoints*
- Detección de manchas o *blobs*

- Detección de puntos o *keypoints*:

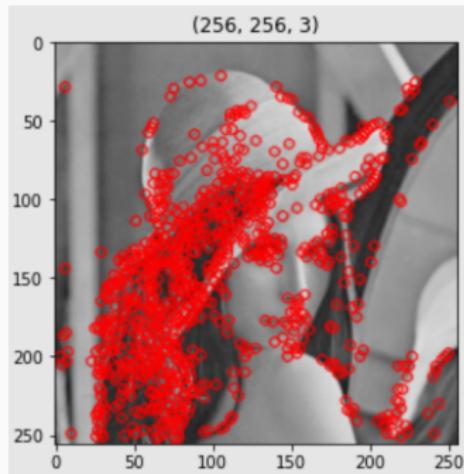
Este conjunto de algoritmos tiene como objetivo **mapear** las distintas estructuras que componen una imagen. Para ello se realiza una **combinación** de los distintos puntos que caracterizan cada una de las partes de la imagen.

Una de las principales ventajas de esta aproximación es la **sencillez** y **rapidez de cómputo**.

# Algoritmo FAST

El algoritmo **Features from Accelerated Segment Test (FAST)**[3] se centra en la detección de **esquinas** como puntos **relevantes** de una imagen.

Una de las principales ventajas de FAST respecto a otros algoritmos como Simultaneous Localization and Mapping (SLAM)[4] es su rápido cálculo, haciendo posible su uso en **tiempo real**.



# Algoritmo FAST

El algoritmo consta de **cinco pasos** diferenciados:

1. Se selecciona un píxel  $p$  de la imagen, que se define con su intensidad  $I_p$ .

# Algoritmo FAST

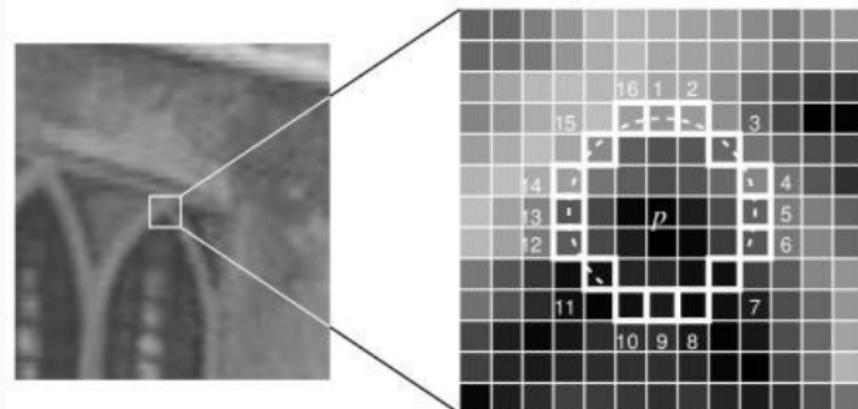
El algoritmo consta de **cinco pasos** diferenciados:

1. Se selecciona un píxel  $p$  de la imagen, que se define con su intensidad  $I_p$ .
2. Se define un umbral  $t$ .

# Algoritmo FAST

El algoritmo consta de **cinco pasos** diferenciados:

1. Se selecciona un píxel  $p$  de la imagen, que se define con su intensidad  $I_p$ .
2. Se define un valor de límite  $t$ .
3. Se dibuja un círculo de **16 píxeles** alrededor del píxel.



[5]

# Algoritmo FAST

El algoritmo consta de **cinco pasos** diferenciados:

1. Se selecciona un píxel  $p$  de la imagen, que se define con su intensidad  $I_p$ .
2. Se define un valor de límite  $t$ .
3. Se escoge un píxel **concreto  $p$**  y se dibuja un círculo de **16 píxeles** alrededor de él.
4. Se evalúa si el píxel  $p$  forma parte es un punto de interés.

Para ello se evalúa si al menos  $n$  (normalmente 12) de los píxeles del círculo son más brillantes que  $I_p + t$  o menos que  $I_p - t$ . Si supera el umbral,  $p$  se considera un punto clave.

# Algoritmo FAST

El algoritmo consta de **cinco pasos** diferenciados:

1. Se selecciona un píxel  $p$  de la imagen, que se define con su intensidad  $I_p$ .
2. Se define un valor de límite  $t$ .
3. Se escoge un píxel **concreto  $p$**  y se dibuja un círculo de **16 píxeles** alrededor de él.
4. Se evalúa si el píxel  $p$  forma una esquina.
5. (Opcional) Supresión no máxima: existe un test de alta velocidad que sólo chequea los píxeles nº **1, 5, 9 y 13**.

Esta variante produce un mayor número de **falsos negativos**, al rechazar muchos menos candidatos.

# Detección de estructuras de interés

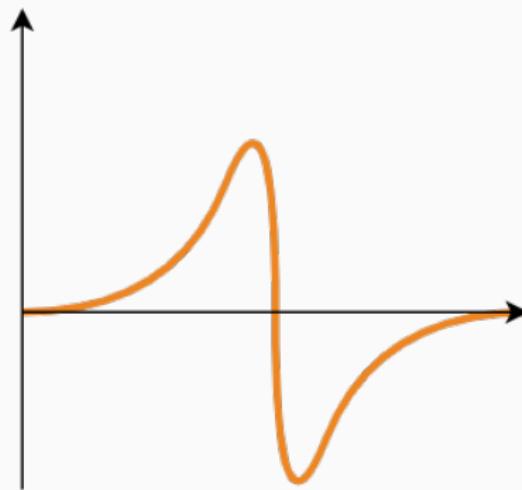
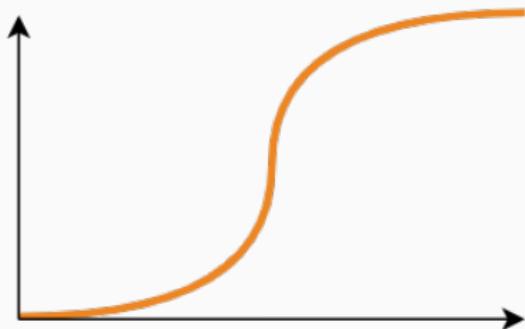
- Detección de manchas o *blobs*:

Los algoritmos de detección de manchas identifican **regiones** de la imagen según su **posición relativa** y **forma**.

Este tipo de estructuras se pueden identificar a través del uso de **segundas derivadas parciales** o **segmentación**.

## ¿Por qué usar segundas derivadas?

Cuando se deriva **dos veces** una función se pueden identificar las pendientes **máximas y mínimas** de una imagen donde la segunda derivada se vuelve cero. Es decir, se buscan regiones donde la intensidad cambia significativamente (bordes o transiciones).



# Identificar regiones con valor cero

La solución pasa por aplicar un **filtrado** de derivada de segundo orden a la imagen y observar qué regiones son mínimas. Dichas regiones indican cambios bruscos en la imagen.

## Problema

Si la **primera derivada** de una imagen es 0, también lo es la **segunda derivada**.

Esto hace que las regiones de **intensidad constante** tengan valor 0.

## Identificar regiones con valor cero

Para identificar los bordes de la imagen se fuerza que la **primera derivada** sea mayor a un **límite**.

$$\left| \frac{dI(x)}{dx} \right| > Th \quad (1)$$

en el caso **unidimensional**.

$$\begin{aligned} |\nabla I(x, y)| &= (I_x^2(x, y) + I_y^2(x, y))^{1/2} > Th \\ \tan \theta &= I_x(x, y) / I_y(x, y) \end{aligned} \quad (2)$$

en el caso **bidimensional**.

## Identificar regiones con valor cero

Al mismo tiempo se busca que la segunda derivada tenga **valor cero**.

$$\frac{d^2I(x)}{dx^2} = 0 \quad (3)$$

en el caso **unidimensional**.

$$\nabla^2 I(x, y) = I_{xx}(x, y) + I_{yy}(x, y) = 0 \quad (4)$$

en el caso **bidimensional**.

## Laplaciano del Gaussiano para detectar blobs

El filtro del Laplaciano del Gaussiano equivale a la segunda derivada de la función Gaussiana.

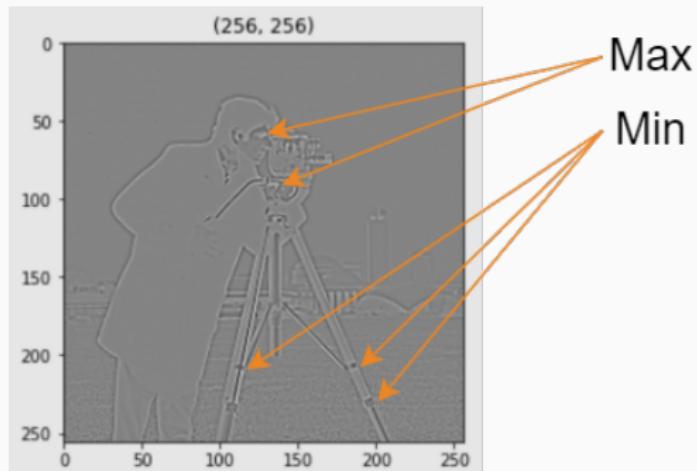
Es por esto que, como hemos visto anteriormente, es utilizado para la detección de bordes de una imagen.

$$\text{ker}^{3 \times 3} = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

# Laplaciano del Gaussiano para detectar blobs

La pregunta ahora es ¿cómo utilizar el filtro Laplaciano para detectar blobs?

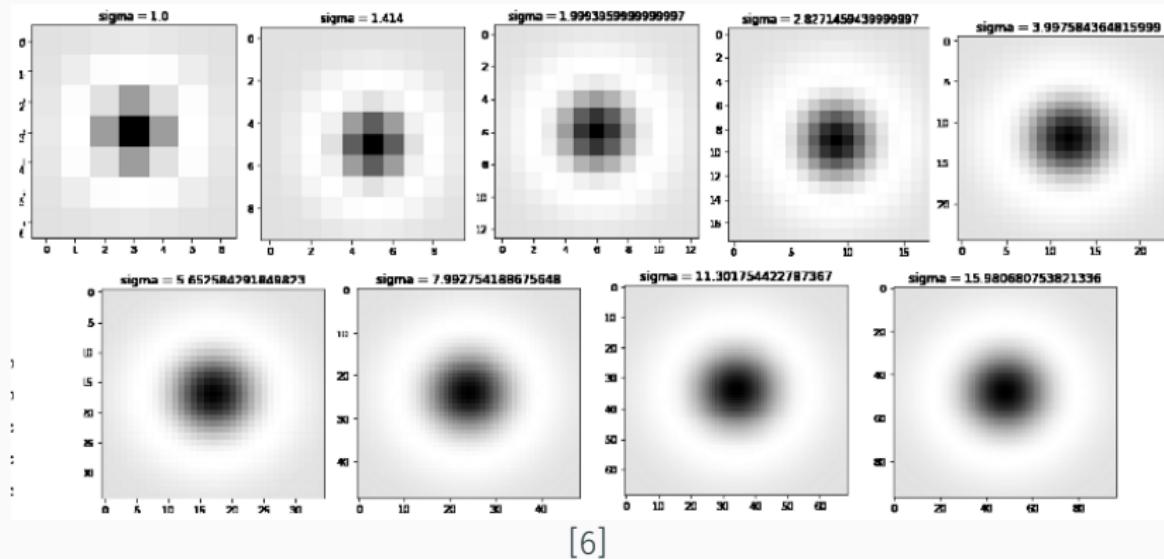
Los puntos máximos o mínimos corresponden con **estructuras de interés** para la **escala** del filtro en concreto (dada por el valor de  $\sigma$ ).



# Laplaciano del Gaussiano para detectar blobs

A la hora de aplicar el Laplaciano del Gaussiano para detectar manchas, este se **aplicar distintas veces** a la misma imagen.

Lo que se pretende es detectar **manchas** a distintas **escalas**, para ello se utilizan filtrados con **distintas desviaciones**.

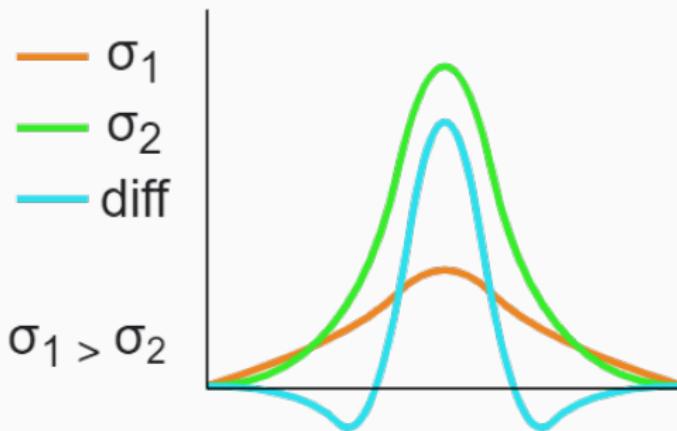


[6]

## Diferencia de Gaussianas

Otro algoritmo para la detección de **blobs** es el de la **diferencia entre Gaussianas**. Este se basa en sustraer una imagen **difuminada** con un filtro Gaussiano de alta desviación, con otra imagen **menos difuminada** con un filtro Gaussiano.

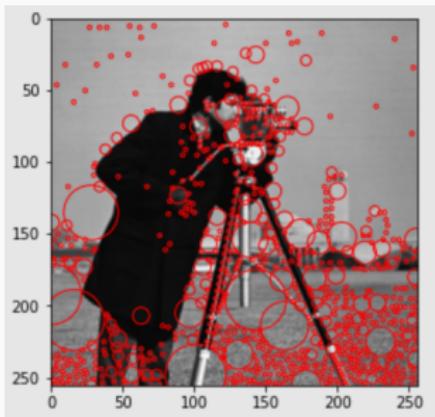
De esta manera se **aproxima** la forma de una **Laplaciana** al mismo tiempo que se **elimina el posible ruido**.



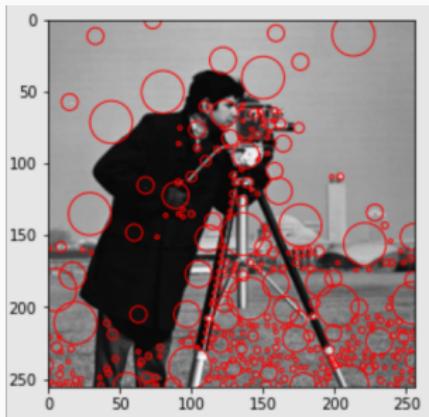
# Diferencia de Gaussianas

Este algoritmo tiene como objetivo **aproximar** los resultados de la Laplaciana de Gaussianas. Para ello también aplica **distintos** filtrados para detectar estructuras a **distintas escalas**.

En comparación con el filtrado Laplaciano del Gaussiano, se consigue una **mejor detección** de manchas de mayor escala.



LoG



DoG

## Determinante de la matriz Hessiana

El cálculo de **blobs** a través de la **matriz Hessiana** se realiza a través del cálculo del **determinante** de dicha matriz, el cual se define con:

$$\det H_{\text{norm}} = t^2 (L_{xx}L_{yy} - L_{xy}^2) \quad (5)$$

donde  $H$  es la matriz Hessiana para cierta escala  $L$ .

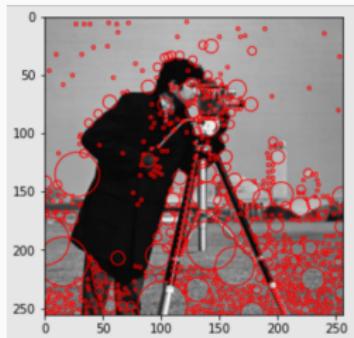
La matriz Hessiana se calcula a través de la siguiente fórmula:

$$H = \begin{pmatrix} \frac{\partial^2 I}{\partial^2 x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial^2 y^2} \end{pmatrix} \quad (6)$$

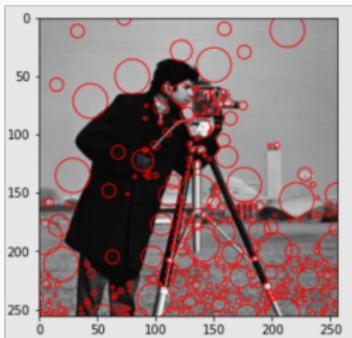
para una imagen  $I$ .

# Determinante de la matriz Hessiana

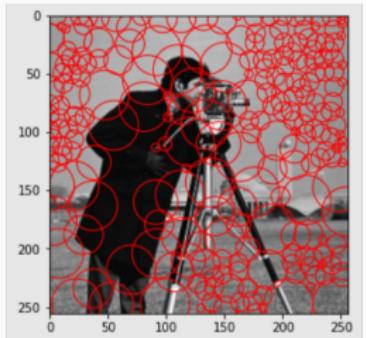
Existen trabajos[7, 8] que atribuyen **mejores resultados** que el Laplaciano del Gaussiano y la diferencia de Gaussianas.



LoG



DoG



DoH

## Referencias i

- [1] Deva Ramanan 16-720 Computer Vision Spring 2017.  
[Online; accessed August, 2022].
- [2] Mana Tarjoman, Emad Fatemizadeh, and Kambiz Badie.  
**An implementation of a cbir system based on svm learning scheme.**  
*Journal of Medical Engineering & Technology*, 37(1):43–47, 2013.
- [3] Deepak Geetha Viswanathan.  
**Features from accelerated segment test (fast).**  
In *Proceedings of the 10th workshop on image analysis for multimedia interactive services, London, UK*, pages 6–8, 2009.
- [4] Hugh Durrant-Whyte and Tim Bailey.  
**Simultaneous localization and mapping: part i.**  
*IEEE robotics & automation magazine*, 13(2):99–110, 2006.

## Referencias ii

- [5] OpenCV.  
**Fast detector images.**  
[Online; accessed August, 2022].
- [6] Nikhil Kumar (Projects Flix).  
**Sigma increment image.**  
[Online; accessed August, 2022].
- [7] Tony Lindeberg.  
**Image matching using generalized scale-space interest points.**  
In *International conference on scale space and variational methods in computer vision*, pages 355–367. Springer, 2013.
- [8] Tony Lindeberg.  
**Image matching using generalized scale-space interest points.**  
*Journal of mathematical Imaging and Vision*, 52(1):3–36, 2015.